Proceedings of

MT Summit XV,
Vol. 1: MT Researchers’ Track

Yaser Al-Onaizan (IBM) & Will Lewis (Microsoft), Eds.

Association for Machine Translation in the Americas
http://www.amtaweb.org

©2015 The Authors. These articles are licensed under a Creative Commons 3.0 license, no derivative works, attribution, CC-BY-ND.
Research Program Co-chairs

Yaser Al-Onaizan IBM T.J. Watson Research Center
Will Lewis Microsoft Research

Research Program Committee

Yuki Arase Osaka University
Wilker Aziz University of Amsterdam
Graeme Blackwood IBM Research
Thorsten Brants Google
Marine Carpuat University of Maryland
Daniel Cer Google
Boxing Chen NRC
Colin Cherry NRC
Jonathan Clark Microsoft Research
Marta R. Costa-jussà Instituto Politécnico Nacional
Steve DeNeefe SDL Language Weaver
Jacob Devlin Microsoft Research
Markus Dreyer SDL Language Weaver
Andreas Eisele DGT, European Commission
Ahmad Emami Bloomberg Labs
Marcello Federico FBK
Christian Federmann Microsoft Research
Minwei Feng IBM Watson Group
Mikel Forcada Universitat d'Alacant
George Foster Google
Kevin Gimpel Toyota Technological Institute at Chicago
Spence Green Lilt Inc
Hany Hassan Microsoft Research
Yifan He New York University
Fei Huang Facebook
Abe Ittycheriah IBM
Philipp Koehn Johns Hopkins University
Roland Kuhn National Research Council of Canada
Shankar Kumar Google
Phillippe Langlais Université de Montréal
Gregor Leusch eBay
Mu Li Microsoft Research Asia
Lemao Liu NICT
Qun Liu Dublin City University
Yanjun Ma Baidu
Wolfgang Macherey Google
Klaus Macherey Google
Saab Mansour eBay
## Contents

<table>
<thead>
<tr>
<th>Page</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Patent Claim Translation based on Sublanguage-specific Sentence Structure</td>
<td>Masaru Fuji, Atsushi Fujita, Masao Utiyama, Eiichiro Sumita and Yuji Matsumoto</td>
</tr>
<tr>
<td>17</td>
<td>Japanese News Simplification: Task Design, Data Set Construction, and Analysis of Simplified Text</td>
<td>Isao Goto, Hideki Tanaka and Tadashi Kumano</td>
</tr>
<tr>
<td>32</td>
<td>Learning Bilingual Distributed Phrase Representations for Statistical Machine Translation</td>
<td>Chaochao Wang, Deyi Xiong, Min Zhang and Chunyu Kit</td>
</tr>
<tr>
<td>44</td>
<td>Learning Bilingual Phrase Representations with Recurrent Neural Networks</td>
<td>Hideya Mino, Andrew Finch and Eiichiro Sumita</td>
</tr>
<tr>
<td>56</td>
<td>A Pilot Study Towards End-to-End MT Training</td>
<td>Feifei Zhai and Liang Huang</td>
</tr>
<tr>
<td>66</td>
<td>Automatic Detection of Antecedents of Japanese Zero Pronouns Using a Japanese-English Bilingual Corpus</td>
<td>Dong Zhan and Hiromi Nakaiwa</td>
</tr>
<tr>
<td>80</td>
<td>METEOR for Multiple Target Languages using DBnary</td>
<td>Zied Elloumi, Hervé Blanchon, Gilles Sérasset and Laurent Besacier</td>
</tr>
<tr>
<td>104</td>
<td>Morphological Constraints for Phrase Pivot Statistical Machine Translation</td>
<td>Ahmed El Kholy and Nizar Habash</td>
</tr>
</tbody>
</table>
Machine translation evaluation made fuzzier: A study on post-editing productivity and evaluation metrics in commercial settings
Carla Parra Escartín and Manuel Arcedillo

A Distributed Inflection Model for Translating into Morphologically Rich Languages
Ke M. Tran, Arianna Bisazza and Christof Monz

Bandit Structured Prediction for Learning from User Feedback in Statistical Machine Translation
Artem Sokolov, Stefan Riezler and Tanguy Urvoy

An Empirical Study on Segment Prioritization for Incrementally Retrained Post-Editing SMT
Jinhua Du, Ankit Srivastava, Andy Way, Alfredo Maldonado-Guerra and David Lewis

Effects of Word Alignment Visualization on Post-Editing Quality & Speed
Lane Schwartz, Isabel Lacruz and Tatyana Bystrova

A Systematic Evaluation of MBOT in Statistical Machine Translation
Nina Seemann, Fabienne Braune and Andreas Maletti

Register-Based Machine Translation Evaluation with Text Classification Techniques
Mihaela Vela and Ekaterina Lapshinova-Koltunski

Error-Tolerant Speech-to-Speech Translation
Rohit Kumar, Sanjika Hewavitharana, Nina Zinovieva, Matthew Roy and Edward Pattison-Gordon

Mixed-Domain vs. Multi-Domain Statistical Machine Translation
Matthias Huck, Alexandra Birch and Barry Haddow

Korean-to-Chinese Word Translation using Chinese Character Knowledge
Yuanmei Lu, Toshiaki Nakazawa and Sadao Kurohashi

Topic Adaptation for Machine Translation of E-commerce Content
Prashant Mathur, Marcello Federico, Selçuk Köprü, Shahram Khadivi and Hassan Sawaf

283 Machine Translation with Source-Predicted Target Morphology
Joachim Daiber and Khalil Sima'an

297 Improved Beam Search with Constrained Softmax for NMT
Xiaoguang Hu, Wei Li, Xiang Lan, Hua Wu and Haifeng Wang

310 Bilingual Distributed Phrase Representations for Statistical Machine Translation
Peyman Passban, Chris Hokamp and Qun Liu

319 Improving the Search Methods for the Interactive Predictions in CAT Systems
Fatemeh Azadi and Shahram Khadivi

333 Improving Semantic SMT via Soft Semantic Role Label Constraints on ITG Alignments
Meriem Beloucif, Markus Saers and Dekai Wu

346 Wiktionary-Based Word Embeddings
Gerard de Melo
Patent Claim Translation based on Sublanguage-specific Sentence Structure

Masaru Fuji  
National Institute of Information and Communications Technology, Kyoto, Japan,  
and Nara Institute of Science and Technology, Nara, Japan  
Atsushi Fujita  
atashi.fujita@nict.go.jp  
Masao Utiyama  
mutiyama@nict.go.jp  
Eiichiro Sumita  
eiichiro.sumita@nict.go.jp  
National Institute of Information and Communications Technology, Kyoto, Japan  
Yuji Matsumoto  
matsu@is.naist.jp  
Nara Institute of Science and Technology, Nara, Japan

Abstract

Patent claim sentences, despite their legal importance in patent documents, still pose difficulties for state-of-the-art statistical machine translation (SMT) systems owing to their extreme lengths and their special sentence structure. This paper describes a method for improving the translation quality of claim sentences, by taking into account the features specific to the claim sublanguage. Our method overcomes the issue of special sentence structure, by transferring the sublanguage-specific sentence structure (SSSS) from the source language to the target language, using a set of synchronous context-free grammar rules. Our method also overcomes the issue of extreme lengths by taking the sentence components to be the processing unit for SMT. The results of an experiment demonstrate that our SSSS transfer method, used in conjunction with pre-ordering, significantly improves the translation quality in terms of BLEU scores by five points, in both English-to-Japanese and Japanese-to-English directions. The experiment also shows that the SSSS transfer method significantly improves structural appropriateness in the translated sentences in both translation directions, which is indicated by substantial gains over 30 points in RIBES scores.

1. Introduction

Advances in reordering techniques based on syntactic parsing (Isozaki et al., 2010b; de Gispert et al., 2015), with growing volumes of parallel patent corpora available, have brought significant improvements in the performance of statistical machine translation (SMT) for translating patent documents across distant language pairs (Goto et al., 2012; Goto et al., 2015). However, among various sentences within a patent document, patent claim sentences still pose difficulties for SMT resulting in low translation quality, despite their utmost legal importance.

A patent claim sentence is written in a kind of sublanguage (Buchmann et al., 1984; Luckhardt, 1991) in the sense that it has the following two characteristics: (i) comprising a patent claim by itself with an extreme length and (ii) having a typical sentence structure composed of a fixed set of components irrespective of language, such as those illustrated in Figures 1 and 2. The difficulties in patent claim translation lie in these two characteristics. Regarding the first characteristic, the extreme lengths cause syntactic parsers to fail with consequent low
reordering accuracy. Regarding the second characteristic, the high regularity of the claim-spe-
cific sentence structure cannot be captured and transferred properly by the models trained only
on the other parts of patent documents, such as the abstract and background description.

This paper presents a method for improving the SMT translation quality of patent claims.
We have developed a system that is used as an add-on to state-of-the-art, off-the-shelf SMT
systems to deal with the sentence structure specific to the patent claim sublanguage. Our method
based on this sublanguage-specific sentence structure (henceforth, SSSS) has two major effects.
(1) Pre-ordering and SMT are applied for each sentence component, rather than for the entire
long sentence. This in effect shortens the input to pre-ordering and SMT, thus improves trans-
lation quality. (2) Claim sentences are translated according to the sentence structure, producing
structurally natural translation outputs. We manually extracted a set of language independent
claim components. Moreover, using these components, we constructed a set of synchronous
rules for English and Japanese to transfer the SSSS in the source language to the target language.

The results of an experiment demonstrate these two major effects of our SSSS transfer
method. Regarding the first effect, when used in conjunction with pre-ordering, our method
improves translation quality by five points in BLEU score (Papineni et al., 2002), in both Eng-
lish-to-Japanese and Japanese-to-English translations. Regarding the second effect, gains in
RIBES score (Isozaki et al., 2010a) of over 30 points are obtained, indicating that our SSSS
transfer is effective in transferring an input sentence structure to the output sentence.

<table>
<thead>
<tr>
<th>Components</th>
<th>Example strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>An apparatus,</td>
</tr>
<tr>
<td>Transitional phrase</td>
<td>comprising:</td>
</tr>
<tr>
<td>Body</td>
<td>Element a pencil;</td>
</tr>
<tr>
<td></td>
<td>Element an eraser attached to the pencil; and</td>
</tr>
<tr>
<td></td>
<td>Element a light attached to the pencil.</td>
</tr>
</tbody>
</table>

*Figure 1. Example of an English patent claim (WIPO, 2014)*

<table>
<thead>
<tr>
<th>Components</th>
<th>Example strings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Body</td>
<td>元鉛筆と；</td>
</tr>
<tr>
<td></td>
<td>鉛筆に取り付けられた消しゴムと；</td>
</tr>
<tr>
<td></td>
<td>鉛筆に取り付けられたライトと</td>
</tr>
<tr>
<td>Transitional phrase</td>
<td>を備える</td>
</tr>
<tr>
<td>Preamble</td>
<td>装置</td>
</tr>
</tbody>
</table>

*Figure 2. Japanese patent claim corresponding to Figure 1*

2. Transferring Claim-Specific Sentence Structure

While patent claims share a common vocabulary and phrases with the rest of the patent docu-
ment, they are written in a distinctive way that is different from the rest of the patent document,
comprising a sublanguage of its own. This writing style of patent claims developed through the
history of filing patent applications, and is now described in the literature. According to the
WIPO Patent Drafting Manual (WIPO, 2014), the fundamental structure of an English claim is
that it is a single sentence consisting of three components:

S → PREA TRAN BODY
where S denotes the claim sentence, PREA the *preamble*, TRAN the *transitional phrase* and BODY the *body*. The *preamble* is an introductory phrase that identifies the category of the invention, the *body* is the main component of the claim that describes the elements or purposes of the invention, and the *transitional phrase* is the component that connects the *preamble* and the *body*.

Figure 1 shows one of the typical structures of English claim sentences, in which the *body* of the claim comprises claim elements. Each of the *elements* is a claim component comprising the invention. Figure 2 shows the structure of a Japanese claim sentence corresponding to the English claim sentence shown in Figure 1. Note that the sets of components comprising the claims in the two languages are identical, although the order of components is different in the two languages.

Our manual analysis revealed that a claim consists of a fixed set of components and the set is common to the two languages. We also found that there are strict generation rules in each language. For example, the English patent claim sentence in Figure 1 is represented by the set of rules in Figure 3, where ELEM denotes the *element* component shown in Figure 1. The symbol “+” denotes a non-null list of the preceding components. The corresponding Japanese sentence is represented by another set of rules comprising the same components, as shown in Figure 4.

Having observed a strong regularity in the structure of patent claim sentences across languages, we represent the structural transfer in the form of synchronous context-free grammar (SCFG). For example, we derive the SCFG rules in Figure 5 by connecting the corresponding rules in Figures 3 and 4, where the numeric indices indicate correspondences between non-terminals in both constituent trees. We handcrafted a set of SCFG rules for translating patent claim sentences. The details of the process are presented in Section 3.1.

![Figure 3. Example of generation rules for an English claim sentence](image3.png)

![Figure 4. Example of generation rules for a Japanese claim sentence](image4.png)

![Figure 5. SCFG rules derived from English rules in Figure 3 and Japanese rules in Figure 4](image5.png)

3. Pipeline for Patent Claim Translation

While patent claim sentences have a distinctive structure, their components, such as the *elements* and *purposes* of the claimed inventions, are described with the same vocabulary and phrases in the other parts of patent documents. We therefore implemented the SSSS transfer as an add-on to off-the-shelf SMT systems. More specifically, given a patent claim sentence in the
source language, our method translates it through the following three-step pipeline (see also Figure 6).

A button comprising: a plurality of first ribs integrally formed on the surface of the plate-like base portion, each rib radially extending from a center towards the circumference of the plate-like base portion; and an annular portion integrally formed on the surface of the plate-like base portion, to which each center ends of the plurality of first ribs are coupled.

(a) Input English sentence

\[
S \begin{array}{l}
\text{[PREA A button]} \quad \text{[TRAN comprising:]} \\
\text{[BODY [ELEM a plurality of first ribs integrally formed on the surface of the plate-like base portion, each rib radially extending from a center towards the circumference of the plate-like base portion;] [ELEM and an annular portion integrally formed on the surface of the plate-like base portion, to which each center ends of the plurality of first ribs are coupled.]}}
\end{array}
\]

(b) Synchronously obtained English SSSS

\[
S \begin{array}{l}
\text{[BODY [ELEM a plurality of first ribs integrally formed on the surface of the plate-like base portion, each rib radially extending from a center towards the circumference of the plate-like base portion;] [ELEM and an annular portion integrally formed on the surface of the plate-like base portion, to which each center ends of the plurality of first ribs are coupled;] [TRAN と備える] [PREA A button]}
\end{array}
\]

(c) Synchronously generated Japanese SSSS

\[
S \begin{array}{l}
\text{[BODY [ELEM plate like base portion of circumference towards center from extending plate like base portion of surface on formed integrally first ribs of plurality, each rib radially;] [ELEM and plate like base portion of surface, plurality of first ribs of each center ends coupled are which to on formed integrally annular portion;] [TRAN と備える] [PREA A button]}
\end{array}
\]

(d) Each SSSS component pre-ordered

\[
S \begin{array}{l}
\text{[BODY [ELEM 前記板状ベース部の前記表面で一体に形成され、各々が前記板状ベース部の中心から外周に向かって放射状に延在する複数の第1リブと、] [ELEM 前記板状ベース部の前記表面で一体に形成され、前記複数の第1リブ各々の中心端が連結された環状部と、] [TRAN と備える] [PREA ボタン]}
\end{array}
\]

(e) Each SSSS component translated by English-to-Japanese SMT

Figure 6. Overview of our translation pipeline
1. **Step 1. SSSS transfer** (Figure 6: (a) → (b), (c)): The given sentence is analyzed using a set of handcrafted SCFG rules. The goal of this step is not to obtain a fine-grained parse tree of the input sentence, but to identify its sublanguage-specific structure, and transfer it to the target language. By the use of the set of SCFG rules, the components in the given sentence are identified, and simultaneously the sentence structure in the target language is generated.

2. **Step 2. Pre-ordering** (Figure 6: (c) → (d)): The words of each component are reordered so that the order becomes close to that in the target language. This process is performed using a constituent parser. As a result of Step 1, shorter word sequences are the input to this process, resulting in higher parsing and reordering accuracy.

3. **Step 3. Translation by SMT** (Figure 6: (d) → (e)): Each component is translated by an SMT system, and the translated components joined up to form a sentence, with words conjugated and conjunctions added as necessary. Again, as a result of Step 1, shorter components are input that are easier to translate.

The rest of this section elaborates Steps 1 and 2 in turn.

### 3.1. SSSS Transfer

As described in Section 1, one of the major issues in patent claim translation is that, despite the high regularity, the claim-specific sentence structure cannot be captured and transferred properly by models trained only on the other parts of patent documents.

This step is introduced to identify the structure of the given patent claim sentence and to generate the structure in the target language simultaneously. This process is performed using a set of handcrafted SCFG rules. We created the rules in the following manner. First, we manually analyzed the English and Japanese claim sentences in our development set (described in Section 4.1) and found that each claim sentence is composed of a fixed set of components and that the set is common to the two languages. The set of components U we have identified is as follows:

\[ U \in \{\text{PREAMBLE, TRANSITIONAL PHRASE, BODY, ELEM, PURPOSE}\}, \]

where the first four are explained in the previous section, i.e., *preamble*, *transitional phrase*, *body* and *element*. PURPOSE denotes the *purpose* component, which is similar to the *element* component in the sense that they comprise the *body* component.

We then constructed a set of generation rules for English and Japanese claims using U as a set of non-terminal symbols, and obtained 8 and 16 generation rules respectively. We obtained a larger number of rules for Japanese, because the writing style of Japanese claim sentences is more flexible than that of English claim sentences. Finally, we handcrafted a total of 16 SCFG rules by combining the generation rules for the two languages that have the same set of symbols on both the left- and right-hand sides, respectively. Table 1 shows the entire SCFG rule set for English-to-Japanese translation. Our SCFG rules for Japanese-to-English translation are produced by reversing the above English-to-Japanese generation rules.

In the actual implementation of the SCFG rules, we designed each of the rules in the rule set to be deterministic, by using regular expressions for obtaining a unique match for a terminal symbol. For example, to analyze input sentences containing more than one occurrence of the string "comprising:" we prepared a regular expression to match the first occurrence. This heuristic rule correctly matches the claim string in most cases.
### Table 1. SCFG rule set for English-to-Japanese translation

<table>
<thead>
<tr>
<th>ID</th>
<th>SCFG rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>$S \rightarrow \langle \text{EIFA}_1 \text{TRAN}_2 \text{BODY}_3 \text{EIFA}_3 \text{BODY}_3 \text{TRAN}_2 \text{EIFA}_1 \rangle$</td>
</tr>
<tr>
<td>R2</td>
<td>$S \rightarrow \langle \text{EIFA}_1 \text{TRAN}_2 \text{BODY}_3 \text{EIFA}_3 \text{BODY}_3 \text{TRAN}_2 \text{EIFA}_1 \rangle$</td>
</tr>
<tr>
<td>R3</td>
<td>$S \rightarrow \langle \text{EIFA}_1 \text{TRAN}_2 \text{BODY}_3 \text{EIFA}_3 \text{BODY}_3 \text{TRAN}_2 \text{EIFA}_1 \rangle$</td>
</tr>
<tr>
<td>R4</td>
<td>$S \rightarrow \langle \text{EIFA}_1 \text{TRAN}_2 \text{BODY}_3 \text{EIFA}_3 \text{BODY}_3 \text{TRAN}_2 \text{EIFA}_1 \rangle$</td>
</tr>
<tr>
<td>R5</td>
<td>$S \rightarrow \langle \text{EIFA}_1 \text{TRAN}_2 \text{BODY}_3 \text{EIFA}_3 \text{BODY}_3 \text{TRAN}_2 \text{EIFA}_1 \rangle$</td>
</tr>
<tr>
<td>R6</td>
<td>$S \rightarrow \langle \text{EIFA}_1 \text{TRAN}_2 \text{BODY}_3 \text{EIFA}_3 \text{BODY}_3 \text{TRAN}_2 \text{EIFA}_1 \rangle$</td>
</tr>
<tr>
<td>R7</td>
<td>$\text{BODY} \rightarrow \langle \text{ELEM}^+, \text{ELEM}^+ \rangle$</td>
</tr>
<tr>
<td>R8</td>
<td>$\text{BODY} \rightarrow \langle \text{PURP}^+, \text{PURP}^+ \rangle$</td>
</tr>
<tr>
<td>R9</td>
<td>$\text{TRAN} \rightarrow \langle \text{comprising}, \text{備えることを特徴とする} \rangle$</td>
</tr>
<tr>
<td>R10</td>
<td>$\text{TRAN} \rightarrow \langle \text{comprising}, \text{備える} \rangle$</td>
</tr>
<tr>
<td>R11</td>
<td>$\text{TRAN} \rightarrow \langle \text{including}, \text{備えることを特徴とする} \rangle$</td>
</tr>
<tr>
<td>R12</td>
<td>$\text{TRAN} \rightarrow \langle \text{including}, \text{備える} \rangle$</td>
</tr>
<tr>
<td>R13</td>
<td>$\text{TRAN} \rightarrow \langle \text{having}, \text{備えることを特徴とする} \rangle$</td>
</tr>
<tr>
<td>R14</td>
<td>$\text{TRAN} \rightarrow \langle \text{having}, \text{備える} \rangle$</td>
</tr>
<tr>
<td>R15</td>
<td>$\text{TRAN} \rightarrow \langle \text{wherein}, \text{ことの特徴とする} \rangle$</td>
</tr>
<tr>
<td>R16</td>
<td>$\text{TRAN} \rightarrow \langle \text{wherein}, \text{する} \rangle$</td>
</tr>
</tbody>
</table>

### 3.2. Pre-ordering

Another major issue in patent claim translation is that the extreme lengths cause syntactic
parsers to fail with consequent low reordering accuracy. To evaluate the effect of introducing
our SSSS transfer on the translation quality, we also implemented a pre-ordering tool using
state-of-the-art techniques (Isozaki et al., 2010b; Goto et al., 2012; Goto et al., 2015).

Our pre-ordering method is based on syntactic parsing. First, the input sentence is parsed
into a binary tree structure by using the Berkeley Parser (Petrov et al., 2006). For example,
when “He likes apples.” is inputted into our English-to-Japanese translation system, it is parsed
as shown in Figure 7. Second, the nodes in the parse tree are reordered using a classifier. For
example, according to the classifier's decision, the two children of the “VP” node, i.e., “VBZ”
and “NP”, are swapped, whereas the order of the two children of the “S” node, i.e., “NP” and
“VP”, is retained. Once such a decision is made for every node with two children (henceforth,
_binary mode_), the word order of the entire sentence becomes very similar to that in Japanese,
i.e., “He (kare wa) apples (ringo ga) likes (suki da) . (.)”

The pre-ordering model is trained on a given parallel corpus through the following pro-
cedure (Section 4.5 of Goto et al., 2015):

1. Parse the source sentences of the parallel corpus.
2. Perform word alignment on the parallel corpus.
3. Reorder words in each source sentence by swapping some binary nodes so that
Kendall's τ over the aligned source and target sentences is maximized. As a

---

1 Note that we used the parse model trained from the source treebank, while Goto et al. (2015) used the
parse model learned via cross-language syntactic projection.
result, every binary node is classified as either SWAP, i.e., the two children of the node are swapped, or STRAIGHT, i.e., they are not swapped.

4. With the above data, a neural network classifier is trained for predicting whether a given node is SWAP or STRAIGHT.2

The constituent parser is also domain-adapted. The initial parsing model for English was trained on the sentences in the Penn Treebank3 as well as 3,000 patent sentences manually parsed by the authors. The initial model for Japanese was trained on the EDR Treebank4 consisting of approximately 200,000 sentences. In contrast to what we did for English, we did not use patent sentences in Japanese because no annotator was available.

We first parsed 200,000 patent sentences using the initial parsing model. We then built a patent-adapted (not claim-adapted) parsing model by applying a self-learning procedure (Huang et al., 2009) to the above automatic parses.

```
(ROOT
  (S
    (NP (PRP He))
    (VP (VBZ likes)
      (NP (NNS apples)))
    (.
     .)))
```

Figure 7. Parsing result of “He likes apples.”

4. Experiments

We evaluated to what extent our SSSS transfer and pre-ordering improved the translation quality. As mentioned in Section 3, these methods are implemented as an add-on to off-the-shelf SMT systems. In particular, we used phrase-based SMT (Koehn et al., 2003) as the base system. We also regard it and its hierarchical version (Chiang, 2005) as baseline SMT systems.

4.1. Data

The training data for SMT consists of two subcorpora. The first is the Japanese-English Patent Translation data comprising 3.2 million sentence pairs provided by the organizer of the Patent Machine Translation Task (PatentMT) at the NTCIR-9 Workshop (Goto et al., 2011). We randomly selected 3.0 million sentence pairs. Henceforth, we call this Corpus A. SMT systems trained on the corpus are reasonably good at lexical selection in translating claim sentences, because the vocabulary and phrases are commonly used in entire patent documents, and Corpus A is of a substantial size to cover a large portion of them. However, the claim-specific sentence structure would never be taken into account, as Corpus A does not contain any claim sentences.

To bring claim-specific characteristics into the SMT training, even for the baseline systems, we also used Corpus B comprising 1.0 million parallel sentences of patent claims. These were automatically extracted from pairs of English and Japanese patent documents published between 1999 and 2012 using a sentence alignment method (Utiyama and Isahara, 2007). The concatenation of Corpora A and B was used to train baseline SMT systems, as well as those for our extensions.

2 Note that Goto et al. (2015) learned the SWAP/STRAIGHT classification problem jointly with the parsing source sentences.
3 https://www.cis.upenn.edu/~treebank/
4 https://www2.nict.go.jp/out-promotion/techtransfer/EDR/index.html
Development and test data were constructed separately from the training data in the following manner. First, we randomly extracted English patent documents from patents filed in the USA in 2014 and extracted up to the first five claims from each patent document. Then, we randomly selected 2,000 sentences from the results and asked professional translators specializing in patent translation to translate them into Japanese, without informing them that their translations would be used for tuning and testing SMT systems. Finally, the resulting set of 2,000 sentence pairs was randomly divided into development and test data respectively consisting of 1,000 English-Japanese claim sentence pairs.

4.2. Systems

In this experiment, we regard the implementation of phrase-based SMT in the Moses toolkit (Koehn et al., 2007) with distortion limit of six as the baseline. We examined each of our SSSS transfer, and pre-ordering modules and their combination over the baseline. For reference, we investigated the performance of phrase-based SMT with a larger distortion limit 20, as well as hierarchical phrase-based SMT.

Throughout the experiments, we used KenLM (Heafield et al., 2013) for training language models and SyMGIZA++ (Junczys-Dowmunt and Szal, 2010) for word alignment. We used the grow-diag-final method for obtaining phrase pairs. Weights of the models were tuned with n-best batch MIRA (Cherry and Foster, 2012) regarding BLEU (Papineni et al., 2002) as the objective. For each system, we performed weight tuning three times and selected for the test the setting that achieved the best BLEU on the development data.

4.3. Evaluation Metrics

Each system is evaluated using two metrics: BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010a). Although our primary concern in this experiment is the effect of long distance relationship, in general, n-gram based metrics such as BLEU alone do not fully illustrate it. RIBES is therefore used alongside BLEU.

RIBES is an automatic evaluation method based on rank correlation coefficients; RIBES compares the word order in the SMT translation output with those in the reference. Hence it readily depicts the effects of drastic rearrangement in sentence components that often occurs between distant languages. In fact, RIBES has shown high correlation with human evaluation in both English-to-Japanese and Japanese-to-English translation tasks including those in the PatentMT at the NTCIR-9 Workshop (Goto et al., 2011).

### Table 2. BLEU and RIBES scores for all systems

<table>
<thead>
<tr>
<th>ID</th>
<th>Settings</th>
<th>SSSS transfer</th>
<th>Pre-ordering</th>
<th>SMT</th>
<th>English-to-Japanese</th>
<th>Japanese-to-English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BLEU</td>
<td>RIBES</td>
<td>BLEU</td>
</tr>
<tr>
<td>P1</td>
<td></td>
<td>PB</td>
<td>d=6</td>
<td>23.9</td>
<td>43.9</td>
<td>21.4</td>
</tr>
<tr>
<td>P1’</td>
<td></td>
<td>PB</td>
<td>d=20</td>
<td>23.4 (-0.5)</td>
<td>49.1 (+5.2)</td>
<td>22.4 (+1.0)</td>
</tr>
<tr>
<td>H1</td>
<td></td>
<td>HPB</td>
<td></td>
<td>24.3 (+0.4)</td>
<td>53.4 (+9.5)</td>
<td>23.2 (+1.8)</td>
</tr>
<tr>
<td>P2</td>
<td>✔</td>
<td>PB</td>
<td>d=6</td>
<td>24.7 (+0.8)</td>
<td>67.9 (+24.0)</td>
<td>20.8 (-0.6)</td>
</tr>
<tr>
<td>P3</td>
<td>✔</td>
<td>✔</td>
<td>d=6</td>
<td>23.7 (-0.2)</td>
<td>55.1 (+11.2)</td>
<td>22.3 (+0.9)</td>
</tr>
<tr>
<td>P4</td>
<td>✔</td>
<td>✔</td>
<td>d=6</td>
<td>28.8 (+4.9)</td>
<td>74.9 (+31.0)</td>
<td>27.5 (+6.1)</td>
</tr>
</tbody>
</table>
4.4. Results

Table 2 summarizes the BLEU and RIBES scores for all systems, where the numbers in the brackets show the improvement over P1, the vanilla PBSMT system. The letter “d” in the SMT column denotes the distortion limit of the SMT decoder. In both English-to-Japanese and Japanese-to-English directions, the combination of SSSS transfer and pre-ordering, i.e., P4, substantially improved the translation quality in terms of BLEU and RIBES scores. While both SSSS transfer alone (P2) and pre-ordering (P3) alone also led to drastic increases of RIBES scores, they achieved only marginal improvement of BLEU scores. Thus the substantial BLEU improvement derived by their combination suggests that SSSS transfer also contributes to improving the performance of pre-ordering.

4.5. Analysis

Experimental results confirm that translation quality can be improved significantly by using our SSSS transfer, irrespective of the existence of the pre-ordering process and translation directions. In this section, we first explain how our initial issues, i.e., extreme lengths and sub-language-specific structures in claim sentences, are resolved by SSSS transfer and pre-ordering. Subsequently, we provide and in-depth analysis of the additional benefit of our SSSS transfer, i.e., making SMT inputs short. Finally, we discuss the different trends of the observed gains in the two translation directions.

Complementary contribution of SSSS transfer and pre-ordering: Figure 8 illustrates a typical sequence of example translations generated by the four configurations, P1 to P4, in our Japanese-to-English experiment. Throughout the figure, a labelled bracketing scheme is used to illustrate claim components. The contributions of SSSS transfer and pre-ordering are summarized as follows.

1. Contribution of SSSS transfer: The order of components is not changed from the input Japanese sentence in P1. However, in P2, with the introduction of SSSS transfer, the components are well arranged in the order of English. The entire translation can be better understood by properly generating the transitional phrase “comprising”. Regarding the translation quality of each component, P1 and P2 do not seem significantly different. In contrast, we obtain a better translation for the second element in P4 than in P3. This is an evidence that SSSS transfer improves pre-ordering effectively.

2. Contribution of pre-ordering: As already demonstrated in the previous work, pre-ordering techniques are effective in generating translations with a reasonable word order in the target language. In fact, the words in P3 are better arranged than in P1: the word order is closer to that of the English reference. However, from the viewpoint of sentence structure, the components are not arranged well, and somehow the preamble is generated twice. Conversely, explicitly teaching the sentence-level structure through SSSS transfer, i.e., as in P4, suppresses such an undesirable error. Furthermore, dividing the input into shorter components, results in the words in each component being properly reordered.

In summary, SSSS transfer and pre-ordering complement each other in generating translations that are natural both structurally and component-wise.
**Effects of shortening SMT inputs:** As seen above, pre-ordering works better on components obtained through SSSS transfer rather than on the entire input sentence. To estimate the shortening effect of SSSS transfer, we compared the distributions of lengths of the processing unit of the succeeding steps, i.e., the entire sentence for P1 and automatically identified claim components in P2. Figure 9 shows the cumulative ratio of original sentences and identified claim components in English and Japanese, respectively. For example, a point (20,70) on the graph indicates that the sentences having lengths between 1 and 20 comprise 70% of all the sentences. As clearly illustrated, SSSS transfer considerably shortened the input to pre-ordering and SMT. This not only makes SMT easier, but also improves the accuracy of syntactic parsing and pre-ordering. Table 3 shows the sentence-wise accuracy of the English parser invoked by our pre-ordering module, calculated on the basis of 100 sentences sampled randomly from the test set. The parse tree of each sentence is manually checked for correctness. The shorter sentences show higher accuracy. This means that SSSS transfer does contribute to improving pre-ordering accuracy in addition to transferring sublanguage-specific sentence structure.

<table>
<thead>
<tr>
<th>Input</th>
<th>[s [ELEM ベース管と、][ELEM 前記ベース管を周む少なくとも部分的に拡張されたスクリーンジャケットと][TRAN を含む][PURP 前記スクリーンジャケットが、前記ベース管上に位置決めされる前に拡張され、前記スクリーンジャケットのフィルタ層が、前記スクリーンジャケットの拡張により、前記スクリーンジャケットの外側シャラウドに接触する。]]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>[s [PREA 一个 well screen system,][TRAN comprising:][ELEM a base pipe][ELEM and an at least partially expanded screen jacket surrounding the base pipe.][PURP the screen jacket being expanded prior to being positioned on the base pipe, wherein a filter layer of the screen jacket contacts an outer shroud of the screen jacket due to expansion of the screen jacket.]]</td>
</tr>
<tr>
<td>P1 Baseline</td>
<td>[s [ELEM The base tube and][ELEM the base tube is extended to at least partially surround a screen jacket][PURP the screen jacket is positioned on the base tube before the expansion of the cooling jacket of the screen, the screen filter layer by an extension of the jacket of the cooling jacket of the screen outer shroud contact well screen.]]</td>
</tr>
<tr>
<td>P2 w/ SSSS transfer</td>
<td>[s [PREA A well screen system,][TRAN comprising:][ELEM a base pipe][ELEM and an at least partially expanded screen jacket surrounding the base pipe.][PURP the screen jacket being expanded prior to being positioned on the base pipe, wherein a filter layer of the screen jacket contacts an outer shroud of the screen jacket due to expansion of the screen jacket.]]</td>
</tr>
<tr>
<td>P3 w/ pre-ordering</td>
<td>[s [PREA A well screen system,][TRAN comprising:][ELEM a base pipe][ELEM and an at least partially expanded screen jacket surrounding the base pipe.][PURP the screen jacket being expanded prior to being positioned on the base pipe, wherein a filter layer of the screen jacket contacts an outer shroud of the screen jacket due to expansion of the screen jacket.]]</td>
</tr>
<tr>
<td>P4 Pipeline</td>
<td>[s [PREA A well screen system,][TRAN comprising:][ELEM a base pipe][ELEM and an at least partially expanded screen jacket surrounding the base pipe.][PURP the screen jacket being expanded prior to being positioned on the base pipe, wherein a filter layer of the screen jacket contacts an outer shroud of the screen jacket due to expansion of the screen jacket.]]</td>
</tr>
</tbody>
</table>

**Figure 8. Example Japanese-to-English translation:** The bracket information in the input, reference, P1 and P3 are not determined automatically. We indicated them for explanation purpose only.
Different trends for translation directions: In terms of RIBES score, pre-ordering improved Japanese-to-English translation substantially, while showing less improvement in the English-to-Japanese setting. We speculate that the difference lies in the difficulty of pre-ordering, and more specifically, in the difficulty of parsing sentences in the source language. As Japanese is a strictly head-final language, parsing sentences is easier than in English. Consequently, pre-ordering alone achieved almost the entire gain in the Japanese-to-English setting. Conversely, English sentences are much more difficult to parse than Japanese. As a result, the pre-ordering module can sometimes fail to bring the English word order close to that in Japanese. Nevertheless, as a result of SSSS transfer, which divides an input English sentence into shorter pieces, pre-ordering became more accurate, and the RIBES score was further improved.

5. Related Work

The quality of machine translation across distant languages has been improved as a result of the recent introduction of syntactic information into SMT (Collins et al., 2005; Quirk et al., 2005; Katz-Brown and Collins, 2008; Sudo et al., 2013; Hoshino et al., 2013; Cai et al., 2014; Goto...
et al., 2015). One of the promising avenues for further improvement appears to be the incorporation of sublanguage-specific information (Buchmann et al., 1984; Luckhardt, 1991). This is particularly important for translating formalized documents that tend to form sublanguage-specific document structures and sentence structures. In dealing with structures across close language pairs, an early study of sublanguage introduced the notion of flat trees which represents both source and target sentences using minimal depth structures for facilitating the transfer between the source and target structures (Buchmann et al., 1984). Much of the recent work relating to document and sentence structures between close languages focuses on structures centered on discourse connectives (Miltsakaki et al., 2005; Piter and Nenkova, 2009; Meyer et al., 2011; Hajlaoui and Popescu-Belis, 2012; Meyer et al., 2012) and on resolving the ambiguity of discourse connectives connecting structural components.

Conversely, when dealing with structures across distant language pairs, a more comprehensive approach is more appropriate. A wide range of research has been conducted in this direction. A study by Marcu et al. (2000) proposed a method for improving Japanese-to-English translation by transforming the source structure generated by a rhetorical structure theory (RST) parser, to the corresponding target structure. Some work in this direction has been conducted in translations across distant languages, in which the source text is parsed using an RST parser, and translation rules are automatically extracted from the source and target pair (Kurohashi and Nagao, 1994; Wu and Fung, 2009; Joty et al., 2013; Tu et al., 2013). There are also approaches of simplifying long sentences by capturing the overall structure of a sentence, or a group of sentences. The skeleton-based approach (Mellebeek et al., 2006; Xiao, 2014) attempts to extract the key elements/structure (or skeleton) from the input sentence using a syntactic parser. The divide-and-translate approach (Shinhori et al., 2003; Sudo et al., 2010; Hung et al., 2012) also makes use of syntactically motivated features, such as phrases and clauses, for extracting subcomponents to be translated by SMT. There are also studies on pattern translation (Xia et al., 2004; Murakami et al., 2009; Murakami et al., 2013) and sentence segmentation (Xiong et al., 2009; Jin and Liu, 2010) for dealing with long input sentences with complex structures.

Our approach is similar to the above models in the sense that it incorporates structural information into SMT, but differs in that it uses sublanguage-specific sentence structures, rather than syntactically motivated structures. This results in significant improvement in translation quality for the claim sublanguage using only a handful of rules.

6. Conclusion

In this paper, we described a method for transferring sublanguage-specific sentence structure for English-to-Japanese and Japanese-to-English patent claim translations. The experimental results show that our proposed method, a combination of SSSS transfer and pre-ordering based on syntactic parsing, achieved five point gains in BLEU scores, in both English-to-Japanese and Japanese-to-English directions. In addition, a substantial gain of more than 30 points in RIBES scores was observed in both SMT settings, indicating a significant contribution of SSSS transfer. We achieved these results with only a handful of SCFG rules.

Our proposed method successfully improved the translation of patent claims with quality comparable to that of the other parts of patent documents. In our future work, we will concentrate on the translation of independent claims which are the longest and most complex of claim sentences.
References


Japanese News Simplification: Task Design, Data Set Construction, and Analysis of Simplified Text

Isao Goto
goto.i-es@nhk.or.jp
Hideki Tanaka	tanaka.h-ja@nhk.or.jp
Tadashi Kumano
kumano.t-eq@nhk.or.jp
NHK Science & Technology Research Laboratories,
1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan

Abstract
In this paper we explore a Japanese news simplification task. We designed a Japanese news simplification task, constructed the data set for the task, and analyzed the manual simplification process. We designed the task focusing on sentence-level simplification, which is part of the process of manual simplification of Japanese news for non-native speakers. We constructed the data set consisting of Japanese news sentences and their corresponding simplified Japanese news sentences, and verified the effectiveness of the data set for automatic simplification by conducting preliminary experiments using phrase-based statistical machine translation. To reveal the processes behind manual simplification, such as simplification associated with word order (syntactic structure), we analyzed manually simplified Japanese news sentences.

1 Introduction
Simplified texts increase readability and understandability for non-native speakers and children. Simplified news texts are especially useful for daily living because news delivers information needed for life. There are a number of simplified texts available in English, such as Learning English provided by Voice of America and the BBC, which are multimedia sources of news and information geared toward learners, Simple English Wikipedia, and simplified technical English based on the ASD-STE100 standard. There is much research on automatic English simplification (Chandrasekar et al., 1996; Carroll et al., 1998; Petersen and Ostendorf, 2007; Zhu et al., 2010; Coster and Kauchak, 2011; Woodsend and Lapata, 2011; Wubben et al., 2012; Kauchak, 2013; Narayan and Gardent, 2014). To realize automatic simplification using statistical machine translation (SMT), a parallel corpus consisting of sentences and their corresponding simplified sentences is needed. As such, an English parallel corpus was constructed using English Wikipedia and Simple English Wikipedia (Coster and Kauchak, 2011). In contrast, there is less research on the simplification of languages other than English: Portuguese (Aluísio et al., 2008), Spanish (Bott and Saggion, 2011; Bott et al., 2012), Italian (Dell’Orletta et al., 2011), French (Seretan, 2012), German (Klaper et al., 2013), and Japanese (Inui et al., 2003; Moku and Yamamoto, 2012; Tanaka et al., 2013). In these studies, work focusing on Japanese simplification can be summarized as follows. Inui et al. (2003) collected paraphrase readability rankings by consulting with teachers at schools for the deaf, and trained a paraphrase ranking model from the collected rankings. Moku and Yamamoto (2012) performed the automatic simplification of official documents in Japanese. They concluded through their pre-experiment, however, that

http://www.asd-ste100.org/
their method was ineffective. Tanaka et al. (2013) reported on the Internet news service NEWS WEB EASY, which provides manually simplified Japanese news produced through the rewriting of Japanese news. Because the simplification is conducted manually by humans, automation becomes the key to solving the issue of efficiency. To our knowledge, there is currently no work on the automatic simplification of Japanese news texts using SMT.

In this paper, we explore a Japanese news simplification task. We design the task, construct the data set for it, then reveal the manual simplification process by analyzing the simplified text. In the process of manually simplifying Japanese news for the NEWS WEB EASY service (Tanaka et al., 2013), designed for foreigners living in Japan, there are two types of operations: article-level shortening for conciseness (especially when articles are long), and sentence-level simplification of expressions. News reporters mainly carry out article-level shortening for conciseness, while Japanese instructors mainly carry out sentence-level simplification of expressions (Section 2). We focus on the process of sentence-level simplification of expressions, which is thought as closer to being serviceable for practical use than automatic article-level shortening, and define this sentence-level process here as the simplification task (Section 3). We construct a data set of parallel sentences consisting of sentences from Japanese news prior to being initially simplified by the Japanese instructors, and the resulting simplified sentences (Section 4). Then, we conduct preliminary experiments on automatic simplification using the data set and phrase-based SMT to verify the effectiveness of the data set (Section 5). Additionally, we produce manually annotated word alignments between parallel sentences and reveal the processes of manual simplification, such as rewriting with a different word order (i.e., syntactic structure) by analyzing the annotated word alignments (Section 6). Our contributions are summarized as follows:

- We proposed a Japanese news simplification task, constructed a data set for the task, and verified the effectiveness of the data set through experiments on automatic simplification using phrase-based SMT.
- We revealed the processes of manual simplification, such as rewrites associated with word order (syntactic structure), which are needed to design effective automatic simplification.

2 Simplification in Easy Japanese News Service

In this section we describe the Japanese news simplification processes in the simplified Japanese (easy Japanese) news service called NEWS WEB EASY (Tanaka et al., 2013); our research target for this study. The service is offered by NHK. NEWS WEB EASY is an online service that provides easy Japanese news for foreigners living in Japan who have reached pre-intermediate-level Japanese. Easy Japanese news is mainly produced through two operations: (1) shortening original news articles for conciseness, especially when the articles are long, and (2) simplifying expressions. Long Japanese news articles can be a burden for non-native speakers to read. Japanese news articles can be long or short, but there are few long articles that feature easy to understand Japanese. When longer news articles are shortened, the resulting articles are sometimes one-half or one-third the original length. The degree of shortening depends on the original length. News reporters mainly shorten with the goal of conciseness, while designated Japanese instructors well-versed in easy Japanese news mainly simplify expressions found in the text. For each original article to be simplified, a news reporter and a Japanese instructor take turns rewriting the article. This rewriting process by the reporter and the Japanese instructor is repeated two or more times, as needed.3

2 These instructors have experience teaching Japanese to non-native speakers.
3 The percentage of original articles first rewritten by Japanese instructors before being rewritten by news reporters changed over time, as follows. In April 2012, when the NEWS WEB EASY test service started, the rate was 58%.
3 Sentence-Level Japanese News Simplification Task

Shortening news articles can be technically simulated to some extent using an automatic summarization technique. However, there are problematic issues related to the use of automatic summarization for actual proper news services. Namely, to shorten long articles into lengths of one-half or one-third of their original lengths, not only are repetitive sections removed, but some of the actual news content is also eliminated. Therefore, there is a risk that some of the news contents that should be delivered to the reader may be deleted. This is a significant risk in terms of accurately conveying news. Additionally, when news reporters shorten an article, the article is reorganized as needed, and this process requires sophisticated editing. For these reasons, it is considered difficult to obtain practical-level quality for news services through automatic summarization. In contrast, the operation Japanese instructors perform does not require shortening and centers mainly on sentence-level simplification of expressions.

Therefore, we divided these two operations, that is, article-level shortening of length and sentence-level simplification of expressions into two separate tasks. In this paper, we define the sentence-level simplification of expressions, which is performed by Japanese instructors, as the simplification task. Automating this operation is thought to be more practical for news services, because sentence-level rewrites do not require removing certain content and a statistical machine translation technique can simulate this operation to some extent. In this task, the input sentences for automatic simplification systems are the sentences prior to being simplified by the Japanese instructors, while the simplified reference sentences are the sentences that are the results of the initial simplification process performed by the Japanese instructors.

4 Data Set for Japanese News Simplification

We produced a data set for the Japanese news simplification task by constructing parallel sentences. These comprised sentences taken from Japanese news articles and their corresponding simplified articles. In this section, we describe the construction method and the constructed data set.

4.1 Construction Method

Here we explain how parallel sentences taken from Japanese news articles and their corresponding simplified Japanese sentences were constructed. In this paper, the sentences prior to being initially simplified by the Japanese instructors are called source sentences, and their corresponding output simplified sentences are called target sentences. Additionally, a source sentence and its corresponding target sentence(s) are called a parallel sentence pair; similarly, an article prior to being initially simplified by a Japanese instructor and its corresponding output simplified article are called a parallel article pair.

However, by May 2013, when the official NEWS WEB EASY service started, the rate had dropped to 6%. In September 2014, the rate was down to 1%. Because both Japanese instructors and news reporters began rewriting at the same time when the service was just beginning, approximately half of the news articles were first rewritten by Japanese instructors. As time went on, however, most of news articles came to be first rewritten by news reporters because it was discovered that it was more efficient to simplify expressions after articles had already been shortened.
Table 1: Noise in parallel sentences consisting of one source sentence and one or more target sentence(s)

<table>
<thead>
<tr>
<th>Number of sentence pairs</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without noise</td>
<td>2,012</td>
</tr>
<tr>
<td>With noise</td>
<td>873</td>
</tr>
</tbody>
</table>

Table 2: Number of sentences in each manually annotated sentence alignment

<table>
<thead>
<tr>
<th>Number of sentences (source–target)</th>
<th>Frequency</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–1</td>
<td>2,201</td>
<td>0.698</td>
</tr>
<tr>
<td>1–2</td>
<td>575</td>
<td>0.182</td>
</tr>
<tr>
<td>1–3</td>
<td>90</td>
<td>0.029</td>
</tr>
<tr>
<td>1–0</td>
<td>71</td>
<td>0.023</td>
</tr>
<tr>
<td>0–1</td>
<td>67</td>
<td>0.021</td>
</tr>
<tr>
<td>2–2</td>
<td>52</td>
<td>0.016</td>
</tr>
<tr>
<td>2–1</td>
<td>46</td>
<td>0.015</td>
</tr>
<tr>
<td>2–3</td>
<td>17</td>
<td>0.005</td>
</tr>
<tr>
<td>1–4</td>
<td>16</td>
<td>0.005</td>
</tr>
<tr>
<td>Others</td>
<td>20</td>
<td>0.006</td>
</tr>
</tbody>
</table>

Manual Extraction of Parallel Sentence Pairs for Test Data

We constructed test sentences and their simplified reference sentences by manual extraction to ensure reliability. We manually aligned source sentences with their corresponding target sentences in parallel article pairs. Parallel sentence pairs can be extracted using these sentence alignments. The objective of the simplified reference sentences is to evaluate the quality of automatically simplified sentences, as each test sentence is automatically simplified independently. Thus, we selected test sentences and their simplified reference sentences from parallel sentence pairs under the following conditions:

- One source sentence corresponds to one or more target sentences.
- We eliminate parallel sentence pairs for which major content in the source sentence is not included in its corresponding target sentence(s), or for which content that is not included in a source sentence is included in its corresponding target sentence(s). There are the following exceptions. When major news content in a source sentence is included in its corresponding target sentence(s), the omission of related detailed information is allowed. In the target sentences, additional information not dependent on context and which is always true is allowed.

The first condition is called the alignment condition, and the second condition is called the noise condition. Here we show an example of additional information that does not depend on context and is always true: “Aomori Prefecture” in a source sentence and “Aomori Prefecture in the Tohoku region” in its target sentence.

We manually aligned sentence pairs for 490 parallel article pairs. Among these article pairs, test sentences and their simplified reference sentences could be extracted from 485 article pairs under the abovementioned conditions. The rates for without noise (satisfying the noise condition) and with noise (not satisfying the noise condition) for the parallel sentences that satisfied the alignment condition are shown in Table 1. The rate for without noise was approximately 0.7.
### Requirements of Automatic Sentence Alignment

We use an automatic sentence alignment method for constructing training data because it reduces cost and allows us to automatically use news data produced daily. We therefore examined the requirements of automatic sentence alignment.

As explained in Section 2, news reporters mainly shorten news articles for conciseness, while Japanese instructors mainly simplify the expressions used within the text. However, these roles are not clearly divided. News reporters sometimes simplify parts of expressions, while Japanese instructors not only replace difficult words with simple words and divide sentences into simpler syntactic structures, but they also sometimes omit detailed content, add supplemental explanations, or change the sentence order in an article. Figure 2 shows an example of sentence alignments and a parallel article pair before and after being simplified by a Japanese instructor (sentence ID of 0 indicates title).

![Figure 2: Example of sentence alignments and a parallel article pair before and after being simplified by a Japanese instructor](image)

### Table 2: Results

<table>
<thead>
<tr>
<th>ID</th>
<th>Sentences before rewrite</th>
<th>Sentences after rewrite</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>サッカーのシティ初めワールドカップで奪うことにした。</td>
<td>日本が初めてワールドカップのサッカーのシティを奪うことにした。</td>
</tr>
<tr>
<td>1</td>
<td>日本のサッカー君は平成13年（2001年）に初登場。</td>
<td>日本のサッカー君は平成13年（2001年）に初登場。</td>
</tr>
<tr>
<td>2</td>
<td>また、5月のリーグ戦でスプリンターの出走も行う。</td>
<td>また、5月のリーグ戦でスプリンターの出走も行う。</td>
</tr>
<tr>
<td>3</td>
<td>リーグ戦での試合を行われるとき、どのチームが勝つか予測する。</td>
<td>リーグ戦での試合を行われるとき、どのチームが勝つか予測する。</td>
</tr>
<tr>
<td>4</td>
<td>しかし、今度ワールドカップでヤブを売ったことはありません。</td>
<td>しかし、今度ワールドカップでヤブを売ったことはありません。</td>
</tr>
<tr>
<td>5</td>
<td>また、平成13年（2001年）に初登場。</td>
<td>また、平成13年（2001年）に初登場。</td>
</tr>
<tr>
<td>6</td>
<td>さらに、即時データの変更も行う。</td>
<td>さらに、即時データの変更も行う。</td>
</tr>
<tr>
<td>7</td>
<td>そして、決戦トーナメントでも、ゴールの数を予想する。</td>
<td>そして、決戦トーナメントでも、ゴールの数を予想する。</td>
</tr>
<tr>
<td>8</td>
<td>このサッカー君は5月31日から売り始めます。</td>
<td>このサッカー君は5月31日から売り始めます。</td>
</tr>
</tbody>
</table>

---

*In the example in Figure 2, many expressions in the sentences on the left were already simplified by a news reporter. Reporters sometimes simplify expressions like these.*

*This value was calculated as follows. We removed target sentences that were not aligned with source sentences. We then projected source sentence IDs to their aligned target sentences. Here, we define the projected order of the target sentences as the order of sentences based on the projected IDs. When two target sentences had the same projected ID, the projected order follows the original target sentence order. We calculated Kendall’s τ between the projected order and the order of the target sentence order.*
Table 3: Number of sentences in each automatic sentence alignment

<table>
<thead>
<tr>
<th>Number of sentences (source–target)</th>
<th>Frequency</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–1</td>
<td>7,893</td>
<td>0.727</td>
</tr>
<tr>
<td>1–2</td>
<td>2,336</td>
<td>0.215</td>
</tr>
<tr>
<td>1–3</td>
<td>383</td>
<td>0.035</td>
</tr>
<tr>
<td>2–1</td>
<td>111</td>
<td>0.010</td>
</tr>
<tr>
<td>2–2</td>
<td>44</td>
<td>0.004</td>
</tr>
<tr>
<td>1–4</td>
<td>39</td>
<td>0.004</td>
</tr>
<tr>
<td>0–1</td>
<td>32</td>
<td>0.003</td>
</tr>
<tr>
<td>1–0</td>
<td>19</td>
<td>0.002</td>
</tr>
<tr>
<td>3–1</td>
<td>5</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 4: Precision and recall of automatic sentence alignment

<table>
<thead>
<tr>
<th>Number of sentences (source–target)</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whole</td>
<td>0.872</td>
<td>0.885</td>
</tr>
<tr>
<td>One–One or more</td>
<td>0.881</td>
<td>0.942</td>
</tr>
</tbody>
</table>

Automatic Sentence Alignment for Training Data

Source sentences and target sentences include many identical words, and the sentence order is almost identical. Therefore, we decided to use an alignment method using identical words and dynamic programming, which can efficiently estimate sentence alignments with consideration of sentence order so that sentence alignments do not include crossing alignments. We used Champollion (Ma, 2006), which is an implementation of such a method, for automatic sentence alignment. This method can treat alignments including two or more source or target sentences, sentence omission, and sentence addition. The number of sentences in each automatic sentence alignment for 1,559 parallel article pairs for training data are shown in Table 3.

Quality Evaluation of Automatic Sentence Alignments

We evaluated the quality of automatic sentence alignments using the 490 parallel article pairs that were manually annotated with sentence alignments. Precision and recall for all sentence alignments and for sentence alignments consisting of one source sentence and one or more target sentences are shown in Table 4. Here, the unit of sentence alignment is defined as one parallel sentence pair (e.g., one-to-one, one-to-two, two-to-one, or one-to-null).

Automatic simplification based on monolingual translation has the following characteristics. When output sentences include errors, the quality of the output sentences decreases compared with the input sentences. However, when input sentences are output without modifications, the quality of the output sentences does not decrease compared with the input sentences. Therefore, it is important for simplification to remove as much noise as possible from the training data.

4.2 Data Set for Evaluating Automatic Simplification

Here we describe the specifications of the constructed data set for evaluating automatic simplification. We constructed the data set using news archives from April 2012 to September 2014. We used news articles with edit histories showing simplification by Japanese instructors. In
keeping with practical use, we divided the data as follows: Data in the latest term were used as the test data, data in the term immediately prior to the test data term were used as the development data, and data in the term immediately prior to the development data term were used as the training data. The specifics of the data are shown in Table 5. The test data consist of 2,012 manually extracted source sentences with their corresponding simplified reference sentences. The development data consist of 723 parallel sentence pairs manually extracted in the same way as the test data. The training data consist of 10,651 automatically extracted parallel sentence pairs. The training data also include 1,559 news articles with full versions of edit histories in the training term.

5 Experiments on Sentence-Level Automatic Simplification

To confirm the effectiveness of the constructed data set, we conducted preliminary experiments on sentence-level automatic simplification using the data set. We conducted automatic Japanese news simplification as a monolingual translation task using phrase-based SMT, as Coster and Kauchak (2011) did.

5.1 Setup

We used MeCab\(^7\) for Japanese segmentation. Continuous Arabic numerals were merged to one word. We used the Moses implementation (Koehn et al., 2007) as the phrase-based SMT system. The translation model was trained using sentences that are 80 words or less. GIZA++ and grow-diag-final-and heuristics were used to obtain word alignments. To assist the word alignments for low frequency words, we added pairs of the same word to the training data when word alignments were estimated. This is because in the monolingual parallel sentences there are many words that should be aligned to the same words. We used a 5-gram language model that was trained using the target side of the training data. The SMT weighting parameters were tuned via MERT (Och, 2003) using the development data. We used distortion limits of 0 or 6 (default value), which limit the number of words for word reordering to a maximum number. We used the MSD bidirectional lexicalized reordering models of Moses (Koehn et al., 2005).

We compare the trained SMT system (MOSES) to the baseline that does not simplify input sentences and outputs the input sentences without any change (BASELINE). We evaluate using the test data of the data set.

5.2 Results and Discussion

We evaluated the simplification quality based on the automatic evaluation scores from the BLEU-4 (Papineni et al., 2002) and RIBES v1.02.4 (Isozaki et al., 2010), which are commonly used for evaluating translation quality. Percentage scores were used for these scores. RIBES is an automatic evaluation measure based on word-order correlation coefficients between reference sentences and output sentences. Evaluation results are shown in Table 6. Bold numbers indicate that values are significantly higher than the result of BASELINE in each evaluation measure. To assess this, we used the bootstrap resampling test at a significance level of $\alpha = 0.01$.

---

\(^7\)http://taku910.github.io/mecab/
Table 6: Results of sentence-level automatic simplification

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>RIBES</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASELINE</td>
<td>41.62</td>
<td>84.04</td>
</tr>
<tr>
<td>MOSES (distortion limit 0)</td>
<td>46.06</td>
<td>85.41</td>
</tr>
<tr>
<td>MOSES (distortion limit 6)</td>
<td>46.02</td>
<td>85.39</td>
</tr>
</tbody>
</table>

(Moses, 2004).

Moses (distortion limit 0), which learned simplification from the training data, obtained a BLEU score that is 4.4 points higher than that of BASELINE. This confirms the effectiveness of the data set for automatic simplification. Moses (distortion limit 0) also obtained a RIBES score that is 1.3 points higher than that of BASELINE. In an experiment on English simplification using Wikipedia (Coster and Kauchak, 2011), the improvement of the BLEU score was 0.5 points. The improvement of the BLEU score using our data set is higher than theirs, indicating that our data set had a larger effect than their English Wikipedia data set. One of the reasons for this larger effect is thought to be that more words in simplified Japanese news were rewritten than those in the Simple English Wikipedia data set because the BLEU score (41.62) of Japanese news BASELINE is lower than the BLEU score (59.37) of English Wikipedia BASELINE. The training data size is smaller than for parallel corpora often used in SMT experiments on translation between languages. When the training data are small, certain methods can improve the translation quality (Xiang et al., 2010; Irvine, 2013; Irvine and Callison-Burch, 2014). Such methods will be useful for our Japanese news simplification task.

When Moses (distortion limit 6), which allows phrase reordering, is compared with Moses (distortion limit 0), which does not allow phrase reordering, the automatic scores did not improve. However, this result does not ensure that phrase reordering is unnecessary because there is room for improvement of the RIBES scores. We cannot know what types of word reordering are needed based solely on these results. Therefore, in the next section we investigate what types of rewrites are needed for simplification.

6 Analysis of Manually Simplified News Sentences

To reveal what types of rewrites are needed for automatic Japanese news simplification, we analyzed what types of rewrites, such as those associated with word order (i.e., syntactic structure), were conducted by Japanese instructors in the manual simplification processes. For this analysis, we produced manually annotated word alignments for 50 parallel article pairs. The 50 article pairs include 309 source sentences and 530 target sentences. As explained in Section 4.1, news reporters sometimes simplify expressions, despite it not being their main role. Thus, when analyzing the simplification of expressions by comparing sentences before and after simplification by Japanese instructors, if we use sentences that were rewritten by news reporters as pre-simplification sentences, then parts of the simplification process may go undetected. This is because it is possible that a certain degree of expression simplification has already been conducted before being simplified by the Japanese instructors. Therefore, to exhaustively detect the simplification of expressions, we only used simplified news articles that Japanese instructors had rewritten prior to being rewritten by news reporters. Here, articles prior to being rewritten by Japanese instructors are called ORG, and articles that have been rewritten by such instructors are called EASY.

As explained in footnote 3, recent articles were first rewritten mostly by news reporters and Japanese instructors first rewrote around half of the articles produced in the early phase. We wanted to conduct this analysis independently of the construction of the data set because we can then start analyzing when small data become available. For these reasons, we selected the 50 article pairs from the data produced in the early phase; that is, the 50 pairs are not a subset of the manually sentence-aligned article pairs in the data set described in Section 4.
Table 7: Rewrite categories associated with word order

<table>
<thead>
<tr>
<th>Category</th>
<th>Frequency</th>
<th>Reordering distance type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing adnominal clauses to sentences</td>
<td>39</td>
<td>global</td>
</tr>
<tr>
<td>Reordering case elements</td>
<td>21</td>
<td>global</td>
</tr>
<tr>
<td>Reversed relation of modification</td>
<td>13</td>
<td>local</td>
</tr>
<tr>
<td>Complement for sentence splitting</td>
<td>13</td>
<td>global</td>
</tr>
<tr>
<td>Changing case of case elements</td>
<td>10</td>
<td>global</td>
</tr>
<tr>
<td>Indicating relation of continuous nouns</td>
<td>8</td>
<td>local</td>
</tr>
<tr>
<td>Changing compound nouns</td>
<td>7</td>
<td>local</td>
</tr>
<tr>
<td>Changing part of speech</td>
<td>7</td>
<td>local</td>
</tr>
<tr>
<td>Adnominal clause modifying formal nouns</td>
<td>5</td>
<td>global</td>
</tr>
<tr>
<td>Verbalizing nouns</td>
<td>4</td>
<td>global</td>
</tr>
<tr>
<td>Changing quantity expressions</td>
<td>3</td>
<td>global</td>
</tr>
<tr>
<td>Extraction of difficult expressions</td>
<td>3</td>
<td>global</td>
</tr>
<tr>
<td>Moving from EOS to BOS</td>
<td>3</td>
<td>global</td>
</tr>
<tr>
<td>Changing clause to noun modifier</td>
<td>2</td>
<td>local</td>
</tr>
<tr>
<td>Others</td>
<td>66</td>
<td>global/local</td>
</tr>
</tbody>
</table>

We then analyzed the simplifications associated with word order along with those not associated with word order.

6.1 Simplification Associated with Word Order

We categorized the simplification with respect to its association with word order for the sentences in the 50 article pairs. We disregarded expressions that were too largely summarized or too largely reorganized. Rewrite categories associated with word order are shown in Table 7. For the item of reordering distance type, “local” represents word reordering in a phrase pair or the reordering of contiguous phrases, and “global” represents longer word reordering compared with local, or the duplication of words. Explanations and examples of each category are shown in Appendix A. These results indicate that the most frequent rewrite type with word reordering is the extraction of adnominal clauses to become independent sentences. When adnominal clauses are extracted to become independent sentences, the syntactic structures of the resulting sentences become simpler. Thus, the effect of the extraction on readability is thought to be large. The second most frequent type, reordering case elements, does not result in a reduction in the complexity of syntactic structure. Thus, the effect of extraction of adnominal clauses on readability is thought to be larger than that of reordering case elements. Rewrites with local reordering also do not reduce the complexity of syntactic structures in many cases. From these analyses, the extraction of adnominal clauses to become independent sentences was found to be the most frequent and effective type of simplification with word reordering.

Although it is possible for phrase-based SMT to perform rewrites with local word reordering, converting adnominal clauses into independent sentences is difficult for phrase-based SMT because it requires long-distance word reordering and the duplication of the words modified by the adnominal clauses. Therefore, we believe that the following two-step conversion is suitable for use in automatic simplification. First, conversion with long-distance word reordering or word duplication, such as extracting adnominal clauses to become independent sentences, is conducted using syntactic structures and conversion rules. Second, sentences are simplified using phrase-based SMT.\(^9\)

\(^9\)The idea of splitting the process into two steps is the same as that of pre-ordering in SMT (Xia and McCord, 2004).
Table 8: Rate of tokens (words) in EASY that were unchanged, changed, or added

<table>
<thead>
<tr>
<th></th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unchanged (identical) tokens</td>
<td>0.56</td>
</tr>
<tr>
<td>Changed (different) tokens</td>
<td>0.37</td>
</tr>
<tr>
<td>Added tokens</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 9: Changing rates for passive voice expressions and causative expressions

<table>
<thead>
<tr>
<th>Change rate (Frequency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passive voice to active voice</td>
</tr>
<tr>
<td>Causative to non-causative</td>
</tr>
</tbody>
</table>

Table 10: Number of sentences produced from one sentence in simplification

<table>
<thead>
<tr>
<th>Number of sentences (source–target)</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–1</td>
<td>0.40</td>
</tr>
<tr>
<td>1–2</td>
<td>0.46</td>
</tr>
<tr>
<td>1–3</td>
<td>0.11</td>
</tr>
<tr>
<td>1–4 or more</td>
<td>0.02</td>
</tr>
</tbody>
</table>

6.2 Simplification Not Associated with Word Order

We also analyzed the word rewrite rates, the rewrite rates of passive voice and causative expressions, and sentence splitting.

We checked the rates of tokens (words) in EASY that were unchanged, changed, or added by Japanese instructors. The results are shown in Table 8. The results indicate that 44% of the tokens in EASY were either changed or added. We also checked the rate of omitted tokens in ORG. It was 0.08.

We checked the rate of passive voice expressions that were changed into active voice expressions, as well as the rate of causative expressions that were changed into non-causative expressions. The results are shown in Table 9. It was found that most of the passive voice expressions were changed into active voice expressions, and that most of the causative expressions were changed into non-causative expressions.

The number of sentences produced from a single sentence during simplification are shown in Table 10. Approximately half of the sentences were split into two separate sentences. The most frequent cause of sentence splitting was the changing of continuous clauses into independent sentences without word reordering, and the second most frequent was the changing of the above-mentioned adnominal clauses into independent sentences. The rate of sentence splitting is larger than the rate shown in Table 2. The main reason for this result is thought to be the fact that the pre-simplification sentences were different. The current analysis used original sentences as the pre-simplification sentences, whereas the analysis of Table 2 mainly used sentences that

---

10 We used MeCab with an IPA dictionary as the morphological analyzer.
11 Expressions that do not contain main content may be dropped. [E.g. 20] in Appendix A is such an example.
12 The lead sentence, which is the first sentence in a news article that describes a summary of the news, may be dropped because its content overlaps with the text of the main body. To check the rate of dropped words under the condition in which the reason for deletion was not overlapping, we checked the rate of omitted tokens using all sentences except the lead sentences.
13 We only checked causative expressions in which the base forms were *seru* or *saseru* and the part of speech was verb–postfix.
14 [E.g. 6] in Appendix A is an example of passive voice being changed into active voice. [E.g. 1] in Appendix A is an example of causative being changed into non-causative.
had already been rewritten to some degree by news reporters as the pre-simplification sentences.

7 Conclusion

We designed a Japanese news sentence-level simplification task and constructed a Japanese news simplification data set, which consisted of Japanese news sentences and their corresponding simplified sentences. This is, to the best of our knowledge, the first time a parallel corpus consisting of sentences sourced from Japanese news and their simplified Japanese counterparts has been constructed. We verified the effectiveness of the constructed data set through preliminary experiments on automatic simplification using phrase-based SMT, and we confirmed that our result was 4.4 BLEU points higher than that of the baseline, which does not change the input sentences. Additionally, we produced manually annotated word alignments between parallel sentences to analyze the human operation of simplifying expressions, and provided what types of rewrites, such as rewrites associated with word order, were conducted in the manual simplification processes. The constructed data set and the knowledge obtained by our analysis of manual simplification will be useful for future research on Japanese news simplification and will serve as assistance in the production of simplified Japanese news.

References


Appendix A. Explanations and Examples of Each Category in Table 7

Here we explain the categories associated with word order in Table 7 and show their examples. Red and blue are used for the parts reordered based on the factor of each category.

**Changing adnominal clauses to sentences** This refers to cases in which adnominal clauses are extracted as independent sentences.

[E.g. 1] NASA は去年 11 月に打ち上げた火星探査機「キュリオシティ」を日本時間の来月 6 日、午後 2 時半過ぎに火星に着陸させます。

NASA は去年 11 月に「キュリオシティ」という火星探査機を打ち上げました。「キュリオシティ」は、日本時間の来月 6 日、午後 2 時半過ぎに、火星に着きます。

[E.g. 2] この遺伝子をマウスの脳の記憶などをつかさどる海馬という部分に大量に組み込みました。

その遺伝子をマウスの脳の中の海馬という部分にたくさん入れました。海馬は記憶などをコントロールする働きがあります。

**Reordering case elements** This refers to cases in which the order of the case elements is changed.

[E.g. 3] 富士山が大規模に噴火した場合、山梨県は ⋯

山梨県は、富士山が大規模に噴火した場合 ⋯

**Reversed relation of modification** This refers to cases in which the word reordering is caused by changing words into other words with reversed relations of modification.

[E.g. 4] 半年余りで ⋯ 約半年で

**Complement for sentence splitting** This refers to cases in which the case elements are replicated to complement split sentences when continuous clauses are changed into independent sentences.

[E.g. 5] この有料サービスは、 ⋯ の現在地を地図で把握できるというもので、10日から運用が始まりました。

この有料サービスは、 ⋯ が今いる場所を地図で知ることができるというもので、このサービスは10日から始めました。

**Changing case of case elements** This refers to cases in which the word reordering is caused by changing the case of the case elements.
indicating relation of continuous nouns  This refers to cases in which the word reordering is caused by indicating the relations of continuous nouns.

changing compound nouns  This refers to cases in which the word reordering is caused by changing compound nouns.

changing part of speech  This refers to cases in which the word reordering is caused by changing the part of speech of words and their modifying points.

adnominal clause modifying formal nouns  This refers to cases in which the verbs in adnominal clauses modifying formal nouns move backward.

verbalizing nouns  This refers to cases in which the word reordering is caused by verbalizing nouns (sahen nouns).

changing quantity expressions  This refers to cases in which the word reordering is caused by changing quantity expressions.

extraction of difficult expressions  This refers to cases in which difficult expressions are extracted and explained as independent sentences.

moving from eos to bos  This refers to cases in which parts of expressions at the end of sentences move to the beginning of the sentence.
Others  This refers to cases for which categorization is difficult because rewrites are complex or they require background information, or when the frequency of the category is one.

[E.g. 19]  IT企業の間では、この分野を強化する動きが広がっています。
　・ 地図に力を入れる IT企業が増えています。

[E.g. 20]  電力会社が提供する需給状況のデータに基づいて、電力需要が少なく価格が安い時間帯に
　・ 電力が足りている時間は、電力会社のデータから分かります。
Learning Bilingual Distributed Phrase Representations for Statistical Machine Translation

Chaochao Wang  
chaochaowang@foxmail.com

Deyi Xiong *  
dyxiong@suda.edu.cn

Min Zhang  
minzhang@suda.edu.cn

Soochow University, Suzhou, China

Chunyu KIT  
ctckit@cityu.edu.hk

Department of Linguistics and Translation, City University of Hong Kong

Abstract

Following the idea of using distributed semantic representations to facilitate the computation of semantic similarity between translation equivalents, we propose a novel framework to learn bilingual distributed phrase representations for machine translation. We first induce vector representations for words in the source and target language respectively, in their own semantic space. These word vectors are then used to create phrase representations via composition methods. In order to compute semantic similarity of phrase pairs in the same semantic space, we project phrase representations from the source-side semantic space onto the target-side semantic space via a neural network that is able to conduct nonlinear transformation between the two spaces. We integrate the learned bilingual distributed phrase representations into a hierarchical phrase-based translation system to validate the effectiveness of our proposed framework. Experiment results show that our method is able to significantly improve translation quality and outperform previous methods that only use word representations or linear semantic space transformation.

1 Introduction

Distributional semantic models provide a means to represent meanings of words under the assumption that words occurring in similar contexts tend to have the same meanings (Harris,1968). Various distributed representations have been successfully applied to various monolingual natural language processing tasks, such as word sense discrimination (Clark and Pulman,2007) and thesaurus compilation (Yang and Powers,2008). In this paper, we explore how to learn semantic representations of bilingual phrases, rather than monolingual words, in the context of statistical machine translation (SMT) to facilitate the computation of semantic similarity between translation equivalents at the phrase level. We also study whether semantic similarity scores calculated in terms of bilingual distributed phrase representations are complementary to phrase translation probabilities estimated by the conventional counting method in SMT Koehn et al. (2003).

Very recently we have witnessed some studies on learning bilingual distributed representations for SMT. Mikolov et al. (2013b) train two neural network (NN) based models to learn word embeddings in the source and the target language, respectively, and then map the embeddings from the source to the target language space using a transformation matrix that is learned

*Corresponding author
by minimizing the mapping cost on all word pairs. Zou et al. (2013) introduce bilingual word embeddings into phrase-based machine translation: Word representations are first learned from language A via an NN-based model, word embeddings in the parallel language B are then initialized according to A’s embeddings and word alignments between A and B, and the final word representations of B are obtained by a further training process that optimizes a combined objective on bilingual data. Gao et al. (2013) extend distributional representations from the word level to the phrase level, adopting a fully connected neural network to transfer bag-of-words vector representations of raw phrases (in the source or the target language) to distributional representations in a language-independent low-dimensional semantic space and having the parameters of the neural network jointly learned with the feature weights of the log-linear model of phrase-based SMT.

Partially inspired by these previous studies, we propose a new framework to learn bilingual distributed phrase representations for SMT via semantic composition and bilingual projection. Our current work differs from the previous ones distinctively in several aspects.

- We learn bilingual phrase representations, instead of representations at the word level Mikolov et al. (2013b); Zou et al. (2013), so as to keep consistency with the SMT that uses phrases rather than words as basic translation units.

- We learn phrase representations from distributed word representations via semantic composition, instead of from raw phrases Gao et al. (2013) in order to avoid the data sparseness issue of directly learning phrase representations from data. Particularly, we empirically compare two different composition methods in our framework, namely, weighted vector addition (Mitchell and Lapata, 2008) and recursive autoencoder Socher et al. (2011).

- Rather than jointly learning phrase representations with feature weights of the log-linear model of SMT Gao et al. (2013), we take a loose coupling strategy to simplify the learning process. We adopt a neural network to project phrase representations from the source onto the target language semantic space, in separation from the process of feature weight tuning in SMT.

- Rather than capturing only the linear transformation between the source and target language semantic space Mikolov et al. (2013b), our neural network for the bilingual projection can model both linear and nonlinear transformation between these two semantic spaces. We hope that the nonlinear transformation can better model the bilingual projection.

We integrate the learned bilingual distributed phrase representations into a hierarchical phrase-based SMT system (Chiang, 2007) by calculating semantic similarity scores of bilingual phrases in terms of their representations. We also empirically compare combinations of learning methods for word representations, and phrase composition methods as well as bilingual projection strategies. Experiments on Chinese-to-English translation show that our best results outperform the baseline by 0.53 BLEU points, indicating the effectiveness of our approach.

The rest of the paper is organized as follows. Section 2 describes how we obtain word representations for the source and target language and vector representations of phrases via two different composition methods. Section 3 presents a nonlinear neural network to learn bilingual phrase representations by projecting source language phrase vectors onto the target language space and Section 4 introduces the integration of bilingual representations into SMT. Section 5 elaborates our large-scale experiments on Chinese-to-English translation and analyzes experimental results. Section 6 discusses related work in relation to ours and Section 7 concludes the paper with future directions of research.
2 Distributed Phrase Representation Acquisition via Semantic Composition

In this section, we introduce semantics-based vector representations of words and vector representations of phrases via two different composition methods.

2.1 Word Representations

Vectors of words are basic elements in our bilingual phrase representation learning framework. We employ two different models, namely a point-wise mutual information (PMI) based vector space model (Pado and Lapata, 2007) and a neural language model Mikolov et al. (2013a), to derive word vectors.

**PMI-based vector space word representations** Vector space model provides an elegant way to represent the meaning of a word: each element in its vector denotes a degree that measures how frequently it co-occurs in a predefined context window with every other word in the vocabulary in question. A well-known measure for this is PMI, which estimates the strength of the relationship between a context word $c$ and a target word $t$ as follows:

$$pmi(c, t) = \log \frac{p(c, t)}{p(c)p(t)}$$ (1)

In order to get around certain unavoidable frequency bias, we use positive point-wise mutual information (PPMI) (Turney and Pantel, 2010) to calculate the elements in a word. It is defined as:

$$ppmi(c, t) = \begin{cases} pmi(c, t) & \text{if } pmi(c, t) > 0 \\ 0 & \text{otherwise} \end{cases}$$ (2)

**Neural word representations** Mikolov et al. (2013a) introduce an efficient neural language model to learn high-quality word embeddings from extremely large amounts of raw texts. We adopt their approach for learning word embeddings. After training the neural language model, we can obtain a word embedding matrix $M \in \mathbb{R}^{n \times |V|}$, where each word in the vocabulary $V$ corresponds to a vector $v \in \mathbb{R}^n$ with $n$ to denote vector size. Given this, the vector representation of the word assigned with index $i$ in $V$ can be retrieved simply by extracting the $i^{th}$ column of $M$.

2.2 Composition Methods

Once having obtained vector representations for words, we can use them to construct those for phrases via various composition methods as phrases are composed of words. We explore two composition methods: one based on simple vector addition (Mitchell and Lapata, 2008) and the other on a recursive autoencoder that takes the inner structure of a phrase into account Socher et al. (2011).

**Weighted vector addition** Given a phrase $p$ that consists of two words $w_1$ and $w_2$, we obtain the vector $\vec{p}$ from its word vectors $\vec{w}_1$ and $\vec{w}_2$ by the following weighted vector addition:

$$\vec{p} = \alpha \vec{w}_1 + \beta \vec{w}_2$$ (3)

where $\alpha$ and $\beta$ are weights denoting the relative importance of each word in the composition. For a phrase with multiple words $p = (w_1, w_2, ..., w_n)$, we can use in a similar way to obtain the vector for $p$ by summing over vectors of all its words,

$$\vec{p} = \sum_{i=1}^{n} \lambda_i \vec{w}_i$$ (4)
Figure 1: *The architecture of a recursive autoencoder, where the nodes with black dots are input word (or phrase) vectors and the nodes with circles are reconstructed vectors for computing reconstruction errors.*

Figure 2: *The architecture of bilingual projection neural network that projects vector representations of source phrases to the semantic space of target language.*

Although weighted vector addition is a simple way for composition, it has proven effective in many tasks (Kartsaklis, 2014). In our task, however, it cannot model word positions in a phrase. Therefore we only use it as a baseline to compare against a more advanced method: recursive autoencoder.

**Recursive autoencoder (RAE)**  
RAE is a neural network that can learn representations for large linguistic expressions such as phrases and sentences in a bottom-up fashion along a tree structure. Normally, word vectors learned via a distributional method can be input as leaf nodes of RAE. Figure 1 presents an illustration to visualize the architecture of RAE. Given a binary branch \( p \rightarrow c_1 c_2 \) where a child node is either a leaf or nonterminal node, the representation of \( p \) can be calculated as:

\[
\overrightarrow{p} = f(W[\overrightarrow{c_1}; \overrightarrow{c_2}] + b)
\]  

(5)

where \([\overrightarrow{c_1}; \overrightarrow{c_2}]\) denotes the combination of the two child vectors, \(W\) and \(b\) are model parameters, and \(f\) is an element-wise activation function such as *sigmoid*. This will be used to further compute representations for larger structures. In order to judge how appropriately a parent vector computed this way can represent its children, we can reconstruct the children in a reconstruction layer as:

\[
[\overrightarrow{c_1'}; \overrightarrow{c_2'}] = W'\overrightarrow{p} + b'
\]  

(6)

For each nonterminal node, we compute the Euclidean distance between its original child vectors and the reconstructed vectors as the reconstruction error of the node according to the following equation:

\[
E_{rec}([\overrightarrow{c_1}; \overrightarrow{c_2}]) = \frac{1}{2} \| [\overrightarrow{c_1}; \overrightarrow{c_2}] - [\overrightarrow{c_1'}; \overrightarrow{c_2'}] \|^2
\]  

(7)

The parameters of an RAE can be learned by minimizing the reconstruction error over the entire tree.
In this paper, we adopt a greedy unsupervised RAE that is proposed in Socher et al. (2011) as an extension to the standard RAE described above. The unsupervised RAE can learn not only the representation of a phrase or sentence but also their tree structures in a greedy manner.

3 Learning Bilingual Phrase Representations

We use the methods introduced above to obtain phrase representations for the source and target language respectively, and adopt a nonlinear bilingual projection neural network to project the phrase representations in the source language onto the semantic vector space of the target language so as to calculate similarity scores of bilingual phrases in the same semantic space. The general architecture for this work is presented in Figure 2.

The adopted neural network for projection is a fully connected neural network with only one hidden layer. The projection can be formulated in the following equation:

$$\vec{p} = \text{sigmoid}(W_2(\text{sigmoid}(W_1 \vec{x} + b_1)) + b_2)$$

(8)

where $W_1$ is the projection matrix from the input layer to the hidden layer, $W_2$ is the projection matrix from the hidden layer to the output layer, $b_1$ and $b_2$ are bias terms. In order to calculate the weights of the network, we need to calculate the squared error function as follows:

$$e = \frac{1}{2} \sum_i (t_i^m - p_i^m)^2$$

(9)

where $p_i^m$ is the vector calculated by the neural network according to the Eq.(8) and $t_i^m$ the real vector of the corresponding target phrase. The weights can be trained via backpropagation by minimizing the error on the set of collected training instances $\{(\vec{s}, \vec{t})\}_n$ where $\vec{s}$ and $\vec{t}$ are vectors of the source and target side of a phrase pair $(s, t)$. If we do not use any hidden layer in the projection neural network, the degenerated neural network will exactly learn Mikolov et al. (2013a)’s linear transformation matrix. Adding a hidden layer with nonlinear activation functions, we enable our projection neural network to model the nonlinear transformation between the semantic spaces of the source and target language. We will empirically compare the nonlinear against the linear projection in Section 5. Once the projection neural network is trained, we can learn projected representations of source phrases in the target semantic space using this neural network.

4 Integrating Bilingual Representations into SMT

A straightforward way to integrate bilingual phrase representations into SMT is to calculate semantic similarity between representations of this kind for translation equivalents. Given a phrase pair $(s, t)$, let $(\vec{s}, \vec{t})$ denote their vector representations on the source and the target language semantic space and $(p(\vec{s}), \vec{t})$ the learned bilingual distributed phrase representations, where $p(\vec{s})$ is the projected vector representation of source phrase $s$ obtained by our projection neural network as presented above. The semantic similarity between $s$ and $t$ can then be calculated as follows:

$$\text{Sim}(p(\vec{s}), \vec{t}) = \frac{p(\vec{s}) \bullet \vec{t}}{\|p(\vec{s})\| \times \|\vec{t}\|}$$

(10)

Given a source sentence $c$, we can build a new semantic similarity model based on our learned bilingual phrase representations according to the following equation:

$$M_{\text{Sim}} = \sum_{(s, t) \in P} \text{Sim}(p(\vec{s}), \vec{t})$$

(11)
Table 1: Results of integrating word representations acquired by the two methods integrated into hierarchical phrase-based SMT

<table>
<thead>
<tr>
<th>System</th>
<th>NIST 06</th>
<th>NIST 08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>30.23</td>
<td>23.21</td>
</tr>
<tr>
<td>PPMI</td>
<td>30.37</td>
<td>23.36</td>
</tr>
<tr>
<td>Neural</td>
<td>30.46</td>
<td>23.40</td>
</tr>
</tbody>
</table>

where \( P \) denotes all possible phrase pairs that are in use to translate the source sentence \( c \) and have distributed vector representations. The semantic similarity can be used as a feature in a log-linear model and can also be integrated into any SMT system that uses bilingual phrase pairs during decoding. In this paper, we integrate this new model into a hierarchical phrase-based SMT system without loss of generality.

Rules in the hierarchical phrase-based SMT can be classified into two types: 1) phrase rules that only contain terminals and 2) non-terminal rules with at least one non-terminal. For phrase rules, the similarity score can be easily calculated according to Eq. (10) in a preprocessing step. As for non-terminal rules, we compute their similarity scores in two steps. Let us take a specific non-terminal rule \( X \rightarrow <X_{1}>X_{2}>L_{X} > \) as an example to show how we compute their similarity. First, we can find phrase pairs ( "选举 委员会" , "the election committee") and ( "举行 选举" , "hold election") via word alignments. The similarity values of these two phrase pairs are estimated according to Eq. (10). In order to ensure decoding speed, the semantic similarities of these phrases are also calculated in a preprocessing step so that they can be quickly retrieved during decoding. Second, we sum up all similarity values, including the similarity values of phrases within nonterminals \( X_{1} \) and \( X_{2} \), according to Eq. (11) by means of dynamic programming.

5 Experiments

We have carried out a number of experiments on Chinese-to-English translation to validate the effectiveness of the proposed framework for learning bilingual phrase representations. Various combinations of different word representation models, composition methods, and projection strategies presented above are tested. Particularly, we intend to explore answers to the following questions:

- Which word representation is better, PMI-based vector space representation or neural representation?
- Would phrase representations be better than word representations when used to calculate semantic similarity scores? Furthermore, would RAE provide more efficient phrase representations than weighted vector addition?
- Is it necessary to project phrase representations in a non-linear fashion?

5.1 Experiment Setup

Our baseline system is hierarchical phrase-based system (Chiang,2007), where translation candidates are scored by a set of features. Our training data consists of 4.1M sentence pairs with 98.9M Chinese words and 112.6M English words from LDC corpora, including LDC 2003E07, LDC 2003E14, LDC 2004E12, LDC 2004T07, LDC 2005T06 and LDC 2005T10. We use the NIST evaluation set of 2005 (NIST 05) as development set, and sets of NIST 06/NIST 08 as our test sets.
Table 2: Results of integrating word and phrase representations into hierarchical phrase-based SMT with “+” and “∗” to mark the statistically significant performance improvement over the baseline with $p < 0.05$ and $p < 0.01$

<table>
<thead>
<tr>
<th>System</th>
<th>NIST 06</th>
<th>NIST 08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>30.23</td>
<td>23.21</td>
</tr>
<tr>
<td>Word Representations</td>
<td>30.46</td>
<td>23.40</td>
</tr>
<tr>
<td>Phrase Representations</td>
<td>weighted vector addition</td>
<td>30.59</td>
</tr>
<tr>
<td></td>
<td>RAE</td>
<td>30.76$^*$</td>
</tr>
</tbody>
</table>

Table 3: Comparison of linear and nonlinear projection, with “+” and “∗” to mark the statistically significant performance improvement over the baseline with $p < 0.05$ and $p < 0.01$.

<table>
<thead>
<tr>
<th>System</th>
<th>NIST 06</th>
<th>NIST 08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>30.23</td>
<td>23.21</td>
</tr>
<tr>
<td>Linear Projection</td>
<td>30.48</td>
<td>23.42$^+$</td>
</tr>
<tr>
<td>Nonlinear Projection</td>
<td>30.76$^*$</td>
<td>23.51$^+$</td>
</tr>
</tbody>
</table>

Word alignments of training data were obtained by running GIZA++ (Och, 2003b) in both directions of our bilingual language source and applying refinement rule grow-diag-final-and Koehn et al. (2003). A 4-gram language model was trained on the Xinhua section of Gigaword by SRILM toolkit Stolcke et al. (2002). We also extracted SCFG rules from the word-aligned training data. The translation performance was measured by case-insensitive BLEU Papineni et al. (2002). We used minimum error rate training (MERT) (Och, 2003a) to tune the log-linear feature weights. As MERT is normally unstable, we ran the tuning process three times for all our experiments and presented the average BLEU scores on the three MERT runs as suggested by Clark et al. (2011).

The open source toolkit DISSECT\(^1\) was applied to obtain PMI-based vector space word representations with a context window of 5 words, and Word2Vec\(^2\) to acquire neural word representations, with each word represented as a 50-dimensional vector. When adopted Word2Vec, we just set the context window of size 5 and using continuous bag-of-words model. DISSECT was also adpoted to train weights in semantic composition of weighted vector addition. Unsupervised greedy RAE was trained in the way following Socher et al. (2011). In the bilingual projection neural network, 50 hidden units were used in the hidden layer.

5.2 PMI-Based Word Representations vs. Neural Word Representations

Our first series of experiments were carried out to compare PMI-based vector space word representations obtained by DISSECT against neural word representations obtained by Word2Vec. Note that we did not perform semantic composition in this series of experiments as we focus on word representations. Source word vector representations were projected onto the target semantic space via the projection neural network described in Section 3. Experimental results are presented in Table 1, from which we find both PMI-based vector space and neural word representations can improve translation quality in terms of BLEU. Since vector space representations obtained by neural word representations are better than PMI-based word representations, we used neural word representations in experiments hereafter.

\(^1\)http://clic.cimec.unitn.it/composes/toolkit/index.html
\(^2\)https://code.google.com/p/word2vec/
### Example 1

<table>
<thead>
<tr>
<th>Source</th>
<th>兰州物价局就牛肉面限价作出解释：只因涨幅过大。</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Lanzhou Price Bureau gives explanation of price controls on beef noodles: It is only because the raises have been too large.</td>
</tr>
<tr>
<td>Baseline</td>
<td>Lanzhou explained beef noodles reduce: only because of the excessive increase.</td>
</tr>
<tr>
<td>NWR</td>
<td>Lanzhou explained that beef noodles reduce only because of excessive price.</td>
</tr>
<tr>
<td>PRR</td>
<td>Lanzhou gives explanation of beef noodles reduce: only because of the excessive raises.</td>
</tr>
</tbody>
</table>

### Example 2

<table>
<thead>
<tr>
<th>Source</th>
<th>高收入是许多人从事小时工兼职的重要原因之一。</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>High wages are one of the major reasons for many people to get second jobs as hourly workers.</td>
</tr>
<tr>
<td>Baseline</td>
<td>High income many people engage in hourly workers outside one of the major causes.</td>
</tr>
<tr>
<td>NWR</td>
<td>High income is one major cause many people engage in hourly workers outside.</td>
</tr>
<tr>
<td>PRR</td>
<td>High wages are one important reason many people engaged in hours for part-time workers.</td>
</tr>
</tbody>
</table>

### Example 3

<table>
<thead>
<tr>
<th>Source</th>
<th>禽流感病毒踪迹不断在欧洲出现，造成人心惶惶。</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>Panic strikes as signs of bird flu virus continue to emerge in Europe.</td>
</tr>
<tr>
<td>Baseline</td>
<td>Traces of avian flu virus keeps on Europe, causing panic.</td>
</tr>
<tr>
<td>NWR</td>
<td>Traces of avian flu virus continued there, causing panic in Europe.</td>
</tr>
<tr>
<td>PRR</td>
<td>Traces of avian flu continue to emerge, causing panic in Europe.</td>
</tr>
</tbody>
</table>

Table 4: Translation examples to illustrate the advantage of RAE composition based phrase representations over others. NWR = neural word representations. PPR = phrase representations with RAE composition.

### 5.3 Comparison of Different Composition Methods

The second series of experiments were aimed at investigating whether we should learn semantic representations for phrases and at examining which composition method, i.e., either weighted vector addition or recursive autoencoder, is a better approach to create phrase vector representations from word representations. Table 2 presents the experimental results, from which we can draw the following observations:

- Integrating bilingual distributed phrase representations leads to a substantial improvement up to 0.53 BLEU points over the baseline.
- Phrase representations gave a better performance than word representations by up to 0.3 BLEU points.
- The RAE composition outperforms the weighted vector addition by up to 0.17 BLEU points.
5.4 Nonlinear vs. Linear Projection

As mentioned in Section 3, a linear variation of the bilingual projection can be derived by removing the hidden layer. In the third series of experiments, we investigated whether linear transformation is sufficient to project source language representations onto the target language semantic space. Our experimental results presented in Table 3 show that the nonlinear projection outperforms the linear one by 0.28 BLEU points. This suggests that the former is effective than the latter for transformation between the semantic spaces of the source and the target language even though they are learned by the same method from the same sets of data.

5.5 Translation Examples

Experimental results presented in the last four subsections show that the nonlinearly projected compositional phrase representations based on RAE give a better performance than the others. In this section, we examine a number of translation examples extracted from the test set, as presented in Table 4, to see the difference that the proposed method makes. These examples illustrate that the decoder equipped with the proposed semantic composition and bilingual nonlinear projection is able to select better translations for both continuous (Example 1 & 2) and non-continuous phrases (Example 3).

6 Related Work

Previous studies related to our research can be categorized into three strands as follows:

Distributed representations in monolingual settings Various methods are explored to learn distributed vector representations for words and phrases. Among them, vector space models are widely used, creating a vector to represent the co-occurrence relations between a target word and its contextual words (Bullinaria and Levy, 2007; Pado and Lapata, 2007). Topic models can be also used to construct distributed representations for words over topics that are learned from data Xiao et al. (2012). Recently, a variety of deep neural networks are applied to learn neural representations for both words and phrases in a continuous semantic space (Bengio et al. 2003; Collobert and Weston, 2008; Turian et al. 2010; Socher et al. 2012). All these methods can be used to create monolingual word representations for use in our framework to underlay the composition and projection operations.

Distributed representations for SMT In addition to the already mentioned three methods (Mikolov et al. 2013b; Zou et al. 2013a; Gao et al. 2013) in the Introduction section, very recently Zhang et al. (2014) have proposed a bilingually-constrained recursive autoencoder in this strand, which extends the traditional semi-supervised recursive autoencoders Socher et al. (2011) to learn semantic phrase representations. They learn representations of one language with constraints from the counterpart language and share learned representations for phrases in the other language while we learn representations for the source and target language separately.

Semantic similarity models Our work is also related to semantic similarity models used in various NLP tasks. Bullinaria et al. (2007) carried out a word-based semantic similarity task to exam the degree of correlation between human judgments for two individual words and vector based similarity values. Xiao et al. (2012) introduced a topic similarity model to measure the similarity of translation rules to a document in terms of topics. We differ from them in that we calculate semantic similarity scores based on bilingual phrase representations learned via semantic composition and bilingual projection.

7 Conclusion and Future Work

We have presented a flexible framework above, which learns bilingual distributed phrase representations for machine translation. In this framework, vector representations of phrases are
obtained by weighted vector addition or recursive autoencoder composition over words, which are represented as PMI-based vectors or continuous-valued vectors. We adopt a bilingual projection neural network to build nonlinear transformations between the source and the target language semantic space that are separately learned.

We integrate learned bilingual phrase representations into a hierarchical phrase-based SMT system. Our experimental results suggest the following:

- A semantic similarity model built on phrase representations is better than one built on word representations.
- Recursive autoencoder is superior to simple weighted vector addition in creating phrase vector representations from word vectors via composition.
- Nonlinear transformation is effective than linear transformation between the source and target language semantic space.

Our future work along this direction is to build stronger RAEs to construct vector representations for sentences.

Acknowledgment

The work was sponsored by the National Natural Science Foundation of China (Grants No. 61403269, 61432013 and 61333018) and Natural Science Foundation of Jiangsu Province (Grant No. BK20140355). We would like to thank three anonymous reviewers for their insightful comments.

References


Learning Bilingual Phrase Representations with Recurrent Neural Networks

Hideya Mino hideya.mino@nict.go.jp
Andrew Finch andrew.finch@nict.go.jp
Eiichiro Sumita eiichiro.sumita@nict.go.jp
National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, JAPAN

Abstract

We introduce a novel method for bilingual phrase representation with Recurrent Neural Networks (RNNs), which transforms a sequence of word feature vectors into a fixed-length phrase vector across two languages. Our method measures the difference between the vectors of source- and target-side phrases, and can be used to predict the semantic equivalence of source and target word sequences in the phrasal translation units used in phrase-based statistical machine translation. Our experiments show that the proposed method is effective in a bilingual phrasal semantic equivalence determination task and a machine translation task.

1 Introduction

In recent years, continuous vector representations of word, phrase, and sentence which alleviate issues of sparsity have successfully been used in a number of natural language processing tasks. Language models with continuous word representations (Bengio et al., 2003; Mikolov et al., 2010; Mikolov, 2012) based on neural networks have outperformed the previous state-of-the-art approaches. These language models map each word to a dense, low-dimensional, real-valued vector, and estimate the probability of words in a continuous space. Representations for phrases have been used in the context of Statistical Machine Translation (SMT). Zou et al. (2013) used phrasal representations for computing the distance between phrase pairs and added a feature based on this distance into the log-linear model of a phrase-based SMT system (Koehn et al., 2003). Their method learned bilingual word representations, and subsequently obtained the phrase-level representations by simply averaging word vectors. Continuous representations for phrases or sentences with neural networks – such as a feed-forward, recursive, or recurrent neural networks – have also been used in SMT. A phrase representation model using a feed-forward neural network for phrase-based SMT was proposed by Schwenk (2007, 2012), and achieved significant BLEU score improvements. Since the model directly projects feature vectors not from words but from phrases or sentences onto a continuous vector space, the representations can contain more global semantic information.

In this paper, we propose a new method to learn bilingual phrase representations for phrase-based SMT using two Recurrent Neural Networks (RNNs) (source- and target-side) combined in a simple linear architecture. We follow the idea of Cho et al. (2014) that the last hidden state of the RNN is a summary representation of the whole input phrase, and the summary representations with the same meaning are trained to be the same vector representation. The procedure is similar to that used in the RNN of Cho et al. (2014), which predicts the next word...
in the sequence with a conditional probability. In contrast to this, our model uses an objective with a similarity distance instead of a conditional probability, and learns to minimize the error distance. Furthermore, we developed a novel extension of the model that uses an autoencoder, which is an architecture trained to provide a latent representation of its input by means of a nonlinear encoder and an associated decoder. The objective involves three kinds of errors: a next symbol error for predicting the next word in a phrase, a semantic error for the comparison of the summary phrase representations, and a reconstruction error for the autoencoder. The prediction error represents how well the intermediate hidden states can predict the next word in a sequence. The semantic error represents the dissimilarity between the final hidden states on the source- and target-side. The reconstruction error represents how well the hidden states represent the words in a phrase.

We introduce a bilingual phrase similarity feature derived from our proposed method as a new feature into the log-linear model of a phrase-based SMT system applied to an English-Japanese translation task, and confirm the effectiveness of our method on this task. The results of the experiments show that our model is able to indicate the effective phrase pairs for machine translation. In this paper, though our model is symmetric and does not differentiate between source- and target-side, we use the following notation: the left-side RNN is referred to as the source-side and the right-side RNN (we use overbars on the symbols to differentiate it) is referred to as the target-side.

2 Related work

In this section, we review recent work on neural network phrase representation models.

Continuous phrase representation models with a feed-forward neural network were studied in Schwenk (2007, 2012). The models estimated translation probabilities for unseen phrases with a continuous vector space of phrases. Le et al. (2012) proposed a similar approach to score phrase pairs using fixed-size inputs and outputs. Devlin et al. (2014) proposed a neural network joint model (NNJM) as an extension of the NNLM (Bengio et al., 2003). The NNJM calculates the target-side word probability by using a target-side language model in combination with a context from the source-side. The NNJM requires a maximum length for the source-side phrases. These approaches employ feed-forward neural networks and are constrained to operate on phrases of limited length.

The use of recursive neural networks addresses the fixed-size issue by using a tree structure of phrases and sentences. The recursive neural network maps features from subsequences of a phrase to a continuous vector on each node of the tree recursively. Li et al. (2013) described an ITG reordering classifier which predicted phrase reorderings in SMT that was able to exploit syntactic and semantic information. Zhang et al. (2014, 2015) proposed bilingually-constrained recursive autoencoders, which generated phrasal embeddings for machine translation by learning to minimize the semantic distance between translation equivalents, and maximizing the distance between non-translation pairs.

In contrast to the work on recursive networks, it is also possible to create continuous phrase representations with RNNs. Here, simpler models are possible that do not need take the tree structure of their input into account. Kalchbrenner and Blunsom (2013) proposed recurrent continuous translation models based on recurrent language models (RLMs), which predict target words from an unbounded history of both source and target words with a conditional probability. In their implementation convolutional neural networks were used to model the source-side. Cho et al. (2014) proposed a gated recurrent unit which adaptively remembers and forgets its state based on the input signal to the unit. This model was used to score each phrase pair in the phrase table for SMT.
3 Bilingual Phrase Representation Model

3.1 Phrase Representations

Our bilingual phrase representation model comprises two RNNs: one for source phrases, and the other for target phrases. Each RNN reads a sequence of word representations, and transforms it into a fixed-length vector that holds the semantic content of the whole input sequence. We call this vector the phrasal representation. Then, the model identifies phrase pairs with the same meaning on both source- and target-side, by computing the similarity distance between the respective source and target side phrase representations.

Figure 1 shows the framework of the bilingual phrase representation model, where \( r_k (0 \leq k \leq l) \) and \( r_k (0 \leq k \leq m) \in \mathbb{R}^{n \times 1} \) are word representations of the phrases \( r \) and \( \bar{r} \) in a phrase pair \((r, \bar{r})\), \( h_k (0 \leq k \leq l) \in \mathbb{R}^{q \times 1} \) are hidden layers, \( a_k (1 \leq k \leq l) \) and \( a_k (1 \leq k \leq m) \in \mathbb{R}^{q \times 1} \) are output layers, \( c \) and \( \bar{c} \in \mathbb{R}^{q \times 1} \), which are also the last hidden layers, are the summary layers which contain the summary representations of the phrases \( r \) and \( \bar{r} \), and four types of transformation matrices: \( I \) and \( \bar{I} \in \mathbb{R}^{n \times n} \), the input vocabulary transformation matrices, \( F \) and \( \bar{F} \in \mathbb{R}^{n \times q} \), the autoencoder transformation matrices, \( R \) and \( \bar{R} \in \mathbb{R}^{q \times q} \) the recurrent transformation matrices, and \( O \) and \( \bar{O} \in \mathbb{R}^{n \times q} \), the output transformation matrices. The parameter \( q \) indicates the size of the summary representations. In Figure 1, \( n := 2 \) and \( q := 3 \) for the purposes of illustration.

Each RNN minimizes the error distance over continuous word representations of each phrase by being trained to predict the inputs, the next inputs, and the summary representation of the whole input sequence which is shared between both source- and target-sides. Hence, the hidden layer activation vectors \( H = \{h_0, h_1, \ldots, h_l\} \) and \( \bar{H} = \{\bar{h}_0, \bar{h}_1, \ldots, \bar{h}_m\} \) contain information of the previous input words and the next word in the input sequence. The summary layers \( c \) and \( \bar{c} \), which are the last hidden layer of each RNN, contain semantic information in common with each phrase. Any language-specific information is weakened by optimizing to jointly predict source and target. The source-side RNN learns according to following steps: first, the activations in the hidden layers \( H \) are calculated from the word representations recursively.
as follows:

\[ h_k = \begin{cases} \sigma(I \cdot r_0) & k = 0 \\ \sigma(R \cdot h_{k-1} + I \cdot r_k) & 1 \leq k < l \end{cases} \]  

(1)

where \( \sigma \) is a nonlinear function such as \( \tanh \). \( r_0 \) and \( R_0 \) are the representations of the source and target start symbols. When there are no constraints on the hidden layers \( H \), the RNNs are able to minimize the error distance by making \( H \rightarrow [0] \), which is undesirable. To prevent such behavior, the hidden layers \( H \) are normalized to have unit length. Then, the output layers \( o \) that predict the next word of the input, the summary layer \( c \) that predicts the target-side summary \( e \), and the autoencoder layers \( a \) that predict the vector \( r_k \) (representing the source word at position \( k \)) are calculated from the hidden layer activations \( H \) in Equation (1) as follows:

\[ o_k = \sigma(O \cdot h_{k-1}) \quad (1 \leq k \leq l) \]  

(2)

\[ c = h_l \]  

(3)

\[ a_k = \sigma(F \cdot h_k) \quad (1 \leq k \leq l) \]  

(4)

Bias values for Equations (1), (2), (3), and (4) are included in the computation. To avoid overfitting, we trained each layer using the dropout method (Srivastava et al., 2014). There are three kinds of prediction error, which we denote: the next symbol error \( E_o \), error, the semantic error \( E_c \), and the reconstruction error \( E_a \). These were calculated by using Euclidean distance:

\[ E_o(r|o; \theta) = \frac{1}{2l} \sum_{k=1}^{l} ||r_k - o_k||^2 \]  

(5)

\[ E_c(e|c; \theta) = \frac{1}{2} ||e - c||^2 \]  

(6)

\[ E_a(r|a; \theta) = \frac{1}{2l} \sum_{k=1}^{l} ||r_k - a_k||^2 \]  

(7)

where \( \theta = \{ I, R, F, O \} \) is the set of source-side parameters to be learned, together with the bias parameters. Equation (5) represents the sum of the next symbol error distance between each input \( r_k \) in the source-side phrase and the output prediction \( o_k \). The output from the last hidden layer \( h_l \) is the summary representation \( c \) (\( c \equiv h_l \)). A shared semantic representation of both source and target is required, and therefore \( c \) and \( e \) are trained jointly using error signals based on the distance between them. The error is calculated using the semantic error defined in Equation (6). Equation (7) is the sum of the reconstruction error distance between each input \( r_k \) in the source-side phrase and the autoencoder’s reconstruction \( a_k \). The autoencoder is used for learning representations of words (Chandar A P et al., 2014), phrases (Zhang et al., 2015), and sentences (Socher et al., 2011; Li et al., 2013). The target-side errors were calculated in the same manner. The objective function \( J \) is the sum of the total error distance from the source and target RNNs, and is represented by using Equations (5), (6), and (7) as:

\[ J = \alpha E_o(r|o; \theta) + E_c(e|c; \theta) + \beta E_a(r|a; \theta) + \gamma ||\theta|| \]  

+ \( \alpha E_o(\bar{r}|\bar{o}; \tilde{\theta}) + E_c(\bar{c}|\bar{c}; \tilde{\theta}) + \beta E_a(\bar{r}|\bar{a}; \tilde{\theta}) + \gamma ||\tilde{\theta}|| \]  

+ \( 2 \cdot E_a(\bar{c}|c; \theta, \tilde{\theta}) \)  

+ \( \alpha(\bar{E}_o(r|o; \theta) + \tilde{E}_o(\bar{r}|\bar{o}; \tilde{\theta})) \)  

+ \( \beta(\bar{E}_a(r|a; \theta) + \tilde{E}_a(\bar{r}|\bar{a}; \tilde{\theta})) \)  

+ \( \gamma(\||\theta|| + ||\tilde{\theta}||) \)  

(8)
where $\tilde{\theta}$ is the set of target-side parameters, and $\alpha$ and $\beta$ are the hyper-parameters for the balance of each error. We also use an $L_1$ regularization term in the objective function. In Equation (8), we group the semantic error $E_c$ terms of source- and target-side (which use the summary vectors $c$ and $\bar{c}$) into one term, and arrange the terms according to error type.

The parameters $\theta$ and $\tilde{\theta}$ are optimized to minimize Equation (8) using the AdaGrad stochastic adaptive subgradient algorithm (Duchi et al., 2011; Green et al., 2013):

$$
\theta_i = \theta_{i-1} - \eta \frac{\partial J}{\partial \theta_{i-1}} G_i^{-1/2} \\
G_i = G_{i-1} + \left( \frac{\partial J}{\partial \theta_{i-1}} \right)^2
$$

where $\eta$ is the learning rate, $i$ is the number of the training iterations, and $G$ is the sum of the squares of the past gradients. $\theta$ and $\tilde{\theta}$ are learned and updated in every iteration through the training data of phrase-pairs. The number of training iterations was determined using development data.

### 3.2 Word representations

Word representations, in which words are represented as real-valued vectors (Bengio et al., 2003; Mikolov et al., 2013), serve as the inputs to our model. The word representations $r$ are calculated as:

$$
r_i = L u_i \in \mathbb{R}^{n \times 1}
$$

where $n$ is the number of dimensions of the vector, $L \in \mathbb{R}^{n \times |V|}$ is a word embedding matrix, $|V|$ is the vocabulary size, and $u_i$ is a binary vector which is zero in all positions except for the $i^{th}$ index. Given a phrase which is a sequence of $l$ words, each word has a vocabulary index $i$ into the columns of the word embedding matrix $L$. The $i^{th}$ column of the embedding matrix is the word’s representation vector. The matrix $L$ is pre-trained by training a neural network on unlabeled monolingual data. In our experiments, we trained the matrices $L$ and $\tilde{L}$ for source and target word representations using the Word2Vec toolkit (Mikolov et al., 2013). The size of the word representation vector $n$ is usually determined empirically.

### 4 Experiments

We conducted two experiments with the Bilingual Phrase Representation Model: a phrase-pair extraction task and a phrase-based SMT task.

### 4.1 Data and model parameters

Both experiments were conducted on two English-Japanese (en-ja) corpora. One was from IWSLT 2007 (Fordyce, 2007) which is in the domain of spoken travel conversation and the other was a patent translation corpus from NTCIR-10 (Goto et al., 2013). The Japanese sentences were tokenized using KyTea (Neubig et al., 2011).

<table>
<thead>
<tr>
<th>Data</th>
<th>Training</th>
<th>Development</th>
<th>Test</th>
<th>Monolingual</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>sent word types</td>
<td>sent word types</td>
<td>sent word types</td>
<td>sent word types</td>
</tr>
<tr>
<td>IWSLT 2007</td>
<td>40K en 9.5K ja</td>
<td>0.5K en 1.2K ja</td>
<td>0.5K en 0.8K ja</td>
<td>0.5K en 0.9K ja</td>
</tr>
<tr>
<td>NTCIR-10</td>
<td>720K en 119K ja</td>
<td>2.0K en 5.0K ja</td>
<td>0.5K en 2.4K ja</td>
<td>41M en</td>
</tr>
</tbody>
</table>

Table 1: Data sets
Table 1 provides statistics on each corpus. The “sent” column indicates the number of sentence pairs, and the “word types” columns of “en” and “ja” indicate the number of English and Japanese unique words. The “Monolingual” column indicates the size of the monolingual data for the training of the word representations described in Section 3.2. For IWSLT 2007, we used the training data for the training of the word representations. For NTCIR-10, we used about 723K sentence pairs belonging to the physics domain, which contains the most documents among the domains, according to International Patent Classification (IPC) code 1. 720K sentence pairs from the documents published between 1993 to 2005 were used as the training data, and 2.0K and 0.5K sentence pairs randomly sampled from the 2006 and 2007 documents were used as the development and test data respectively. We also used the 2006 and 2007 documents to extract the similar phrases. Furthermore, we used the English and Japanese monolingual corpus in NTCIR-10 for the training of word representations.

For the extraction of phrase pairs, we used MGIZA++ (Gao and Vogel, 2008) and grow-diag-final-and heuristics of the Moses toolkit (Koehn et al., 2007). To facilitate effective learning, we used only phrase pairs that contained content words (i.e. had at least one noun or verb word in the phrase) and had a high translation probability (a threshold on the source-given-target conditional probability was used). We extracted phrase pairs from the training, development and test data. The training phrase-pairs were used for training the neural network models. The development phrase-pairs were used to control the training of the models. The model was trained for 4,000 iterations, and estimated parameters \( \theta \) and \( \tilde{\theta} \) by evaluating the highest accuracy of the top-1 phrase pair extracted with the development data. The evaluation of the accuracy was performed as follows: for each source-side phrase in a 100-phrase pair development set, the system was requested to choose the top-\( n \) candidate target word sequences from the target-side 100 phrases. The minimum error distance defined in Eq. (8) was used to produce the top-\( n \) list. The test phrase-pairs were used for the experiment on the phrase-pair extraction. Consequently, we extracted about 316K phrase pairs from IWSLT 2007 and 23M phrase pairs from the NTCIR-10 training set. Due to the computation time, we randomly selected 10K phrase pairs for IWSLT 2007 and 100K phrase pairs for NTCIR-10 as the training set from the full set of phrase pairs. For the development set, we randomly selected another 300 phrase pairs. We also extracted 100 phrase pairs from the test data for the experiment on phrase-pair extraction.

For the parameters of the model, the input and output vector-size \( n \) was set to 200. The summary vector-size \( q \) was also set to 200. The activation function \( \sigma \) was tanh. The dropout rate was 0.9 for the hidden layers and 0.5 for the other layers. The learning rate \( \eta \) was set to 0.01 for the experiments with IWSLT 2007 and 0.02 for the experiments with NTCIR-10. The regularization rate \( \lambda \) was set to 0.01. The hyperparameters \( \alpha \) and \( \beta \) in Equation (8) were set to 0.01. All weight parameters \( \theta \) and \( \tilde{\theta} \) were randomly initialized, and all bias parameters were initialized to zero.

4.2 Experimental design

4.2.1 Phrase-pair Extraction

We did two sub-experiments for the phrase-pair extraction: the evaluation of the accuracy and the extraction of phrases with similar meaning. The accuracy was calculated with the test phrase-pairs. In order to mitigate the issue of the training process terminating in a local minimum, we evaluated the accuracy at each iteration on four data sets: three different development data sets (DEV.1, DEV.2, and DEV.3) and a fourth (closed) set which was the test data itself. This resulted in four different models, each defined by the estimated parameters \( \theta \) and \( \tilde{\theta} \) at the iteration that gave rise to the highest accuracy on the respective data set. Each of the development data sets contained 100 phrase pairs sampled randomly without replacement from the full

1SECTION G of IPC code indicates the physics domain
The data for extracting the similar phrases was obtained by searching for English phrases that were close to their Japanese counterpart phrases in NTCIR-10. We used the English phrases in unseen sentences published in 2006 and 2007 and the Japanese phrases from sentences randomly selected from the training sets. We calculated the error distance between the Japanese and the English phrases with the model terminated using the accuracy on DEV.1. To limit the number of the English phrase candidates, we only used the English sentences that were similar to the Japanese sentences. The similarity was assessed using the number of lemmatized words in the Japanese sentences, that could be translated to lemmatized words in the English sentences by using a Japanese-English dictionary.

### 4.2.2 Phrase-based SMT

The phrase-based SMT experiments were performed with the two models, in which the parameters $\theta$ and $\hat{\theta}$ were estimated on DEV.1 and DEV.2 of the phrase-pair extraction experiment, using the phrase-pairs extracted using the Moses toolkit. We added the inverse of the error distance used in the ranking experiments, as a feature into the log-linear model of the Moses decoder. The 5-gram language models were built using the SRILM toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1996). For word and phrase alignments, we used MGIZA++ and grow-diag-final-and heuristics. To tune the weights with respect to the BLEU score (Papineni et al., 2002), we used $n$-best batch MIRA (Cherry and Foster, 2012). The distortion limit parameter was set to 10. We evaluated each model on BLEU using the NIST’s mteval-v13a.pl script. Statistical significance testing of the BLEU differences was performed using paired bootstrap resampling (Koehn, 2004).

For both experiments, we tested with two models. The first was the proposed Bilingual Phrase Representation Model (BPRM). The second was the same BPRM model with the autoencoder layer removed.

### 5 Results and Analysis

Tables 2 and 3 present the results of the phrase pair extraction task, and Table 4 presents the results of the phrase-based SMT task.

---

Table 2: Accuracy of the phrase-pair extraction: 1-best and 10-best on three development sets

<table>
<thead>
<tr>
<th>Development Set</th>
<th>BPRM (without autoencoder)</th>
<th>DEV.1</th>
<th>DEV.2</th>
<th>DEV.3</th>
<th>closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT 2007</td>
<td>1-best</td>
<td>0.03</td>
<td>0.06</td>
<td>0.04</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>10-best</td>
<td>0.23</td>
<td>0.25</td>
<td>0.24</td>
<td>0.32</td>
</tr>
<tr>
<td>BPRM</td>
<td>1-best</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>10-best</td>
<td>0.27</td>
<td>0.28</td>
<td>0.28</td>
<td>0.28</td>
</tr>
<tr>
<td>NTCIR-10</td>
<td>BPRM (without autoencoder)</td>
<td>1-best</td>
<td>0.20</td>
<td>0.17</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>10-best</td>
<td>0.45</td>
<td>0.43</td>
<td>0.44</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>1-best</td>
<td>0.20</td>
<td>0.24</td>
<td>0.20</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>10-best</td>
<td>0.43</td>
<td>0.41</td>
<td>0.45</td>
<td>0.47</td>
</tr>
</tbody>
</table>

---

2http://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

---

Miami, Oct 30 - Nov 3, 2015   | p. 50
5.1 Phrase-pair Extraction

In Table 2, the results of the experiments on NTCIR-10 data show higher levels of accuracy than the experiments on IWSLT 2007 data. The reason for this may be indicated in Table 1 which shows that the number of the word types on IWSLT 2007 is smaller than on NTCIR-10, and the proportion of words that appear multiple times in the corpus was 70% for IWSLT 2007 and 30% for NTCIR-10. The set of phrases from the IWSLT 2007 data is likely to contain many similar phrases, thereby making the discrimination more difficult. The differences between DEV.1, DEV.2 and DEV.3 were small. It is likely that there is no local minimum problem in these three models. Table 3 shows examples of phrase pairs in the training data and the English phrases which were extracted from the English monolingual documents with BPRM, illustrating the kinds of semantically similar phrases our model is capable of identifying.

5.2 Phrase-based SMT

The results of the phrase-based SMT experiments on IWSLT 2007 data show that the proposed method was able to improve machine translation quality. The statistical significance tests between PBMT and the other models shows a significant improvement on both DEV.1 and DEV.2 at $p < 0.05$. Although the results were not statistically significant, the full BPRM approach achieved higher BLEU scores than the BPRM without the autoencoder. Therefore we believe it is likely that the autoencoder is effective for the improvement of translation quality. For NTCIR-10, the improvements in performance were smaller than on the IWSLT 2007 data set.

In terms of computational time for training the model, training with 10K phrase pairs on IWSLT 2007 took about 40 seconds for one iteration, and training with 100K phrase pairs on NTCIR-10 took about 3 minutes for one iteration. Training was performed on an 8-core 2.00GHz Intel Xeon CPU.

In summary, our model was capable of identifying phrase-pairs with semantically source and target word sequences, and this knowledge could be exploited to yield an respectable improvement in machine translation quality.
6 Conclusion

In this paper, we proposed a Bilingual Phrase Representation Model which learns phrase representations by using source- and target-side Recurrent Neural Networks. We demonstrated the effectiveness of the proposed model on an English-Japanese corpus on two tasks: phrase-pair extraction and statistical machine translation. Future avenues of research include investigating hyper-parameter tuning for the objective function, and discovering a method to select appropriate initial values of the weights which were set randomly in this work.

Acknowledgements

We are deeply grateful to Taro Watanabe, Atsushi Fujita and anonymous reviewers for their suggestions and insightful comments on the early version of this paper.
References


A Pilot Study Towards End-to-End MT Training

Feifei Zhai  ffzhai2012@gmail.com
Queens College, City University of New York, Queens, NY 11367
Liang Huang†  lianghuang.sh@gmail.com
School of EECS, Oregon State University, Corvallis, OR 97330

Abstract

Typical MT training involves several stages, including word alignment, rule extraction, translation model estimation, and parameter tuning. In this paper, different from the traditional pipeline, we investigate the possibility of end-to-end MT training, and propose a framework which combines rule induction and parameter tuning in one single module. Preliminary experiments show that our learned model achieves comparable translation quality to the traditional MT training pipeline.

1 Introduction

Typically, as shown in Figure 1(a), traditional machine translation (MT) training involves a long pipeline of several stages, including word alignment (by GIZA++), rule extraction, translation model (TM) estimation (by max-likelihood), and parameter tuning by MERT (Och, 2003), PRO (Hopkins and May, 2011), or MIRA (Watanabe et al., 2007; Chiang et al., 2008). This cascaded procedure inevitably propagates errors downstream, while at the same time making MT training overly complicated.

In this paper, instead of following the traditional training pipeline, we explore the possibility of end-to-end MT training, which would ideally induce a full model with translation

---

†Work done while Prof. Liang Huang was in City University of New York.
rules at the same time (as shown in Figure 1(b)). Practically, it is a big challenge, because the search space would be too prohibitive if we consider all possible rules from scratch without any constraints from word alignment.

To make this idea computationally affordable, we take a step back and propose a joint learning framework to do rule induction and parameter tuning together, shown in Figure 1(c). Instead of knowing nothing at the beginning, we use word translation probabilities from GIZA++ to compute the lexical translation probabilities of phrase pairs, which is an effective feature guiding us to select good translation rules. Note that even with these probabilities, this is still a non-trivial problem, since the joint learner still needs to deal with the entire space of all possible translation rules and all possible decoding derivations.

For simplicity reasons we use phrase-based translation. Final experiments show that the joint learned model achieves comparable performance to the conventional training pipeline in a small scale. To our best knowledge, this is the first time, although on small data, verifying that it is possible to do effective end-to-end MT training. The most significant contribution of this paper lies in this point. We believe this will be a promising direction to MT research.

2 Joint Framework for Rule Induction and Parameter Tuning

Algorithm 1 gives our joint framework in detail. We start with an empty rule set \( R = \emptyset \), and for each sentence pair \((x, y)\) where \(x\) is the source input and \(y\) the target translation, we try forced decoding finding all derivations that can map \(x\) to \(y\) (line 5), using all possible phrase pairs from \((x, y)\). This is because without a rule set to start with, theoretically any portion of \(x\) could map to any portion of \(y\) and the learner needs to figure out which phrase pairs make more sense. Here \(f(\cdot)\) and \(e(\cdot)\) return the source and target projection of a derivation respectively. This forced decoding, although much more constrained than real decoding, is still intractable, and we have to resort to beam search similar to those employed in real decoding, i.e., the set \(D\) in line 5 is the “best-scoring” subset of all possible forced derivations.

Algorithm 1 Joint rule induction and param. tuning.

1: \( R \leftarrow \emptyset \) \(\triangleright\) initial rule set
2: \( w \leftarrow 0 \) \(\triangleright\) initial parameters
3: repeat
4: for each sentence pair \((x, y)\) in bitext do
5: \( D(x, y) \leftarrow \{d \mid f(d) = x, e(d) = y\} \) \(\triangleright\) forced decoding
6: \( R \leftarrow R \cup \{r \mid r \in d, d \in D(x, y)\} \) \(\triangleright\) add new rules
7: recalculate conditional probs for each \( r \in R \)
8: for each sentence pair \((x, y)\) in bitext do
9: \( d' \leftarrow \arg\max_{d \mid f(d) = x} w \cdot \Phi(x, d) \) \(\triangleright\) real decoding
10: if \( e(d') \neq y \) then \(\triangleright\) wrong translation?
11: \( d^* \leftarrow \arg\max_{d \in D(x, y)} w \cdot \Phi(x, d) \) \(\triangleright\) best gold deriv.
12: \( w \leftarrow w + \Phi(x, d^*) - \Phi(x, d') \) \(\triangleright\) update model
13: until converged

After forced decoding we collect translation rules in the forced derivations and add them to standing rule set \( R \) (line 6). Then we recalculate the rule conditional probabilities (line 7, TM estimation). After that we perform real decoding, trying to find the best translation (line 9).

1At first, since all the weights are set to 0, all derivations have the same score 0. Both the best-scoring forced derivation and real decoding derivation are selected randomly.
If this translation $e(d')$ is different from the reference translation $y$, then an update is needed to reward the highest-scoring forced (or “gold”) derivation $d^*$ and to penalize the highest-scoring non-gold (Viterbi) derivation $d'$ (line 12). We apply a latent-variable max-violation perceptron (Huang et al., 2012), which has been successfully used in MT training (Yu et al., 2013; Zhao et al., 2014), to do update.

Technically, this framework is similar to (Xiao and Xiong, 2013). The main difference is that we try to do end-to-end MT training, and combine rule induction, TM estimation, and parameter tuning together, while they only focus on rule induction. Moreover, to accommodate the vast amount of search errors in decoding, we update weights by max-violation perceptron, performing prefix instead of full-sequence updates, whereas the updates in (Xiao and Xiong, 2013) are still full-sequence updates which are insensitive to search errors. For simplicity reasons we do not make this difference explicit in line 12.

3 Phrase-based Forced Decoding

Forced decoding generates those derivations that can produce the exact reference translation. For example, given the following sentence pair,

Source: Bush yù Shålóng jüxing le huıtán
Target: Bush held a talk with Sharon

One possible derivation created by forced decoding is as follows:

\[
\begin{align*}
(0, \ldots) & : (0, "") \\
(\bullet_1, \ldots) & : (s_1, "Bush") \\
(\bullet_\ldots s_6) & : (s_2, "Bush held a talk") \\
(\bullet_\ldots s_3) & : (s_3, "Bush held a talk with Sharon") \\
\end{align*}
\]

where each hypothesis is in form $(v) : (s, p)$, in which $v$ is the coverage vector (a $\bullet$ indicates the source word at this position is already “covered (or translated)”), and $(s, p)$ is the score and partial translation of each state.

Generally, we can employ a traditional phrase-based decoder to do forced decoding. However, the distortion limit in the decoder will prohibit long-distance reorderings, and exclude many sentence pairs from getting forced derivations, especially for language pairs with very different word orders, such as Chinese and English.

Hence, in order to do better forced decoding, we use a more flexible limit to constrain the number of gaps during decoding (also known as “IBM constraint” in (Zens et al., 2004)), rather than distortion limit. Here, a gap refers to a consecutive of positions that are not covered in the coverage vector. For example, consider the third hypothesis of the above derivation, $(\bullet_\ldots s_3) : (s_3, "Bush held a talk")$, its coverage vector has one gap, i.e., the two untranslated words. Also, we don’t want the decoding process to be too flexible on reordering, so we demand that there are at most two gaps in a specific coverage vector (See Figure 2).

Figure 2: Two possible scenarios in gap-based decoding. The gray boxes denote covered segments.
Obviously, compared to distortion limit, gap limit is more flexible for decoding, especially when there is only one gap in the coverage vector, we can jump to any uncovered position of the source sentence for translation. Hereafter, we use dl-based decoding to denote the decoding algorithm using distortion limit, while gap-based decoding denotes the algorithm using gap limit.

In addition, same as forced decoding, real decoding can also be dl-based or gap-based. We use the same choice for forced and real decoding.

4 Rule Generation and Collection

As described in §2, since there is no predefined rule set, we need to consider all possible phrase pairs from the training sentence pair. To reduce the search space of phrase pairs, we set the maximum phrase length to 3. As an example, considering the above sentence pair, we enumerate all possible source and target phrases from it for forced decoding:

<table>
<thead>
<tr>
<th>possible source phrases</th>
<th>possible target phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bǎshi</td>
<td>Bush</td>
</tr>
<tr>
<td>Bǎshi yǔ</td>
<td>Bush held</td>
</tr>
<tr>
<td>Bǎshi yǔ Shālóng</td>
<td>Bush held a</td>
</tr>
<tr>
<td>yǔ Shālóng</td>
<td>held</td>
</tr>
<tr>
<td>yǔ Shālóng jìxìng</td>
<td>held a</td>
</tr>
<tr>
<td></td>
<td>held a talk</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

From these phrases, we can pick a source phrase and a target phrase arbitrarily to form a phrase rule, and apply it to forced decoding. For example, as a start, if we select Bǎshi and Bush respectively, we can create hypothesis \( (1, \_\_\_\_\_\_) : (s_1, "Bush") \) in the above derivation. We consider all possible combinations of these phrases as rules for forced decoding.

In addition, since training sentence pairs do not always contain equal information on both sides, we introduce two null rules \( \langle f, \text{null} \rangle \) and \( \langle \text{null}, e \rangle \) to capture the redundant information for forced decoding. \( \langle f, \text{null} \rangle \) deletes a source word, and \( \langle \text{null}, e \rangle \) inserts a target word to the translation.

After forced decoding, we collect the rules used in forced derivations and add them to the standing rule set R (line 6 in Algorithm 1). Based on the rule counts in R, we recalculate the rule conditional probabilities by the formula from (DeNero et al., 2006):

\[
\phi(e|f) = \frac{c(f,e)}{c(f) + k^{l-1}}
\]

where \( f \) and \( e \) are the source and target phrase, \( c(\cdot) \) is the count of phrase or phrase pair, \( l \) is the length of phrase \( f \), and \( k \) is a tuning parameter. The formula boosts the probability of short phrases, and results in better translation quality. After some validation experiments, we set \( k = 4.0 \) finally.

Similar to our rule generation process, Wuebker and Ney (2013) has proposed a length-based training method to do rule induction by EM algorithm. The major difference between our framework and theirs is that their algorithm only relates to rule induction, and still need MERT to do parameter tuning, while we combine rule induction and parameter tuning together. In this way, the two step in our framework can help each other, but the two step in Wuebker and Ney.

\[^2\] We have also tried longer length limit, but the performance becomes worse, because with longer limit, the learner greatly prefers longer phrases, which are not good at generalization.

\[^3\] We count all rules in all unpruned forced derivations, which are stored in a lattice. The rules are counted based on its contribution to the derivation, i.e., the score it added to the derivation.
are isolated. We think it is possible to integrate their method into our framework. We leave it as our future work.

5 Parameter Tuning

We apply a max-violation perceptron, which has successfully scaled the MT tuning process from dev set to training data (Yu et al., 2013), to do parameter tuning. We skip the details here. The basic idea is to find the step where the difference (violation) between the best forced decoding derivation and the best real decoding derivation is maximal, and then update parameters at this step so that most information can be learned.

Theoretically, the max-violation perceptron allows arbitrary features. However, in our joint learning scenario, we find it is very easy to get overfitting with sparse features. We conjecture that this is because sparse features have a tight connection to rules. Once some bad rules are introduced, it is difficult for the learner to correct them. We will make more effort on this in future.

Here, we only use dense features, which are the same as the ones for phrase-based translation, including bidirectional translation probabilities (computed by Formula (1) in §4), bidirectional lexical translation probabilities (estimated based on (Koehn et al., 2003) by word translation probabilities from Moses (Koehn et al., 2007) based on GIZA++), language model, rule penalty, length penalty, and distortion cost. To simplify the system, we haven’t used the lexicalized reordering model here.

6 Related Work

On a high level, this work is a combination of two different research directions. One direction is to induce translation rules directly from bitext, rather than using word alignment (Marcu and Wong, 2002; Cherry and Lin, 2007; Zhang et al., 2008; DeNero et al., 2008; Blunsom et al., 2009; Neubig et al., 2011; Levenberg et al., 2012; Xiao and Xiong, 2013). They can learn better translation rules, but don’t care about parameter tuning of SMT. Another direction is discriminative training for MT parameter tuning (Liang et al., 2006; Arun and Koehn, 2007; Blunsom et al., 2008; Flanigan et al., 2013; Green et al., 2013; Yu et al., 2013; Zhao et al., 2014). Both the two directions have achieved promising results. We differ from these works in that we make efforts to combine their spirit together, and try to do rule induction and parameter tuning in one step.

7 Experiments

To evaluate our method, we conduct experiments on Chinese-to-English translation. Since we need to do forced decoding and real decoding on training corpus each iteration, to guarantee the training efficiency, here we use a small scale data. It includes about 100K sentence pairs from FBIS, where the length of each sentence is less than 30 words. We use GIZA++ and grow-diagonal-final-and strategy to create symmetric word alignment. We train a trigram language model on 1.5M English sentences.

We base our experiments on Cubit, a state-of-the-art phrase-based system in Python (Huang and Chiang, 2007). For the joint learning method, we set the beam size for forced decoding as 10, real decoding as 30. A maximum phrase length of three was used for both baseline and our joint system. The beam size for final test decoding is set to 50.

We take the newswire portion of NIST MT 2006 data as our dev set, and the NIST MT 03-05 data and the newswire portion of 2008 data as the test set. For baseline, we use MERT (Och, 2003) to tune weights.

For the newly generated rules which do not have any counts, we assign a very small probability for them.
Figure 3: The percentage of reachable sentences in forced decoding at each iteration, with various decoding options.

7.1 DL-based Decoding vs. Gap-based Decoding

We use dl-based decoding and gap-based decoding respectively to promote our joint learning framework. Since we only use dense feature for training, the learning process peak very fast, and we can reach the best BLEU score on dev set in 4 - 5 iterations for both dl-based decoding or gap-based decoding.

Figure 3 compares the forced decoding reachability, and Table 1 shows the corresponding translation result of different decoding settings. From the Figure and Table, we can first certify the necessity of Null Rule, which improves translation quality significantly. Also, distortion limit has a big influence. Big limit (dl=10) allows more flexibility on forced decoding, and thus extract more effective rules, and get better translation quality than smaller limit (dl=6). Finally, gap limit gets the best translation quality among all the three, verifying the effectiveness of gap-based decoding.

<table>
<thead>
<tr>
<th>decoding</th>
<th># rule</th>
<th>dev</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o Null Rule</td>
<td>dl=6</td>
<td>0.44M</td>
<td>22.04</td>
</tr>
<tr>
<td></td>
<td>dl=10</td>
<td>0.55M</td>
<td>23.86</td>
</tr>
<tr>
<td></td>
<td>gap ≤ 2</td>
<td>0.59M</td>
<td>23.85</td>
</tr>
<tr>
<td>w/ Null Rule</td>
<td>dl=6</td>
<td>0.93M</td>
<td>23.05</td>
</tr>
<tr>
<td></td>
<td>dl=10</td>
<td>0.94M</td>
<td>24.19</td>
</tr>
<tr>
<td></td>
<td>gap ≤ 2</td>
<td>1.02M</td>
<td>24.42</td>
</tr>
</tbody>
</table>

Table 1: Rule set size and BLEU results of our joint learning method with different decoding algorithms.

An interesting phenomenon in Figure 3 is that the gap-based decoding algorithm gets more forced decodable sentences than dl-based decoding (dl=10) without Null Rule, but gets fewer ones after adding Null Rule. This is because when we use Null Rule, once the jump exceeds
distortion limit (bigger than 10), the dl-based forced decoder will not jump, but use Null Rule to generate the corresponding target word, and adopt another Null Rule to delete the source word later. In this way, the dl-based decoder could handle any long-distance jump. Since dl-based decoding is also more flexible than the gap-based one in short-distance jump, it is reasonable that dl-based decoding gets more forced decodable sentences after adding Null Rule. This is also the reason why dl-based decoding collects fewer translation rules (Table 1) with more forced decodable sentences. Since it utilizes many Null Rules on building forced derivations, the number of useful rules decreases in these derivations, resulting in fewer useful translation rules.

Table 2: BLEU scores of different translation systems. The baseline system uses conventional pipeline training. “All” is a combination of the 4 test sets.

<table>
<thead>
<tr>
<th>system</th>
<th># rule</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>08</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>dl=6</td>
<td>1.18M</td>
<td>21.72</td>
<td>25.47</td>
<td>22.01</td>
<td>23.37</td>
</tr>
<tr>
<td></td>
<td>gap</td>
<td>1.18M</td>
<td>21.19</td>
<td>25.18</td>
<td>21.60</td>
<td>21.57</td>
</tr>
<tr>
<td>joint</td>
<td>dl=6</td>
<td>0.93M</td>
<td>20.45</td>
<td>24.37</td>
<td>20.52</td>
<td>20.12</td>
</tr>
<tr>
<td></td>
<td>gap</td>
<td>1.02M</td>
<td>21.93</td>
<td>25.92</td>
<td>21.86</td>
<td>21.85</td>
</tr>
</tbody>
</table>

7.2 Translation Results

Table 2 shows the final translation results. First, it is interesting that gap-based decoding is better than dl-based in our joint framework, but slightly worse in baseline. This is because with target translation as a constraint in forced decoding, gap limit is more flexible to get better forced derivation (3), and thus more effective rules for better translation. But with a fixed rule set in baseline, its flexibility will introduce more noise into the beam, leading to worse performance.

Moreover, our gap-based joint learned model is better than gap-based baseline by 0.6 BLEU points, and better than dl-based baseline by 0.3 BLEU points.

7.3 Large Data

As previous experiment only trains word alignment on the small 100K corpus, which might create bad alignment and be unfair to baseline, we try another experiment and train word alignment on 2M sentence pairs, but translation model on the original 100K data. The result on the “All” test set is 24.46 for dl-based baseline (dl=6), and 24.57 for our gap-based joint learned model.

7.4 Discussions

The above two experiments have shown that our joint learning framework is comparable to the traditional MT training pipeline.

Based on the experiments, we conclude that three factors influence translation quality. First, in the training corpus, only about 82.9% of all sentence pairs are forced decodable, meaning that 17.1% sentence pairs are excluded from extracting useful rules. Second, due to the overfitting problem, we only use dense features here. Yu et al. (2013) has shown that if only use dense features, max-violation perceptron MT training cannot get good translation quality. At last, we need to handle the space of all possible phrase pairs, and the space of all possible decoding derivations. The search space is too large to optimize effectively.

However, even with these problems, we can still get comparable results with the baseline system, indicating that end-to-end MT training has a great potential to improve, and would be a promising direction for MT research.
8 Conclusion and Future Work

We present a pilot study on end-to-end MT training, and propose a joint framework combining rule induction, TM estimation, and parameter tuning in one module. Preliminary experiments show comparable translation quality to the conventional training pipeline.

Before we can fulfill the ambitious goal of end-to-end MT training, there are still a lot of things to do. In future, we will explore how to effectively use sparse features to improve the translation quality. We also plan to investigate the solution of scaling this framework to large data set. Currently, we need to solve two problems for scaling. The first one is how to deal with long sentences. The number of candidate phrase pairs is exponential to the length of training sentence pairs, making the learning process intractable for long sentence pairs. The practical solution is to segment them into several short ones. We can use IBM model 1 to do that, which has been demonstrated to be effective for SMT [Xu et al., 2005]. The second problem is how to use the large data set with millions of sentence pairs. Currently, parallelization seems to be the only solution. After segmentation, we can get millions of short sentence pairs, and then parallelize them to clusters for training.

9 Acknowledgement

We thank the three anonymous reviewers for the valuable comments, and Kai Zhao for discussions. This project was supported in part by DARPA FA8750-13-2-0041 (DEFT), NSF IIS-1449278, and a Google Faculty Research Award.

References


Automatic Detection of Antecedents of Japanese Zero Pronouns Using a Japanese-English Bilingual Corpus

Dong Zhan 1301211064@pku.edu.cn
School of Software and Microelectronics, Peking University, Beijing, 100871, China

Hiromi Nakaiwa nakaiwa@is.nagoya-u.ac.jp
Graduate School of Information Science, Nagoya University, Aichi, 464-8601, Japan

Abstract

In this paper we present a method of detecting zero pronouns in Japanese clauses and identifying their antecedents using aligned sentence pairs from a Japanese-English bilingual corpus and open resource tools. We use syntactic and semantic structures and the alignment of words and phrases in the sentence pairs to automatically detect zero pronouns and determine their antecedents using English translations. We build rules to link antecedents with zero pronouns and create filters to remove problematic sentence pairs. Experimental results confirm the effectiveness of our method. The proposed method allows the construction of an annotated corpus of zero pronoun sentences in which the antecedents of the missing pronouns are flagged. This would be very useful for machine translation (MT), because zero pronoun detection is a vital problem when translating languages which allow zero pronouns.

1 Introduction

In many languages, such as Japanese and Chinese, elements which can be easily deduced by the reader are frequently omitted from subsequent expressions in discourses. Omissions related to obligatory subject and object cases are often referred to as zero pronouns. A zero pronoun can be thought of as a noun phrase which is obligatory but which is not expressed explicitly. Zero pronoun resolution is more complicated than overt pronoun resolution because a natural language processing (NLP) system has to detect zero pronouns before it can identify their antecedents. Determination of the antecedents of zero pronouns is vital in many NLP applications, including information extraction, question answering, machine translation, etc.

Many algorithms have been proposed for pronominal anaphora resolution [9, 2, 10]. However, resolving the anaphora problem is much harder in languages which allow zero pronouns compared to languages which do not allow zero pronouns, such as English, since detection of zero pronouns should be done before anaphora resolution. Several methods have been proposed to solve this problem [12, 3, 14]. Some of these methods use semantic and pragmatic constraints, such as semantic constraints on cases, modal expressions, or verbal semantic attributes, to determine the referents of zero pronouns [12]. Other methods use a machine learning approach for anaphora resolution, such as using semantic attributes as computable features to perform identification and resolution

Most of these methods focus on zero pronoun resolution and detection of antecedents of zero pronouns as a crucial part of zero pronoun resolution. All of these methods employ either an annotated corpus containing zero pronoun sentences, with anaphoric relationships tagged by annotators [3, 4], or an annotated corpus provided by other persons or institutes [14, 1]. There are many ways to construct such an annotated corpus. Most of these corpora are monolingual and are annotated by hand. Manual annotation of these corpora can be very time consuming and may require relevant knowledge, so in this study we attempt to find a way to automatically detect the antecedents of zero pronouns.

Different languages contain different types of anaphoric expressions. Since similar languages often contain similar anaphoric expressions, languages which are very different from each other, such as English and Japanese, generally differ more markedly in regard to how they use pronouns and how these pronouns are linked to their antecedents. Thus, if we use a bilingual corpus, we are able to contrast how each language handles corresponding occurrences of subjects, objects and pronouns in the same sentence, and we can utilize the translation to identify the antecedents of zero pronouns.

Some research on automated corpus annotation using a bilingual corpus has been conducted [11]. The purpose of that study was to formulate rules for the anaphoric resolution of Japanese zero pronoun sentences using aligned sentence pairs. The same method was also used in another study, in which a valency transfer dictionary which contained 16,000 pairs of Japanese case-frame patterns was used for zero pronoun detection [6]. It showed the relationship between corresponding Japanese verbs and their English translations using different semantic models. For example, "yomu" in the patterns "N1(human)-ga N2(abstract thing)-o yomu" should translate as "N1 read N2" in English, but in the pattern "N1(subjects)-ga N2("vote")-o yomu" should translate as "N1 predicts (the outcome of the vote will be) N2". This dictionary makes it easy to find the obligatory elements of Japanese sentences and it can also be used to automatically recognize intransitive verbs. But since there were no open source parsers for this dictionary, this method of zero pronoun resolution could not be widely implemented. Several years have passed since this method was first proposed however, and in that time many useful tools for natural language processing have been created which have achieved good accuracy in tasks such as morphological analysis, parsing and alignment. These tools are widely used in the field of NLP and most of them are open source software. Therefore, in this study we propose a method of automatically annotating Japanese zero pronouns and their antecedents into a Japanese-English bilingual corpus, using open source tools.

2 Task Definition

For the purpose of machine translation, it is important to recognize that some grammatical elements not present in one language could be obligatory elements in another language. In particular, subjects and objects are often omitted in Japanese, but are obligatory in English. So it is natural to consider using the differences between Japanese and English to resolve the issue of missing pronouns in Japanese. A Japanese-English bilingual corpus could be useful if the antecedents of zero pronouns are difficult to recognize in Japanese sentences but these missing pronouns have equivalent antecedents in English sentences. For example, subject is zero pronoun in Japanese sentence of example (1) and it has a equivalent antecedent "I" in English sentence, then zero pronoun
can be aligned with equivalent antecedent in English.

(1) 私は疲れているとき、本を読みたい。

Literally translated, "When I am tired, want to read book." In grammatically correct English, it would be translated as, "When I am tired, (I) want to read a book." The first, bold "I" is the antecedent, and in this sentence it allows us fill in the "I" in parenthesis, which is the English equivalent of the missing zero pronoun in Japanese.

So in this study, we propose a method to detect the antecedents of Japanese zero pronouns using English translations of Japanese sentences. This method involves two steps. First, we need to detect whether or not the original sentence contains zero pronouns, and if it has zero pronouns we want to know where the zero pronouns occur. Then we try to detect the antecedents in the English translation and decide which antecedent corresponds to the missing zero pronoun. The translation process may result in some errors and unsuitable sentences, however, so we have to detect and remove unsuitable sentence pairs. Zero pronouns can be divided into three types, based on the location of the antecedent:
The antecedent of the zero pronoun is located in sentence further back, mostly in the previous sentence:

(2) 彼はとてもいい人です。昨日、(φ-ga) 私に手伝ってくれました。
He is a good person. Yesterday helped me.
The antecedent of the zero pronoun is located in the same sentence:

(3) 彼は宿題を終わって、(φ-ga) テレビを見ました。
He finished homework, watched TV.
The antecedent of the zero pronoun does not exist at all in context:

(4) (φ-ga) 東京にいきたい。 Want to go to Tokyo.

In this paper we focus on both simple sentences and complex sentences, which contain clauses in addition to one simple sentence. But to detect the antecedent of a zero pronoun within a clause of a complex sentence or in a previous sentence, we will need to combine our translation comparison method with other anaphora resolution methods. For example, our method can identify that the antecedent of the zero pronoun in the second clause of sentence (2) is "he", but it does not know that "he" refers to the character "彼" in the previous clause. If we can find the antecedent of a zero pronoun in an English translation, we can also determine its referent using anaphora resolution on the translation, and then align the referent and the corresponding Japanese word. This will allow us to resolve even intra-sentential zero pronoun anaphora. There have been many studies on anaphora resolution in English, but in this study we only focus on detecting the antecedents of zero pronouns in English translations of Japanese sentences.

3 Corpus

For this study, we used the Japanese-English Scientific Paper Excerpt Corpus (ASPEC-JE)\(^1\), which contains 3,008,500 parallel sentence translations collected from Japanese-English scientific paper abstracts. There are no syntactic or semantic structures associated with the sentences in this corpus, so we need to do segmentation, Part-Of-Speech tagging and parsing. We removed especially long sentences from the corpus (more than 150 characters in one sentence) because the syntactic structures of these sentences were too complex to be used. We removed 486,958 long sentences, and then used the remaining 2,521,542 sentence pairs to train the Japanese to English alignment model. To examine the effectiveness of this method, only 1,000 sentences were used for a closed test and 200 sentences for an open test, because this seems to be a reasonable number of

\(^1\)http://lotus.kuee.kyoto-u.ac.jp/ASPEC/
sentences which could be evaluated manually. All 1,200 of these sentences were chosen randomly.

4 Linking zero pronouns and their antecedents by aligning Japanese-English sentence pairs

Our method can be divided into several steps: 1) morphological analysis of the Japanese-English corpus (MeCab\(^2\)[8] for Japanese and Stanford segmentation\(^3\) for English); 2) alignment of Japanese words and English words (GIZA++\(^4\)[13]); 3) syntactic and semantic analysis of Japanese sentences; 4) syntactic analysis of English sentences; 5) identification of unsuitable sentence pairs; and 6) linking of Japanese zero pronouns with their antecedents. For Japanese syntactic and semantic analysis, we use two tools, CaboCha\(^5\)[7] for syntactic analysis and SynCha\(^6\)[5] for Japanese predicate-argument structure analysis. These tools allow us to recognize which part of a sentence is the subject, object or predicate and to identify the dependency relationships between word phrases. For English syntactic analysis, the Stanford parser\(^7\) is used. There are two types of output; context-free, phrase structure grammar representations and universal dependencies. We use the latter to build syntactic relationship structures. Overview of the process is shown in Figure 1.

4.1 Identification of antecedent candidates in English sentences

Before linking Japanese zero pronouns with their antecedents as indicated in English translations, we need to specify a range of antecedent candidates. Anaphoric expressions are common in English, and there is a set of special identity words which tend to often appear as antecedents. In order to not miss these common antecedents, we must consider not only definite nouns and noun phrases, such as "the company", but also these identifying words, as possible translation equivalents. These identifying words include personal pronouns (I, you, he, she, they, we, it); impersonal words (one, everyone, no one, men, women, human beings, people) and demonstratives (this, that, these, those, each, every).

\(^2\)http://taku910.github.io/mecab/
\(^3\)http://nlp.stanford.edu/software/tokenizer.shtml
\(^4\)http://www.statmt.org/moses/giza/GIZA++.html. We use alignments of translations in both directions (from Japanese sentences to English sentences and vice versa) and combine them for the final alignment. This means Japanese words or phrases and their English counterparts are only linked if they are aligned in the translation results in both directions.
\(^5\)http://taku910.github.io/cabocha/
\(^6\)http://www.cl.cs.titech.ac.jp/ryu-i/syncha/
\(^7\)http://nlp.stanford.edu/software/lex-parser.shtml
4.2 Identification of unsuitable sentence pairs

There are situations where the aligned sentence pairs are not suitable for annotating antecedents of zero pronoun. These sentences need to be identified and moved out. For example, when English parser was unable to make a full syntactic structure, we cannot use it for identification of antecedent candidates. Considering the accuracy rate of tools we used and the quality of the bilingual corpus, aligned sentence pairs are identified as unsuitable if they have the problems caused by corpus, such as freely translation, or errors caused by tools, like parser and alignment errors. For identifying these unsuitable sentences, we have made some rules to filter these sentences out automatically after having analyzed the result of original rules. The details of these filters will be explained in section 5.3.

4.3 Zero pronoun detection

Zero pronoun sentences occur frequently in Japanese and zero pronouns usually appear where either the grammatical subject or object would appear. Moreover, understanding the referent of the zero pronouns in these positions is vital for sentence comprehension. Hence, we focus on zero pronouns which appear as subjects and objects. Normally, a subject is an essential component of a complete sentence, unless the sentence is formulated in a passive voice ("The mailman was attacked.") or in the imperative mood ("Attack the mailman!"). If a Japanese sentence does not have a subject, and the sentence is not in the passive voice or imperative mood, we assume there is a zero pronoun in subject position. Regarding objects, our assumption is based on whether the predicate contains a transitive or intransitive verb. If the predicate of a Japanese sentence contains a transitive verb but there is no object, then it is assumed that there must be a zero pronoun in object position. These assumptions are the basis of zero pronoun detection in Japanese sentences. However, we are using an open source parser which cannot recognize intransitive verbs in Japanese sentences. Furthermore, the same verbs tend to be transitive or intransitive in English as in Japanese. But with the help of an English translation, we can recognize intransitive verbs in Japanese sentences by noting if the corresponding English verb in the sentence has no direct object.

4.4 Linking Japanese zero pronouns with their antecedents

Now that we have ways to filter out unsuitable sentences and to identify zero pronouns, we need a way to link an antecedent which appears in an English sentence to a zero pronoun in a Japanese sentence. Nakaiwa proposed ten rules for this purpose (1999), and we used these rules as our starting point. However, based on an analysis of the results and the features of the corpus, we decided to formulate new rules to improve performance. The ten rules proposed in Nakaiwa’s paper can be described as follows:

**Rule 1:** Alignment between subjects
**Rule 2:** Alignment between objects
**Rule 3:** Alignment of Japanese subjects and English possessive pronouns
**Rule 4:** Unalignment of Japanese subject when the passive voice is used in English
**Rule 5:** Non-alignment of Japanese subjects, and alignment of Japanese objects with English subjects, when a passive voice is used in English
**Rule 6:** Alignment of both subjects and objects
**Rule 7:** First default alignment rule for remaining unaligned zero pronoun
**Rule 8:** Second default alignment rule for remaining unaligned zero pronouns
**Rule 9:** Default unalignment rule for unaligned zero pronouns
**Rule 10:** Identification of intra-sentential antecedents of any aligned zero pronouns
These rules are applied from 1 to 10 in numeric order. If a rule is satisfied for a zero pronoun, the process will stop. Each rule has its own application conditions, and a rule is applied only if all of the conditions for the rule are satisfied. Details of Rule 4, Rule 5, Rule 7, and Rule 9 are shown below, since the results of these rules will be analyzed in the following section:

**Rule 4:** Non-alignment of Japanese subjects when a passive voice is used in English

IF Sj of Pj is an unaligned φ & Oj of Pj is aligned with Se of USe & Pe of USe is in a passive voice

THEN there are no antecedents of Sj because a passive voice is used in the English translation: unalignable

For example, in sentence pair (5), the process for Rule 4 is as follows:

(5) (φ-ga) 内部リレーの使い方を説明した。
The usage of the inside relay was explained.
Oj: 使い方 <-> Se: usage & Pe: "was explained" is a passive voice => Sj: φ-ga = unalignable

**Rule 5:** Non-alignment of Japanese subject, and alignment of Japanese object with English subject, when a passive voice is used in English

IF Sj of Pj is an unaligned φ & Oj of Pj is an unaligned φ & Pe of USe is in a passive voice & Se of USe is an antecedent candidate Ai

THEN there are no antecedents of Sj because a passive voice is used in the English translation: unalignable & align φ-Oj with Se

(6) (φ-ga)(φ-o) 外用剤で治療した。
It was treated by the external preparation.
Pe: "was treated" is a passive voice & Se: it is an antecedent candidate Ai => Sj: φ-ga = unalignable & Oj: φ-o <-> Se: it

**Rule 7:** Default alignment rule for remaining unaligned zero pronouns

IF there is only one unaligned zero pronoun case in Japanese and there are one or more unaligned candidate antecedents

THEN we determine the antecedent of zero pronoun based on the following priority: personal pronoun > one > demonstratives > definite NP

**Rule 9:** Default non-alignment rule for unaligned zero pronouns

IF there are any remaining cases of unaligned zero pronouns in Japanese

THEN these are determined to be φ-Cj's whose antecedents are not explicitly translated in USe and are marked as unresolved

### 5 Experiment

#### 5.1 Evaluation

We evaluated the proposed method using aligned sentence pairs from the ASPEC-JE corpus. A closed test was conducted to evaluate the performance of the rules proposed by [11]. After analyzing the results, we added several filters to remove unsuitable sentence pairs with errors caused by processing tools or which had inaccurate translations. We then added additional rules to detect the antecedents of zero pronouns. A closed test
was again performed on the same data sets, and the process was repeated several times. Finally, an open test was conducted on unseen data to evaluate the effectiveness of the proposed method with an unknown corpus.

### 5.2 Closed test of Nakaiwa method

We first evaluated the effectiveness of Nakaiwa’s original 10 rules for detecting of zero pronoun antecedents. Before the evaluation, sentences without zero pronouns should be removed, and according to the parsing results, 95 sentences without zero pronouns were removed automatically, since these types of sentences are not our target. The remaining sentences were then examined using Nakaiwa’s original ten rules. Some rules, such as Rule 5, Rule 6 and Rule 9, determine the alignment or non-alignment of zero pronouns more than once at the same time. Rule 5 and Rule 6 process zero pronouns in both the subject and object positions of the same sentence, for example. Rule 9 identifies sentences which are unalignable. In order to evaluate our results, we had to check the output manually. The results are shown in Table 1.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Times rule applied correctly/Times rule applied (Accuracy)</th>
<th>Number of zero pronouns determined and linked correctly/Number of zero pronouns determined (Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95/96 (99.0%)</td>
<td>95/96 (99.0%)</td>
</tr>
<tr>
<td>2</td>
<td>10/11 (90.9%)</td>
<td>10/11 (90.9%)</td>
</tr>
<tr>
<td>3</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>4</td>
<td>333/333 (100%)</td>
<td>333/333 (100%)</td>
</tr>
<tr>
<td>5*</td>
<td>17/104 (16.3%)</td>
<td>121/208 (58.2%)</td>
</tr>
<tr>
<td>6*</td>
<td>1/2 (50%)</td>
<td>2/4 (50%)</td>
</tr>
<tr>
<td>7</td>
<td>28/59 (47.5%)</td>
<td>28/59 (47.5%)</td>
</tr>
<tr>
<td>8</td>
<td>9/12 (75.0%)</td>
<td>9/12 (75.0%)</td>
</tr>
<tr>
<td>9</td>
<td>85/424 (20%)</td>
<td>85/502 (16.9%)</td>
</tr>
<tr>
<td>10</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>Totals</td>
<td>578/1041 (55.5%)</td>
<td>683/1225 (55.8%)</td>
</tr>
</tbody>
</table>

Table 1: Results using Nakaiwa’s original rules

*Correct results means both subject and object zero pronouns were correctly determined.

### 5.3 Analysis of results

From these results we can see that only 55.5% of applied rules determined zero pronouns and their antecedents correctly, and that the overall accuracy rate of zero pronoun and its antecedent determination was 55.8%. Rule 1, for determining zero pronouns in the subject position, achieved a precision rate of 99.0%, with the one incorrect result caused by a Japanese parser error. Rule 2, for determination of zero pronouns in the object position, achieved 90.9% precision. Rule 4, for detection of passive voices, achieved 100% precision. These results show that the original method is successful for many sentences. However, there were no or very limited cases in which Rule 3, Rule 6 and Rule 10 could be applied.

Rule 5, for both detection of zero pronouns in the object position and detection of passive voices, achieved only 58.2% precision for zero pronoun determination and Rule 9 achieved very poor precision (16.9%) for zero pronoun determination, which differs markedly from Nakaiwa’s own results (1999), in which accuracy rates of 92.9% and 44.8% for Rule 5 and Rule 9 were achieved, respectively. After checking the results for Rule 5, we found that in most cases the English translations were in a passive voice. These cases should have been processed by Rule 4, but went unrecognized because there are no objects of Japanese sentences which do not satisfy the condition of Rule 4. As
for Rule 9, its poor performance was due to its inability to detect intransitive verbs. Thus, we clearly need to adjust the rules in order to improve accuracy.

We also want to determine the recall of our method. To do this, we need to know the actual number of zero pronouns in the sentences used in the closed test. As mentioned in section 4.3, we can determine whether zero pronouns exist or not based on whether subjects or objects exist in the sentences. The number of zero pronouns in the subject position is counted automatically, but since our system cannot determine whether a Japanese verb is transitive or intransitive automatically, we need to check the number of zero pronouns in the object position manually. Based on whether or not the subject or object is missing, all of the sentences in the closed test can be divided into one of four types:

<table>
<thead>
<tr>
<th>Type of sentence (automatically determined)</th>
<th>Number of simple sentences</th>
<th>Zero pronouns in subject position</th>
<th>Number of zero pronouns in object position</th>
</tr>
</thead>
<tbody>
<tr>
<td>No zero pronoun</td>
<td>95</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Subject is missing, object exists</td>
<td>490</td>
<td>490</td>
<td>0</td>
</tr>
<tr>
<td>Subject is missing, object unknown</td>
<td>231</td>
<td>231</td>
<td>33</td>
</tr>
<tr>
<td>Subject exists, object unknown</td>
<td>278</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: Distribution of zero pronouns

The total number of zero pronouns in the closed test corpus can be calculated as follows: $490 + 231 + 33 + 8 = 762$. We can then calculate the recall of the result: $676/762 = 88.7\%$. Since there were many incorrect determinations of zero pronouns in our experiment, we tried to analyze these errors to improve accuracy. In our analysis of incorrect determinations of default results for Rule 5 (87 cases), Rule 7 (31 cases) and Rule 9 (339 cases), we found the following types of errors:

(A) There is no explicit antecedent in the English sentence due to faulty translation.
(B) Japanese parser error. Syntactic and semantic analysis may have caused errors in detecting semantic relationships.
(C) English parser error, such as syntactic and semantic analysis errors.
(D) Alignment error. Words that should be aligned cannot be found or were mismatched.
(E) No antecedent candidates.
(F) Cases where Japanese verbs were translated into continuous tenses or as gerunds in English, leading to incorrect predicate classification.
(G) Incorrectly detected zero pronouns in the object position.
(H) Intransitive verbs which act as predicates in Japanese sentences. In such cases, there is no object in the sentence but the system cannot detect this automatically. This is caused by the absence of an open source tool to determine if a Japanese verb is transitive or intransitive.

<table>
<thead>
<tr>
<th>Total</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule5</td>
<td>87</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>87</td>
<td>0</td>
</tr>
<tr>
<td>Rule7</td>
<td>31</td>
<td>15</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>Rule9</td>
<td>339</td>
<td>7</td>
<td>11</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Distribution of errors by type of error for Rule 5, Rule 7 and Rule 9

The statistical results are shown below in Table 3. According to the results, all of the Rule 5 errors were caused by incorrect detection of zero pronouns in the object position, and most of the Rule 9 errors were caused by faulty intransitive verb detection. Poor translations (type A) and parser errors (Type B and Type C) were responsible
for many of the errors. By analyzing these errors, we can adjust the rules to improve performance.

5.4 Modification of original rules

We found that all of the Rule 5 errors were caused by false detection of zero pronouns in the object position which did not exist, and that most of these sentences were translated into a passive voice. Such sentences should have been processed using Rule 4, but since these sentences do not have objects, they do not satisfy the conditions of Rule 4. This is the result of a syntactic difference between Japanese and English.

(9) 標記課題について検討した。 The problem with the title was examined.

For example, in sentence pair (9), the corresponding English translation is in a passive voice, and the noun phrase before “について” should be the object of the English sentence (the problem with the title). In Japanese “について” often connects noun phrases and predicates using the pattern “noun について verb”. The noun acts as an object in this case, but the Japanese parser cannot recognize it. So we developed a new rule to handle these situations, hoping this would improve our results. This new rule is not for the detection of antecedents of zero pronouns, but is instead a complement of tools to recognize whether there is a zero pronoun in the object position.

**Rule11:** Detection of zero pronouns as objects

IF Oj of Sj are unaligned φ & Cj in Noun+ について modifies the predicate

THEN replace φ-Oj with Cj

(10) 標記課題について検討した。 The problem with the title was examined.

Oj: φ-o & Nj (標記課題) について exists & Nj (標記課題) modifies 検討した。=>

Replace φ-o with Nj (標記課題)

<table>
<thead>
<tr>
<th>Rule</th>
<th>Times rule applied correctly/Times rule applied (Accuracy)</th>
<th>Number of zero pronouns determined and linked correctly /Number of zero pronouns determined (Accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95/96 (99.0%)</td>
<td>95/96 (99.0%)</td>
</tr>
<tr>
<td>2</td>
<td>10/11 (90.9%)</td>
<td>10/11 (90.9%)</td>
</tr>
<tr>
<td>3</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>4</td>
<td>471/471 (100%)</td>
<td>471/471 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>17/17 (100%)</td>
<td>34/34 (100%)</td>
</tr>
<tr>
<td>6</td>
<td>0/1 (0%)</td>
<td>0/1 (0%)</td>
</tr>
<tr>
<td>7</td>
<td>35/64 (54.7%)</td>
<td>35/64 (54.7%)</td>
</tr>
<tr>
<td>8</td>
<td>4/5 (80.0%)</td>
<td>4/5 (80.0%)</td>
</tr>
<tr>
<td>9</td>
<td>61/360 (16.8%)</td>
<td>61/371 (16.4%)</td>
</tr>
<tr>
<td>10</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>Totals</td>
<td>693/1025 (67.6%)</td>
<td>710/1053 (67.4%)</td>
</tr>
</tbody>
</table>

Table 4: Result of method with added rule

We insert this new rule between Rule 2 and Rule 3 (not showed in Table 4). We repeated the experiment. The results of this experiment are shown in Table 4. We can see that the total number of sentences decreased from 1,041 to 1,025. This is because in the 16 sentences in which a subject existed but an object did not exist, the object was added by applying Rule 11, so that no zero pronouns remain in these sentences. Thus, they will not be processed by the other rules and will not be counted in the total number of detected zero pronoun sentences. A total of 173 sentences were processed by Rule 11. The number of sentences processed by Rule 4 increased from 333 to 471, and the largest increase in processing occurred with Rule 5 (87 sentences), even though there are sentences which Rule 5 cannot be applied to, because they do not satisfy the...
condition of having antecedents in English, in which case the default rules are applied (51 cases). These sentences are now processed by Rule 4 after Rule 11 was added. As a result, sentences being processed by Rule 5 and by the default rules have decreased. We can also see that the total precision of zero pronoun determination increased from 61.6% to 67.4% and that recall increased from 88.7% to 93.2% (710/762).

5.5 Importation of filters

From our analysis of Table 3, we could see that many errors originated before processing, such as errors caused by incorrect translations. It is hard to avoid these types of errors, but we can add filters to detect problematic sentences and remove them in advance.

In some cases, part of the original Japanese sentence is translated into a noun phrase or gerund. Sentence pairs which had different numbers of unit sentences were rejected as unsuitable in Nakaiwa’s research (1999). We use a similar rule, but we only filter out unsuitable sentence pairs when English translation contains fewer unit sentences than the Japanese original so that if the English translation contains more unit sentences than the Japanese original, we still try to find the antecedent of a zero pronoun by trying to align the zero pronoun with possible antecedents in each of the unit sentences.

**Filter 1:** Filter for sentence pairs when English translation contains fewer unit sentences than the Japanese original. If English translation contains fewer unit sentences than the Japanese original, then mark the sentence pair as unsuitable.

(11) 柔軟性を重視してJAVAを用いた。JAVA was used because of its flexibility.

Pj: 2 unit sentences & Pe: 1 unit sentence => unsuitable

A large number of errors were caused by faulty detection of intransitive verbs, described as Type H errors in Table 3, thus we built a filter to determine if there is a zero pronoun in the object position.

**Filter 2:** Filter for zero pronoun in the object position with intransitive verb

If there is no object in the English sentence, and a subject and predicate exist in the Japanese sentence which are aligned with the English predicate, and no unaligned words remain, then there is no zero pronoun in object position.

(12) 画素構造の改造が（φ-o）進んでいる。Modification of the pixel structure is progressing.

Pj: 進んでいる <-> Pe: has advanced & Sj: 改造が exist & object not exist in English sentence => Sj: Obj is not zero pronoun

Filter 3 is used to detect errors caused by the limitations of Japanese language analysis tools. For example, in sentence(13), the object “事実” (“fact”) does exist in the Japanese sentence, but the Japanese parser cannot determine that the word before “は” is an object.

**Filter 3:** Filter for non-alignment of zero pronouns in the object position

If subject, object and predicate all exist in the English sentence and are aligned with Japanese words, and no words are remain to serve as antecedents of the object zero pronoun, then there is no zero pronoun in object position.
(13) この事実は著者が既にこのシリーズで (φ-ga) 論じてきた。 The author has already discussed this fact in this series. Pj: 論じてきた <-> Pe: has already discussed & Sj: 著者が <-> Se: The author & Oe: fact <-> other words in Japanese & no object left in English => unsuitable sentence pair &Sj: Obj is not a zero pronoun

**Filter 4:** Detecting mismatched sentence pairs. In a case where both the Japanese and English sentences contain sub-sentences, if the predicate and subject of the Japanese sub-sentences are aligned with elements of different sub-sentences in the English sentence, this means the sentence pairs are mismatched. If this occurs, we can reject the sentence pair as unsuitable and avoid searching for antecedents for zero pronouns.

<table>
<thead>
<tr>
<th>Filter</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of times filters were applied</td>
<td>8</td>
<td>214</td>
<td>34</td>
<td>1</td>
<td>256</td>
</tr>
<tr>
<td>Number of zero pronouns detected using each filter</td>
<td>18</td>
<td>214</td>
<td>34</td>
<td>2</td>
<td>267</td>
</tr>
</tbody>
</table>

Table 5: Number of sentence pairs processed and zero pronouns detected by each filter

These filters were applied in numerical order from Filter 1 to Filter 4. By applying these filters before applying our rules, we were able to remove some unsuitable sentences successfully, as shown in Table 5. We then reapplied our modified rules, and our new results are shown in Table 6. Due to filtering, the number of rules which were applied during processing was reduced, while also achieving a large increase in the precision of zero pronoun determination (from 67.4% to 92.2%). Recall remains unchanged at 93.2% because the number of correctly determined zero pronouns was unchanged.

<table>
<thead>
<tr>
<th>Rule</th>
<th>With added rule</th>
<th>With added rule and filters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rules applied correctly/Rules applied (Accuracy)</td>
<td>Zero pronouns determined and linked correctly/Zero pronouns determined (Accuracy)</td>
</tr>
<tr>
<td>1</td>
<td>95/96 (99.0%)</td>
<td>95/96 (99.0%)</td>
</tr>
<tr>
<td>2</td>
<td>10/11 (90.9%)</td>
<td>10/11 (90.9%)</td>
</tr>
<tr>
<td>3</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>4</td>
<td>471/471 (100%)</td>
<td>471/471 (100%)</td>
</tr>
<tr>
<td>5</td>
<td>17/17 (100%)</td>
<td>34/34 (100%)</td>
</tr>
<tr>
<td>6</td>
<td>0/1 (0%)</td>
<td>0/1 (0%)</td>
</tr>
<tr>
<td>7</td>
<td>35/64 (54.7%)</td>
<td>35/64 (54.7%)</td>
</tr>
<tr>
<td>8</td>
<td>4/5 (80.0%)</td>
<td>4/5 (80.0%)</td>
</tr>
<tr>
<td>9</td>
<td>61/360 (16.9%)</td>
<td>61/371 (16.4%)</td>
</tr>
<tr>
<td>10</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>Totals</td>
<td>693/1025 (67.6%)</td>
<td>710/1053 (67.4%)</td>
</tr>
</tbody>
</table>

Table 6: Results with added rule vs. results with added rule and filters

In Table 7, we analyze the distribution of different types of zero pronoun antecedents (based on the results of the added rule and filters, and only including correctly identified antecedent results). The majority of zero pronoun sentences are sentences written in a passive voice, which is due to the nature of the corpus. However, we can also see the distribution of explicit antecedents and specific nouns, the latter of which are especially valuable for accurately translating Japanese sentences into English.
Table 7: Distribution of zero pronoun antecedents determined using selected rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Number of zero pronoun</th>
<th>They/they</th>
<th>It/it</th>
<th>1</th>
<th>We/we</th>
<th>noun</th>
<th>Passive voice</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>95</td>
<td>27</td>
<td>20</td>
<td>0</td>
<td>11</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>471</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>471</td>
</tr>
<tr>
<td>5</td>
<td>34</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>10</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>35</td>
<td>1</td>
<td>7</td>
<td>4</td>
<td>22</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td>632</td>
<td>28</td>
<td>35</td>
<td>15</td>
<td>82</td>
<td>488</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Open test results: using original rules and using added rule with filters

<table>
<thead>
<tr>
<th>Rule</th>
<th>Zero pronouns determined and linked correctly/Zero pronouns determined (Accuracy)</th>
<th>Original rules</th>
<th>Added rules with filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15/16 (93.8%)</td>
<td>15/16 (93.8%)</td>
<td>15/15 (100.0%)</td>
</tr>
<tr>
<td>2</td>
<td>3/3 (100.0%)</td>
<td>3/3 (100.0%)</td>
<td>3/3 (100.0%)</td>
</tr>
<tr>
<td>3</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>4</td>
<td>53/53 (100.0%)</td>
<td>53/53 (100.0%)</td>
<td>71/71 (100.0%)</td>
</tr>
<tr>
<td>5</td>
<td>1/12 (8.3%)</td>
<td>23/24 (95.8%)</td>
<td>4/5 (80.0%)</td>
</tr>
<tr>
<td>6</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>7</td>
<td>3/11 (27.3%)</td>
<td>3/11 (27.3%)</td>
<td>4/10 (40.0%)</td>
</tr>
<tr>
<td>8</td>
<td>1/3 (33.3%)</td>
<td>1/3 (33.3%)</td>
<td>1/2 (50%)</td>
</tr>
<tr>
<td>9</td>
<td>9/105 (8.6%)</td>
<td>9/123 (7.3%)</td>
<td>9/16 (56.2%)</td>
</tr>
<tr>
<td>10</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
<td>0/0 (-)</td>
</tr>
<tr>
<td>Totals</td>
<td>95/203 (46.8%)</td>
<td>107/233 (45.9%)</td>
<td>107/122 (87.7%)</td>
</tr>
</tbody>
</table>

Table 8: Open test results: using original rules and using added rule with filters

The filters removed 67 unsuitable sentences containing 69 zero pronouns. To calculate recall, we counted the number of zero pronouns, as in Table 2. The total number of zero pronouns in each category was 129. Using the original rules, the precision of zero pronoun determination was 45.9% and recall was 107/129 = 82.9%. Using the added rule with filters, the precision of zero pronoun determination was 85.4% and recall was 111/129 = 86.0%. From these results, we can see that the improved method achieved higher precision and recall compared to the original method. Furthermore, our method was shown to be effective for detecting and resolving zero pronouns of Japanese sentences in unknown corpus. By applying this method to many aligned sentence pairs, it will be possible to collect large amounts of data on recognition of antecedents of zero pronouns, and that data can then be used in many natural language processing applications.

6 Conclusion

This paper proposes a method to detect and identify missing pronouns in written Japanese so they can be inserted into English translations. The proposed method uses aligned sentence pairs from a Japanese-English bilingual corpus and open source tools. Experimental results showed that our proposed method achieved a rule application accuracy rate of 93.3% for all sentence pairs and a 92.2% precision rate for zero pronoun
identification in a closed test (Table 6). We also achieved a rule application accuracy rate of 87.7% and a zero pronoun identification rate of 85.4% in an open test (Table 8). The effectiveness of this method will allow us to automatically construct an annotated, Japanese-English corpus which links zero pronouns in Japanese with their antecedents, which will greatly improve the accuracy of machine translations. In the future, we plan to combine our method with a machine learning technique to investigate Japanese anaphora resolution. We also plan to apply our method to Japanese-English bilingual corpora with other genres, allowing us to examine the full range of zero pronoun phenomena, and to Japanese-Chinese bilingual corpora, allowing us to examine the effectiveness of the proposed method with a bilingual corpus in which both languages have many zero pronouns. The effect of using different syntactic and semantic parsers will also be examined.

7 Acknowledgements

This research was carried out while Zhan Dong was an exchange student at Nagoya University, from September 2014 to July 2015. We would like to thank Prof. Kazuya Takeda for his helpful comments regarding this research.

References


Abstract

This paper proposes an extension of METEOR, a well-known MT evaluation metric, for multiple target languages using an in-house lexical resource called DBnary (an extraction from Wiktionary provided to the community as a Multilingual Lexical Linked Open Data). Today, the use of the synonymy module of METEOR is only exploited when English is the target language (use of WordNet). A synonymy module using DBnary would allow its use for the 21 languages (covered up to now) as target languages. The code of this new instance of METEOR, adapted to several target languages, is provided to the community. We also show that our DBnary augmented METEOR increases the correlation with human judgements on the WMT 2013 and 2014 metrics dataset for English-to-(French, Russian, German, Spanish) language pairs.

1. Introduction

Machine translation (MT) is the process of automatically translating a text in a source language into a corresponding text in a target language. In order to evaluate and compare the quality of several MT systems, we need to rate the translation hypothesis produced by each MT system either with the help of human experts (subjective evaluation) or compare it to pre-existing human translations (using automatic evaluation metrics, objective evaluation). In practice, subjective evaluation considers various aspects to grade the translation quality, such as adequacy, fluency, and intelligibility. However, subjective evaluation conducted a posteriori often costs too much (in term of human resources) and, thus, objective evaluation metrics (fast and cheap as long as references are available) are often preferred nowadays.

One drawback with automatic evaluation metrics is that they compare the MT hypothesis with few (and sometimes only one) reference translations. This is definitely not enough to capture lexical variation in translation. For this reason, metrics which exploit synonymy or stem similarities, such as METEOR (Banerjee and Lavie, 2005), exhibit better correlation with human judgement. METEOR maps words with the same stem or the same synset using lexico-semantic resources. However, so far, the full potential of METEOR is only exploited when English is the target language (use of WordNet).
Contribution This paper proposes an extension of METEOR for multiple target languages using a lexical resource called DBnary (Sérasset, 2015). DBnary is an extraction in RDF of the lexical data of multiple editions of Wiktionary. It has several millions of triples describing lexical entries of the extracted languages, and more than 4.6 million translations from 21 languages to more than 1500 target languages. The modified code allowing to call METEOR for new target languages (French, Russian, German, Spanish) is made available to the research community. More target languages (today 21 in total) could be plugged very quickly by interested users using the same lexical resource (DBnary notably includes Bulgarian, Dutch, English, Finnish, French, German, (Modern) Greek, Indonesian, Italian, Japanese, Latin, Lithuanian, Malagasy, Norwegian, Polish, Portuguese, Russian, Serbo-Croat, Spanish, Swedish, Turkish). We also present initial experiments on the WMT 2013 and 2014 metrics dataset and show that our new METEOR slightly increases correlation with human judgments of translation quality, for language pairs with a target language different than English.

2. State of the art

Since METEOR was first introduced in 2005, it has been improved and extended to include more features and accommodate more languages for a subset of its features.

2.1. METEOR: the basics

Banerjee and Lavie (2005) introduced METEOR to overcome several weakness of BLEU (Papineni, 2002) and NIST (Doddington, 2002) they identified as: the lack of recall, an indirect only measure of level of grammatical wellformedness, the lack of explicit word-matching between translation and reference, and the use of geometric averaging of n-grams.

The goal of METEOR was to aim for better correlation with human judgments of translation quality using not only word-to-word alignment between the translation hypothesis and the reference translation(s). The alignment is incrementally produced by a three-leveled mapping approach between the hypothesis and the reference, using additional resources if needed: exact match of the surface forms of the words, exact match of the computed stems of the words, synonymy overlap through shared WordNet “synset” of the words. The second mapping level uses a stemmer and the third mapping level uses English WordNet.

While no free WordNets are available for languages such as French, Spanish or German, current implementation of METEOR for such languages do not support the third mapping level.

2.2. METEOR: the recent extensions

METEOR-NEXT (Denkowski and Lavie, 2010a), was introduced to better correlate with human-targeted Translation Edit Rate (HTER) (Snover et al., 2006), a semi-automatic post-editing based metric which measures the distance between a MT hypothesis and its post-edited version. The goal was to go beyond a strictly world-level metric with a new aligner supporting phrases (multi-word) matches. Thus, a fourth mapping level was added to implement this new feature using a paraphrase database. For English, the database was developed by Snover (2009a). Later, Denkowski and Lavie (2010b), released paraphrase databases for Czech, German, Spanish and French.

In 2014, METEOR Universal was released (Denkowski and Lavie 2014) that enabled the construction of the paraphrase database using only the parallel corpora used to develop the MT system (which was not the case in 2010).
In order to prevent synonyms/paraphrases corresponding to different senses to be treated as semantically equivalent, Apidianaki and Marie (2015) proposed METEOR-WSD. The English references are further disambiguated and annotated using Babelfly (Moro et al., 2014) for several language pairs (French, Hindi, German, Czech and Russian to English). For their experiment, Apidianaki and Marie (2015) got a better segment-level Kendall’s τ correlation than METEOR for 4 language pairs when the paraphrase module was activated.

2.3. Lexical resources

2.3.1. WordNet

WordNet is a large lexical database for English, developed by linguists of Princeton University (Fellbaum, 1998). Nowadays, it has become an important and a very useful resource for NLP applications, such as machine translation, word sense disambiguation, cross-lingual information retrieval etc. WordNet links nouns, verbs, adjectives and adverbs to sets of cognitive synonyms (called synsets), where each synset represent a specific concept. Synsets are interconnected through conceptual semantic and lexical relations, including synonymy, antonymy, hyponymy etc. Note that words with multiple meanings belong to several synsets, and their senses are arranged by order of frequency. There are different versions of WordNet in languages other than English, such as Arabic WordNet, French WordNet, etc. However, these lexical resources in other languages are not freely available. As already said, METEOR uses WordNet to increase the chance of the MT output words to match the reference words.

The latest version of WordNet 3.0, contains in total 117 659 synsets: 82 115 noun synsets, 13 767 verb synsets, 18 156 adjective synsets and 3621 adverb synsets.

To Lemmatize forms, METEOR uses the Morphy-7WN\(^1\) function included in WordNet. This function uses a two-step process to find lemma of a particular word \(W\). Firstly, \textit{Morphy} checks for exceptions in a list (containing morphological transformations that are not regular). If \(W\) is not in exception list, \textit{Morphy} uses the rules of detachment for NOUN, VERB and ADJ categories (no rules applied to ADV). After each transformation, WordNet is searched for the resulting string in the syntactic category specified.

2.3.2. DBnary

DBnary is a multilingual lexical resource in RDF (Klyne & Carroll, 2004) collected at LIG (Sérasset, 2015). The lexical data is represented using standard vocabularies. The lexicon structure is defined using the LEMON vocabulary (McCrae et al., 2011). Most parts of speech informations are mapped to the Lexinfo or OliA standard vocabularies (Cimiano et al. 2011, Hellman et al. 2015), making it highly reusable in many contexts. It is available either as a set of downloadable files or as Linked Open Data directly accessible to browsers or applications. It may also be queried online using a public SPARQL endpoint.

The available lexical data is automatically extracted from 21 different language editions\(^2\) of Wiktionary, the dictionary counterpart of Wikipedia. Among available lexical data, one may find 2.9M \textit{lexical entries} (with parts-of-speech, canonical form for all of them, along with pronunciations when available and inflected forms for some languages). \textit{Lexical entries} are subdivided into 2.5M \textit{lexical senses} (with their definitions and some usage example).

\(^{1}\) MORPHY(7WN) manual page: https://wordnet.princeton.edu/man/morphy.7WN.html

\(^{2}\) Bulgarian, Dutch, English, Finnish, French, German, (Modern) Greek, Indonesian, Italian, Japanese, Latin, Lithuanian, Malagasy, Norwegian, Polish, Portuguese, Russian, Serbo Croat, Spanish, Swedish, Turkish
DBnary also contains more than 4.6M translations going from the 21 extracted sources languages to more than 1500 different target languages. Additionally, DBnary contains lexico-semantic relations (syno/antonymy, hypo/hypernymy and mero/holonymy and troponymy).

Table 1 shows the size of the data for languages involved in the experiments later reported in this paper. Additional figures are available on the DBnary public web site³.

<table>
<thead>
<tr>
<th>Language</th>
<th># of entries</th>
<th># of senses</th>
<th># of synonyms</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>620,369</td>
<td>498,451</td>
<td>35,437</td>
</tr>
<tr>
<td>French</td>
<td>322,018</td>
<td>416,323</td>
<td>36,019</td>
</tr>
<tr>
<td>Russian</td>
<td>185,910</td>
<td>176,335</td>
<td>31,345</td>
</tr>
<tr>
<td>German</td>
<td>104,505</td>
<td>116,290</td>
<td>33,282</td>
</tr>
<tr>
<td>Spanish</td>
<td>86,388</td>
<td>126,411</td>
<td>21,024</td>
</tr>
</tbody>
</table>

Table 1. Number of entries, senses, and lexico-semantic relations in DBnary for the target languages considered in this study.

3. METEOR-DBnary for multiple target languages

The principal goal of this study is to propose an evaluation metric that uses synonyms in order to improve MT evaluation for target languages other than English.

3.1. Resources prepared

As mentioned in the section 1, METEOR package uses the synset dictionary of WordNet, which is a rich resource of 147,306 unique synsets belonging to four categories (nouns, verbs, adjectives and adverbs) for English. To gather new external resources for our augmented METEOR, we downloaded and installed Dbnary dataset⁴ and set up a virtuoso-opensource⁵ server in order to interrogate Dbnary locally. Then, we launched SPARQL queries on DBnary in order to extract every synonymy relations in the database for English, French, Russian, German and Spanish. The result of the extraction is in the format of: lemma -> synonym. Next, we performed a processing to match the same format as WordNet (which is already compatible with METEOR). This treatment is to assign an ID for each lemma and to build a list of synonyms for each lemma under format ID eg. lemma -> ID_syn1 ID_Syn2 ID_Syn3.

METEOR computes its scores using WordNet as an external lexical resource. In order to measure the difference between the use of DBnary synonyms and the use of WordNet synonyms, we built two English synonym dictionaries extracted automatically from DBnary: one that contains the same four categories available in WordNet (called METEOR-DB-4-catg) and another containing all the existing categories in DBnary for English (called METEOR-DB-All-catg).

³ data, docs and examples are available at http://kaiko.getalp.org/about-dbnary/
⁴ http://kaiko.getalp.org/about-dbnary/dataset/
⁵ https://github.com/openlink/virtuoso-opensource
Table 2. METEOR-Baseline vs METEOR-DBnary for 2 systems picked up randomly from WMT14 data (French-English MT)

The results of Table 2 above show that METEOR-DBnary-4-catg and METEOR-Baseline (based on WordNet) both obtain very similar scores while the size of the WordNet dictionary is 2.5 times larger than that of DBnary (4-catg). Moreover, using all existing categories in DBnary, we notice an increase of +0.20% in the final score. In other words, slightly more synonym matches are obtained with the latter metric based on the full English DBnary.

3.2. Lemmatisation issues

For English, METEOR uses the Morphy-7WN function as well as an exception list, attempting to find the lemma of a given word. However, for other target languages, it is very difficult to identify rules in order to lemmatize an inflected form. Thus, for the moment, we use TreeTagger (Schmid, 1994) to lemmatize words for other languages.

In order to avoid relaunching TreeTagger for each new entry, we adopt a preprocessing step that is needed before launching our modified METEOR. During this step, TreeTagger is run on the full evaluation corpus to collect a list of unique words with their respective lemmas.

We compare the impact of the lemmatization tool used (Morphy vs TreeTagger) by METEOR on the same two systems of WMT 2014 (see Table 3 results).

Table 3. Impact of lemmatization; METEOR-Morphy vs METEOR-TTG for 2 systems picked up randomly from WMT14 data (French-English MT)

In Table 3, METEOR-TTG shows a slight increase in the score, compared to METEOR-Morphy, because TreeTagger lemmatizes all categories (including Adverbs), whereas Morphy lemmatizes only three categories (Noun, Verb and Adjective).

4. Correlation with human judgements

In order to evaluate the correlation of our proposed METEOR-DBnary with human judgements of machine translation outputs, we used the data from the WMT13 Metrics Shared Task (Machacek and Bojar, 2013) for English-to-Spanish MT, and from the WMT14 Metrics Shared Task (Machacek and Bojar, 2014) for French-English, English-French, English-German and English-Russian MT.
We present the results in a similar fashion as in the WMT metrics task methodology using the following metrics. More details and formulas can be found in (Machacek and Bojar, 2013) or (Machacek and Bojar, 2014).

- System-level using *Pearson* correlation coefficient between system ranking based on human judgments *versus* METEOR (we will use our augmented metric and compare it to the baseline METEOR).

- Segment-level using *Kendall*’s $\tau$ correlation between system ranking, at the sentence level, based on human judgments *versus* METEOR (we will use our augmented metric and compare it to the baseline METEOR).

Our results were obtained with two different configurations of METEOR:

- **METEOR-Baseline**: currently available METEOR Universal tool with the synonym module activated for English only (using the WordNet resource) - see table 4.

- **METEOR-DBnary**: our augmented-METEOR with the synonym module activated for English, French, Spanish, German and Russian, using our lexical resource DBnary - see table 4.

It is worth mentioning that, in order to use our new synonym dictionaries and evaluate our approach, we activated the synonym module in METEOR for the following languages: French Spanish Russian and German, by assigning a weight of 0.8 to each languages (same weight as for the English module).

<table>
<thead>
<tr>
<th>WMT14</th>
<th>WMT13</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FR-EN</strong></td>
<td><strong>EN-FR</strong></td>
</tr>
<tr>
<td><strong>EN-RU</strong></td>
<td><strong>EN-GE</strong></td>
</tr>
<tr>
<td><strong>EN-ES</strong></td>
<td></td>
</tr>
</tbody>
</table>

| METEOR-Baseline | .975 | .941 | .923 | .263 | .886 |
| METEOR-DBnary   | .973 | .943 | .928 | .320 | .895 |

**Table 4.** System-level correlations (*Pearson* Correlation Coefficient) between METEOR-Baseline (or METEOR-DBnary) and the WMT13/WMT14 human rankings.

<table>
<thead>
<tr>
<th>WMT14 $\tau$</th>
<th>WMT13 $\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FR-EN</strong></td>
<td><strong>EN-FR</strong></td>
</tr>
<tr>
<td><strong>EN-RU</strong></td>
<td><strong>EN-GE</strong></td>
</tr>
<tr>
<td><strong>EN-ES</strong></td>
<td></td>
</tr>
</tbody>
</table>

| METEOR-Baseline | .406 | .280 | .238 | .427 | .184 |
| METEOR-DBnary   | .406 | .284 | .240 | .435 | .187 |

**Table 5.** Segment-level correlations (*Kendall*’s $\tau$) between METEOR-Baseline (or METEOR-DBnary) and the WMT13/WMT14 human rankings.
Table 4 shows that the use of DBnary slightly improves the system-level correlations of the METEOR score to human judgments in all language pairs except for French-English. Table 5 shows the same trend for segment-level correlations which confirms that DBnary can be a useful resource for MT evaluation. The use of DBnary seems very promising for Russian and German as target languages.

Finally, Table 6 shows the absolute values of both METEOR (Baseline vs DBnary) for the same language pairs and for a system randomly chosen in the WMT datasets (system rbmt-1 is a rule-based machine translation system). As expected, METEOR score increases when used with DBnary since in that case the metric maps more words with the same meaning, using DBnary as lexical resource for synonymy.

<table>
<thead>
<tr>
<th></th>
<th>WMT 14</th>
<th>WMT13</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN-FR</td>
<td>50.94</td>
<td>36.21</td>
</tr>
<tr>
<td>EN-RU</td>
<td>38.06</td>
<td>49.88</td>
</tr>
<tr>
<td>EN-DE</td>
<td>41.51</td>
<td>51.04</td>
</tr>
<tr>
<td>EN-ES</td>
<td>50.67</td>
<td>51.04</td>
</tr>
</tbody>
</table>

Table 6: Comparison of METEOR-Baseline vs METEOR-DBnary (for system rbmt-1)

We present below some examples of matches obtained for METEOR-Baseline and for METEOR-Dbnary.

**Example 1 : EN-FR (system rbmt-1)**
- **Reference**: Si la personne la plus puissante d'Europe peut être visée, alors les dirigeants d'entreprise sont sûrement aussi des cibles potentielles.
- **Hypothesis**: Si la personne la plus puissante de l'Europe peut être visée, alors sûrement les chefs de file des affaires sont également les cibles potentielles.

- **Synonym match**: word → lemma → synonym list
  - dirigeants → dirigeant → [chef, maître, leader, directeur]
  - chefs → chef → [tête, maître, cuisinier, leader, maître_queux, patron]

  => the lemma “chef” exists in the synonym list of the word “dirigeant”, thus “dirigeants” and “chefs” are considered as synonyms.
  - aussi → aussi → [ainsi, également, itou]
  - également → également → [aussi, pareillement, de même, par ailleurs]

  => METEOR considers “aussi” and “également” as synonyms, because “aussi” belongs in the synonym list of “également” and “également” exists in the synonym list of “aussi”.

- **Segment score**:
  - METEOR-Baseline : 0.6762
  - METEOR-DBnary : 0.7290

**Example 2: EN-FR (system rbmt-1)**
- **Reference**: J'estime qu'il est concevable que ces données soient utilisées dans leur intérêt mutuel.
- **Hypothesis**: Je pense qu'il est concevable que ces données soient employées pour le bénéfice mutuel.
Synonym match: word → lemma → synonym list
- employées → employer → [occuper, utiliser]
- utilisées → utiliser → [user]

During word-to-word alignment, METEOR considers the words “utilisées” (in REF) and “employées” (in HYP) as synonyms, because in the step of synonym match, we find that the lemma “utiliser” exists in the synonym list of the lemma “employer”.

Segment score:
METEOR-Baseline : 0.6609
METEOR-DBnary : 0.7133

Example 3: EN-FR (system rbmt-1)
- Reference: Il me parlait, m’encourageait constamment, il habitait mon corps.
- Hypothesis: Il me parlerait, m’encouragent constamment, il a vécu dans mon corps.

Synonym match: word → lemma → synonym list
- habitait → habiter → [occuper]
- vécu → vivre → [habiter, nourriture]

The lemma “habiter” exists in the synonym list of the word “vivre”, thus “habitait” and “vécu” are considered as synonyms.

Segment score:
METEOR-Baseline : 0.6743
METEOR-DBnary : 0.7688

5. Conclusion

We proposed an extension of METEOR, a well-known MT evaluation metric, for multiple target languages using our in-house lexical resource called DBnary. Our augmented METEOR obtained a better correlation with human judgements than the baseline METEOR, on the WMT 2014 metrics dataset for English-to-(French, Russian, German, Spanish) language pairs. The modified code allowing to call METEOR for new target languages (French, Russian, German, Spanish) is made available to the research community from the following link (http://kaiko.getalp.org/about-dbnary/meteor-with-dbnary/).

In a near future, more target languages (today 21 in total) could be plugged very quickly by us or by interested users (please contact us if you want to contribute) using the same lexical resource (DBnary). The same adaptation of synonym matches could be done to TER-Plus (Snover et al., 2009b). Finally, using WSD, such as done in (Apidianaki and Marie, 2015), is another interesting avenue for improving correlation between automatic evaluation metrics and human judgements.
References

Proceedings of the 9th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST'9), 4 June, Denver, Colorado, pp. 49--51.


Japanese Controlled Language Rules to Improve Machine Translatability of Municipal Documents

Rei Miyata  
rei@p.u-tokyo.ac.jp  
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan

Anthony Hartley  
a.hartley@rikkyo.ac.jp  
Rikkyo University, 3-34-1 Nishi-Ikebukuro, Toshima-ku, Tokyo, 171-8501, Japan

Cécile Paris  
Cecile.Paris@csiro.au  
CSIRO, Crn Vimiera and Pembroke Roads, Marsfield NSW, 2122, Australia

Midori Tatsumi  
tatsumi@rikkyo.ac.jp  
Rikkyo University, 3-34-1 Nishi-Ikebukuro, Toshima-ku, Tokyo, 171-8501, Japan

Kyo Kageura  
kyo@p.u-tokyo.ac.jp  
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan

Abstract

We report on experiments to test the effectiveness of controlled language (CL) rules on texts from Japanese municipal websites. We compiled a set of rules by trial and error, systematically rewriting Japanese source texts and analysing the machine translation (MT) outputs. We then employed native English speakers with little knowledge of Japanese as human evaluators and tested the understandability and accuracy of the English machine translated text. Comparing the results of four MT systems showed that the effectiveness of CL rules varies depending on the particular MT systems. A preliminary selection of optimal rules for each system showed more than 15% increase in MT performance. We also assessed the readability of the Japanese source texts and discuss the way to make them compatible with the quality of the MT outputs.

1 Introduction

It has become increasingly important for Japanese municipalities to provide information in multiple languages. Although combining machine translation (MT) and human post-editing has increasingly proven to be an effective MT workflow, it may not be an option for the websites of municipalities, which must offer a large volume of information, such as daily announcements, procedural guidelines, emergency information and policy white papers, with the need for frequent updates. Having all these documents translated into a number of languages using human translators or post-editors is simply too expensive and time consuming. In such cases, raw machine translation output is often provided on-demand.

Translations between languages that have completely different structures such as Japanese and English are, however, still difficult compared to those between European languages or between Japanese and Korean, and in many cases the quality of raw outputs of MT is of low or even not understandable quality. When post-editing is not a standard option, and the machine translation system is used as a ‘black box’, controlled authoring remains the key to the improvement of translation quality. Controlled authoring involves a suite of technologies and environments that provide document templates, glossary management, grammar and style checkers. In the current study, we focus on controlled language (CL) rules as a starting point. We chose En-
lish as the target language since English is still an overwhelming choice in translating Japanese municipal texts, followed by Chinese, Korean and Portuguese (Carroll, 2010).

In our scenario, we assume that effective CL rules should (i) help to raise the quality of MT output, and (ii) not degrade the quality of the Japanese source texts. To achieve both of these requirements, we devised empirical procedures to formulate CL rules and conducted an evaluation to assess the efficacy of each rule in terms of machine translatability and Japanese readability.

The remainder of this paper is structured as follows. We describe related work in Section 2. In Section 3, we discuss how we constructed our CL rules, while Section 4 explains our experimental setup for human evaluation to assess the rules. We present our results accompanied with a discussion in Section 5. Section 6 concludes with implications for future work.

2 Related studies

Controlled (natural) language or C(N)L is ‘a constructed language that is based on a certain natural language, being more restrictive concerning lexicon, syntax, and/or semantics while preserving most of its natural properties’ (Kuhn, 2014, p.123). A number of English CL rule sets have been proposed to improve MT performance as well as human comprehension, and they have been actually implemented, mainly in technical documentation (e.g., Kamprath et al., 1998; Nyberg et al., 2003). Evaluation experiments on CL for MT have also been undertaken to assess machine translatability and post-editing productivity (Pym, 1990; Bernth and Gdaniec, 2001; O’Brien and Roturier, 2007; Aikawa et al., 2007), showing evidence of the effectiveness of CL.

In the case of Japanese CL, Nagao et al. (1984) devised a controlled grammar to syntactically disambiguate Japanese sentences. Yoshida and Matsuyama (1985) also advocated the need for Japanese CL in parallel with MT development and conducted pioneering work. Little practical implementation, however, resulted from their efforts. One of the reasons for this setback in Japanese CL is the difficulty in producing significant results because the machine translation task from Japanese to another major language such as English is hard, compared to the task of automatically translating between European language pairs such as English and French.

From the 1990’s to the 2000’s, a number of studies in Japanese computational linguistics have addressed the automatic rewriting (or pre-editing) and paraphrasing for MT (Kim and Ehara, 1994; Shirai et al., 1998; Inui and Fujita, 2004). However, fully automatic reformulation of natural language copes badly with highly complex expressions, which tend to require contextual information. The scope of variations in linguistic patterns was, therefore, limited.

More recently, Ogura et al. (2010) proposed ‘Simplified Technical Japanese’ (STJ) to improve MT performance. They constructed the rules by (i) identifying linguistic patterns which appeared related to MT output quality, (ii) defining putative rules, and (iii) conducting a preliminary assessment of their efficacy. They finally formulated the STJ rule set consisting of about 50 rules, while pointing out that it does not comprehensively cover the range of Japanese expression patterns. Another attempt to create Japanese CL is the on-going ‘Technical Japanese’ project, which focuses mainly on documents for business purposes. It published a ‘Patent Documents Writing Manual’, which consists of 31 rules designed to improve the clarity and translatability of patent texts (Matsuda, 2014).

While recent work focused mainly on technical documents for industry and business, Tatsumi et al. (2013) formulated 22 CL rules for municipal website documents drawing on existing wisdom about technical writing in Japanese. Since the study chiefly referred to writing guidelines intended for human understandability or readability, the overall efficacy of the rules with MT was not significant. Thus, there remains much room for investigating other patterns impact-
O’Brien (2006) argued persuasively for the need to tune CL rule sets to language pair and MT system. The results of evaluation experiments by (Hartley et al., 2012) also suggested there were differences between rule-based machine translation (RBMT) and statistical machine translation (SMT) systems in terms of the impact of specific CL rules on their performance, although the MT systems as such were not the focus of their investigation. In short, it is still uncertain to what extent CL rules can be effectively generalised across MT systems or how much improvement can be attained if we compile rules specifically tuned to a given system.

Practical deployment of CL requires that the readability of the source text (ST) should not be compromised in the interests of making it more tractable for MT in order to improve target text (TT) quality. Indeed, the stated aim of Technical Japanese mentioned above is to improve both readability and machine tractability (Watanabe, 2010). However, when Hartley et al. (2012); Tatsumi et al. (2013) investigated the efficacy of CL rules with respect to both the readability of the Japanese source and the quality of the MT output, they observed for some rules a ‘trade-off’ between ST and TT quality, which needs further exploring.

3 Controlled language rules

3.1 CL formulation protocol

Given that municipal texts should serve readers of Japanese and English equally, any CL rule we propose here should contribute to the quality of MT outputs without degrading that of source texts. We assume that comparing original source texts and more machine translatable ones rewritten by humans enables us to derive insights into how to (re)write texts amenable to MT, while still guaranteeing source text readability, as long as human authors take charge of the whole rewriting process.

To materialise our assumption above, we devised the following empirical protocol to detect linguistic or textual features potentially effective for MT performance:

1. Rewrite a source text aiming at a better quality of MT outputs.
2. Record how the text was changed and assess the quality of the outputs.
3. Repeat steps 1 and 2, until achieving satisfactory quality of the MT outputs.

Examples of the detected linguistic features are shown in Table 1.

<table>
<thead>
<tr>
<th>ST</th>
<th>MT</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>電力会社に連絡、使用開始手続き完了後、プレーカーのスイッチを入れます。</td>
<td>You can turn on the contact, the procedures after completion of the electric power companies.</td>
<td>[original sentence]</td>
</tr>
<tr>
<td>電力会社に連絡します。使用開始の手続きが完了した後、プレーカーのスイッチを入れます。</td>
<td>Will contact the electric power company. Procedures for activation is complete, you turn on the breaker.</td>
<td>Split sentence Add ‘します’ Add ‘の’ Expand ‘完了後’</td>
</tr>
<tr>
<td>電力会社に連絡してください。使用開始の手続きが完了した後に、プレーカーのスイッチを入れてください。</td>
<td>Please contact the electric power company. Please turn on the breaker after the procedure of the activation is completed.</td>
<td>Change ‘ます’ to ‘てください’ Add particle ‘に’</td>
</tr>
</tbody>
</table>

Table 1: Example of rewriting the source text

3.2 CL formulation for municipal documents

To formulate CL rules for Japanese municipal documents through this protocol, we first extracted 100 original Japanese sentences from municipal websites and one of the authors con-
ducted the above protocol using three MT systems, i.e., one RBMT system (TransGateway\(^1\)) and two SMT systems (Google Translate\(^2\) and Minnano Jido Hon'yaku\(^3\)). We then summarised the linguistic and textual features expected to have an impact on MT quality and classified them into five categories: Mood/Modal, Structural, Lexical, Textual/Orthographical, and Terminological. In this study, we focus on Structural, Lexical, and Textual/Orthographical categories. We adopted a total of 38 features (Table 2) which had not been covered in the set of 22 CL rules proposed by (Tatsumi et al., 2013) and formulated 38 CL rules to regulate these features, such as ‘Avoid using multiple verbs in a sentence’ (rule 1).

In this phase, we did not take the variability of the systems into account, and extracted a wide range of features which we assumed could improve MT performance. These rules are not necessarily effective for every MT system. In formulating them, we observed that a few rewriting rules intended for one MT system had no effect or even a negative effect on the others. We thus conducted a quantitative human evaluation to determine which rules were effective for which MT system(s).

<table>
<thead>
<tr>
<th>Structural</th>
<th>20. omission</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. multiple verbs in a sentence</td>
<td>21. suffix</td>
</tr>
<tr>
<td>2. lack of subject</td>
<td>22. particle Made (まで)</td>
</tr>
<tr>
<td>3. lack of object</td>
<td>23. particle De (で)</td>
</tr>
<tr>
<td>4. connection</td>
<td>24. particle No (の) to mean ‘by’ or ‘from’</td>
</tr>
<tr>
<td>5. particle Ga (が) for object</td>
<td>25. per A</td>
</tr>
<tr>
<td>6. enumeration A-Mo, B-Mo (Aも、Bも)</td>
<td>26. particle Te (て)</td>
</tr>
<tr>
<td>7. Te-kuru (てくる) / Te-iku (ていく)</td>
<td>27. if particle To (と)</td>
</tr>
<tr>
<td>8. inserted adverbial clause</td>
<td>28. particle He-Ha (へは)</td>
</tr>
<tr>
<td>9. ending clause with noun</td>
<td>29. particle Ni-Ha (には)</td>
</tr>
<tr>
<td>10. Sahen-noun(^4) + auxiliary verb Desu (です)</td>
<td>30. particle No-Ka (のか)</td>
</tr>
<tr>
<td>11. attributive use of Shika-Nai (しか-ない)</td>
<td>31. demonstrative pronoun (ko-so-a-do)</td>
</tr>
<tr>
<td>12. verb + You-ni (ように)</td>
<td>32. particle Ni (に)</td>
</tr>
<tr>
<td>13. A or not</td>
<td>33. Japanese Kana / Chinese Kanji</td>
</tr>
<tr>
<td>14. Sahen-noun + Wo (を) + Suru (する)</td>
<td>34. bullet mark</td>
</tr>
<tr>
<td>15. Sahen-noun + Sare-ru (される)</td>
<td>35. machine dependent character</td>
</tr>
<tr>
<td></td>
<td>36. punctuation (sentence separation)</td>
</tr>
<tr>
<td></td>
<td>37. square bracket</td>
</tr>
<tr>
<td></td>
<td>38. wave dash (〜)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>16. particle Nado (など/等)</td>
<td>40. bullet mark</td>
</tr>
<tr>
<td>17. giving and receiving verb</td>
<td>41. machine dependent character</td>
</tr>
<tr>
<td>18. redundant word</td>
<td>42. punctuation (sentence separation)</td>
</tr>
<tr>
<td>19. compound word</td>
<td>43. square bracket</td>
</tr>
</tbody>
</table>

Table 2: A list of features to be regulated

4 Experimental setup

The aim of the evaluation was (1) to gauge how effective our CL rules are to different MT systems, and (2) to investigate whether the rules which contribute to MT performance also maintain source readability. Using texts from a municipal website and four MT systems, we assessed these two parameters through human evaluation.

---

\(^1\)http://www.kodensha.jp/platform/
\(^2\)https://translate.google.com
\(^3\)https://mt-auto-minhon-mlt.ucri.jgn-x.jp
\(^4\)A Sahen-noun is a noun which can be connected to the verb Suru (する) and act as a verb.
4.1 Data
We extracted 11,075 sentences with no more than 70 characters from the Toyohashi City website\(^5\) in five categories (public information, Q&A, department information, news articles, and topical issues), and selected sentences violating our 38 rules – four sentences for each rule – resulting in a total of 152 Japanese-Original (JO) sentences. One of the authors rewrote all 152 sentences in accordance with each rule, so generating 152 Japanese-Rewritten (JR) sentences.
We used four MT systems, two commercial RBMT systems (The Hon’yaku\(^6\) and TransGateway, hereafter, systems A and B) and two freely available SMT systems (Google Translate and Minnano Jido Hon’yaku, hereafter, systems C and D), without user dictionaries or any sort of customisation, to translate the 152 JO and 152 JR sentences into English. The result was 1,216 machine translated sentences: 152 sentences for each label AO (system A-Original), AR (system A-Rewritten), and so on – BO, BR, CO, CR, DO and DR.

4.2 MT quality evaluation
Our main interest in evaluating the MT quality was to assess whether or not the translation was understandable in terms of practical use of information. We followed a simple method proposed by (Tatsumi et al., 2013), which focuses on understandability at an acceptable level, disregarding grammatical and lexical errors as long as they do not impair the reader’s comprehension.

Questionnaire design
In order to find out whether or not an MT output was judged understandable and, if so, whether or not the reader’s understanding was in fact correct, we adopted a two-step evaluation method. In Step 1, we showed the judges an MT output without telling them that it was the result of MT, and asked them to indicate how well they understood the text, and how much effort was required to understand it, by selecting one of the following options:

1. I understood fully what this sentence is saying, after reading it once.
2. I understood fully what this sentence is saying, after reading it more than once.
3. I understood partially what this sentence is saying, after reading it more than once.
4. I have no idea what this sentence is saying even after reading it more than once.

In Step 2, the judges were shown a human translation (HT) corresponding to the MT output shown in Step 1 as an ‘alternative translation’. The question asked at that point differed depending on the answer to the question in Step 1. If [1] or [2] had been selected, the judges were asked to indicate how close the meaning of the new sentence was to the first sentence (the MT output) by selecting one of the following options:

5. Exactly the same meaning
6. Mostly the same meaning
7. Partly the same meaning
8. Completely different meaning

Considering that the judge’s memory from Step 1 might not last long enough to compare their understanding at Step 2, we showed the MT output again here. At the same time, in order to discourage direct comparison between the two texts, the judges were asked to compare only the general meaning, not focusing on the difference in word choice.

If [3] or [4] had been selected at Step 1, i.e., when the MT output was not understandable, it was of little use to know whether their understanding was correct or not. Instead, we needed to know if it was because of the bad quality of the MT or a problem with the content itself. The judges were shown the HT as an alternative. They were then asked to indicate how much of it

---

\(^5\)`http://www.city.toyohashi.lg.jp
\(^6\)`http://pf.toshiba-sol.co.jp/prod/hon_yaku/
they understood and how much effort was required to do so, with the same options as at Step 1, i.e., [9]=[1], [10]=[2], [11]=[3], [12]=[4]. In this case, the first sentence (the MT output) was no longer shown.

**Implementation**

We employed 24 judges, all adult English native speakers. They were all living in Japan, yet had little knowledge of Japanese. The judges were thus highly representative of the intended readers of the target texts.

Each judge was asked to evaluate 152 MT sentences that corresponded to the 152 Japanese source sentences but were a mix of translations from eight sources (AO, AR, BO, BR, CO, CR, DO, DR). The evaluation was conducted online.

### 4.3 Japanese readability evaluation

The Japanese source texts and their manually rewritten versions were both of understandable quality from the outset. In order to deal with the subtle differences in the sentence readability, we adopted the following evaluation method following (Hartley et al., 2012): the judges were presented with pairs of sentences JO and JR, whose ordering was randomised. Each judge was asked to evaluate each sentence of the pair on a four-point scale: easy to read; fairly easy to read; fairly difficult to read; difficult to read. We instructed the judges in advance not to focus on the minute grammatical exactness, but to judge the ease of reading.

For this task, we recruited three Japanese native speaker university graduate students as judges. Each judge evaluated 152 pairs of Japanese sentences, i.e., all 152 JO and 152 JR.

### 5 Results and discussions

#### 5.1 MT quality

**Overall result**

Firstly, we classified the results into four categories (Table 3). The numbers in square brackets correspond to the choices in each step as described in Section 4.2. An MT output is considered useful only when either [5] or [6] was chosen at Step 2 (MT–Useful) of the evaluation task described above. Those classed [7] or [8] are the dangerous instances, as they mean that the MT output is understandable while conveying inaccurate information (MT–Inaccurate). In the case of [9] or [10], the MT output is not intelligible and is thus useless. Yet it is less dangerous than [7] and [8]. Finally, [11] and [12] mean that even the human translation was not understandable. This may suggest that the problem lies in the content which requires some domain knowledge or contextual information for full understanding.

<table>
<thead>
<tr>
<th>Selected option</th>
<th>Category</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[5][6]</td>
<td>MT–Useful</td>
<td>The reader understood the MT output and their understanding was correct</td>
</tr>
<tr>
<td>[7][8]</td>
<td>MT–Inaccurate</td>
<td>The reader understood the MT output but their understanding was not correct</td>
</tr>
<tr>
<td>[9][10]</td>
<td>MT–Unintelligible</td>
<td>The reader did not understand the MT output, but they understood the corresponding HT</td>
</tr>
<tr>
<td>[11][12]</td>
<td>HT–Unintelligible</td>
<td>The reader did not understand either the MT or the corresponding HT</td>
</tr>
</tbody>
</table>

Table 3: Result categories

Table 4 shows the percentage of judgements that fell into each category. Overall, applying the CL rules increased the percentage of [MT–Useful] by around 3–4% for three systems, A, B.
and C, while system D shows a slight decrease. Although our focus is not on the analysis of the rules as a whole but on the diagnosis of the efficacy of each rule, it is worth noting that only about 30% of the MT outputs were deemed useful even after applying each CL rule.

<table>
<thead>
<tr>
<th>Label</th>
<th>MT–Useful</th>
<th>MT–Inaccurate</th>
<th>MT–Unintelligible</th>
<th>HT–Unintelligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>AO</td>
<td>27.4%</td>
<td>4.6%</td>
<td>62.1%</td>
<td>5.9%</td>
</tr>
<tr>
<td>AR</td>
<td>30.9%</td>
<td>5.5%</td>
<td>58.8%</td>
<td>4.8%</td>
</tr>
<tr>
<td>BO</td>
<td>23.2%</td>
<td>5.0%</td>
<td>66.0%</td>
<td>5.7%</td>
</tr>
<tr>
<td>BR</td>
<td>27.2%</td>
<td>5.7%</td>
<td>63.4%</td>
<td>3.7%</td>
</tr>
<tr>
<td>CO</td>
<td>26.5%</td>
<td>3.9%</td>
<td>64.7%</td>
<td>4.8%</td>
</tr>
<tr>
<td>CR</td>
<td>30.0%</td>
<td>6.8%</td>
<td>58.3%</td>
<td>4.8%</td>
</tr>
<tr>
<td>DO</td>
<td>27.0%</td>
<td>6.4%</td>
<td>61.0%</td>
<td>5.7%</td>
</tr>
<tr>
<td>DR</td>
<td>26.3%</td>
<td>6.8%</td>
<td>60.1%</td>
<td>6.8%</td>
</tr>
</tbody>
</table>

Table 4: Overall results of MT quality

A comparison between the four MT systems shows the different effects of the CL rules on the four MT systems. While in the case of systems A and B the application of the CL rules mostly resulted in decreasing the number of [MT–Unintelligible] and increasing that of [MT–Useful], in the case of system C, it resulted in a notable increase in the number of [MT–Inaccurate]. The following example shows system C generating an [MT–Inaccurate] output after the application of rule 14.

**Rule 14: avoid using Sahen-noun + Wo (を) + Suru (する)**

CO: At the venue, the video for each of the agenda is projected on a large screen, was the description of the agenda in an easy-to-understand local residents.

CR: At the venue, the video for each of the agenda is projected on a large screen, I explained the agenda in an easy-to-understand local residents.

HT: At the venue, images related to each topic were projected on a large screen and local residents explained the topics in an easy to understand manner.

We changed only ‘説明-をしました’ (‘gave an explanation’) into ‘説明-しました’ (‘explained’) in the source. System C then incorrectly (and unexpectedly) inferred the subject ‘I’ though the true subject ‘地城住民’ (‘local residents’) is present in the source, and generated an understandable but misleading output. This kind of unpredictable change not directly related to the feature in question tends to occur with SMT systems.

Importantly, the score of around 5% in the [HT–Unintelligible] category shows that even human translated sentences are sometimes not understandable, implying a fundamental difficulty with evaluating at the sentence-level. We had instructed the human translator to translate the source sentences without adding explanations or suppressing information to make them comparable with the MT outputs. Moreover, we did not show the judges the context of each sentence, so the occasional failure of judges to understand the human translation even though it was grammatically correct could be due to a lack of knowledge of the Japanese municipal domain.

**Generally applicable CL rules**

To diagnose the effectiveness of each CL rule for different MT systems in detail, we focused on the [MT–Useful] cases, since an increase in [MT–Useful] mostly corresponds to an decrease in [MT–Unintelligible], except for system C, which showed a significant rise in [MT–Inaccurate] together with [MT–Useful]. We counted the judgements that fell in this category for each rule and calculated the improvement (or degradation) scores as a percentage, emphasising the improvements in bold (Table 5).
Table 5: Improvement in [MT–Useful] category

We can see that four rules – 10, 13, 25 and 35 – have a positive effect on all four MT systems. Some examples of translations of the original sentences and their rewrites are listed below.

**Rule 10: avoid using Sahen-noun + Desu (です)**

**AO:** An admission ticket is the 10:00 a.m. sales start on Fri., April 22.

**AR:** An admission ticket starts sale at 10:00 a.m. on Fri., April 22.

**BO:** An admission ticket is sales starting on Friday, April 22 at 10:00am.

**BR:** An admission ticket begins to sell it at 10:00am on Friday, April 22.

**CO:** Admission ticket is sales start at April 22 (gold) 10 am.

**CR:** Tickets will start April 22 (Friday) at 10 am selling.

**DO:** Admission ticket is April 22 (Kim) 10 a.m. launch.

**DR:** Tickets will be on sale at 10 a.m. on April 22 (Kim).

**HT:** Ticket sales will start at on Friday, April 22 at 10:00 AM.

In this case, we rewrote Sahen-noun + Desu construction ‘開始-です’ into Sahen-noun + Suru construction ‘開始-します’, which resulted in more natural expressions in the MT outputs.

Rules 2, 4, 8, 12, 20, 28 and 37 show a positive effect on three systems and can also be regarded as generally applicable rules. We provide below examples of the application of one of these rules.

**Rule 2: avoid omitting subject**

**AO:** A home and the community are places where a child spends much time daily, and study that it is various in a life.

**AR:** A home and the community are places where a child spends much time daily, and a child studies that it is various in a life.
BO: A house and an area are the place where a child spends much time daily, and various things will be learned in the life.
BR: A house and an area are the place where a child spends much time daily, and a child will learn various things in the life.
CO: Home and regions, children are routinely spend place a lot of time, you will learn a variety of things in life.
CR: Home and regions, children are routinely spend place a lot of time, children will learn a variety of things in life.
HT: Homes and communities are places where children spend a lot of time every day, and where they learn many things about life.

In Japanese writing, subjects tend to be omitted. Humans normally have no problems inferring the subjects from the context. In this case, for example, ‘children’ (子ども) or ‘they’ (彼ら) can be inferred in the latter clause. MT systems, however, often have difficulties in dealing with null-subject expressions. This is evidenced in AO, BO and CO above: system A did not insert a subject, but this caused a disagreement between the subject ‘a child’ and the verb ‘study’; system B adopted a passive construction ‘will be learned’; system C wrongly inserted ‘you’ as a subject. We can see that inserting ‘子ども’ as a subject in the original Japanese sentence enhanced the performance of the three systems (see AR, BR and CR).

MT-dependent CL rules
As Table 5 clearly demonstrates, it is also important to note that the effectiveness of each CL rule is variable. For example, rule 11 had a positive effect on the output of systems C and D. We look at it in more detail below.

Rule 11: avoid using attributive use of Shika-Nai (しか-ない)
AO/AR: Although it is a plant of a greenhouse, please look at this flower that makes a flower bloom only at this time once.
BO/BR: It’s a plant in a greenhouse, but please see this flower which makes a flower bloom only at this time once by all means.
CO: Although it is greenhouse plants, please come visit once this flower only at this time does not bloom.
CR: Although it is greenhouse plants, please come visit once the flowers bloom this time only to flower.
DO: Is a plant of the greenhouse, you take the time to peruse this flower only during this period not bloom.
DR: Is a plant of the greenhouse, you take the time to peruse the flowers that bloom only in this time of the year.
HT: Among the greenhouse plants, please be sure to take a look at this flower, which only blooms during this time of year.

In this case, we rewrote the attributive expression Shika-Nai (しか-ない) into another attributive particle Dake (だけ). For the RBMT systems A and B, both attributive patterns were linguistically processed in the same manner, using the adverb ‘only’, and this rule shows no improvement. On the other hand, for the SMT systems C and D, Shika-Nai (しか-ない) is dealt with as a kind of negative construction, which leads to an unnecessary negation in the output. Thus, regulating it is effective in improving MT quality.

While this particular rule triggered differing reactions in the RBMT versus SMT systems, we could not discern a regular correlation between system architectures and the effectiveness of CL rules. Instead, the results showed the idiosyncrasy of each system.
It should also be added that, in our experimental design to examine the effectiveness of each rule, we rewrote only that part of the sentence which was directly related to the applicable rule. This caused two major issues in our results. First, applying a single CL rule did not necessarily address the quality of the entire sentence, and thus did not contribute to as great an increase in [MT–Useful] as we had expected. In particular, there are cases where rewriting the source text did not change the MT outputs at all, as shown in the examples of rule 11 above. Even worse, rule 32 produced no improvement in any of the four MT systems. Second, in some cases a CL rule successfully brought a local improvement in the quality of the translation, yet other critical mistranslations (which, we observed, often stemmed from technical terms and proper nouns) overrode its positive effect and thus led to an overall low grade. This suggests that piece-by-piece modifications are not always enough to improve MT output quality.

**Optimal rule set for each system**

Table 6 shows how much improvement we could see if we select effective CL rules for each MT system. We preliminarily selected those rules which produced an increase in [MT–Useful] according to Table 5, i.e., 18 rules for system A, 19 for B, 19 for C and 16 for D, and summarised the results as in Table 4.

For all systems, [MT–Useful] category increases by more than 15 % with no or little increase in the [MT–Inaccurate] category. This result clearly indicates the necessity of tailoring the selection of rules to a particular MT system. In addition, further improvement can be expected if all applicable optimal rules are applied to a given sentence.

<table>
<thead>
<tr>
<th>Label</th>
<th>MT–Useful</th>
<th>MT–Inaccurate</th>
<th>MT–Unintelligible</th>
<th>HT–Unintelligible</th>
</tr>
</thead>
<tbody>
<tr>
<td>AO</td>
<td>20.6%</td>
<td>3.9%</td>
<td>68.9%</td>
<td>6.6%</td>
</tr>
<tr>
<td>AR</td>
<td>37.3%</td>
<td>3.9%</td>
<td>55.7%</td>
<td>3.1%</td>
</tr>
<tr>
<td>BO</td>
<td>20.6%</td>
<td>4.4%</td>
<td>70.6%</td>
<td>4.4%</td>
</tr>
<tr>
<td>BR</td>
<td>38.2%</td>
<td>5.5%</td>
<td>54.4%</td>
<td>3.9%</td>
</tr>
<tr>
<td>CO</td>
<td>21.1%</td>
<td>4.8%</td>
<td>70.2%</td>
<td>3.9%</td>
</tr>
<tr>
<td>CR</td>
<td>36.4%</td>
<td>5.3%</td>
<td>53.1%</td>
<td>5.3%</td>
</tr>
<tr>
<td>DO</td>
<td>17.7%</td>
<td>7.8%</td>
<td>69.3%</td>
<td>5.2%</td>
</tr>
<tr>
<td>DR</td>
<td>34.4%</td>
<td>6.8%</td>
<td>53.1%</td>
<td>5.7%</td>
</tr>
</tbody>
</table>

Table 6: Overall results of MT quality (after optimal rules were selected)

### 5.2 Japanese readability

The judgements of readability can be separated into two levels: acceptable and unacceptable. We defined the first two options of the question – *easy to read* and *fairly easy to read* – as acceptable and the other two options – *fairly difficult to read* and *difficult to read* – as unacceptable, on the assumption that the gap between the former and the latter is significant from the point of reading ease by humans.

Columns JO and JR in Table 7 show the percentages of judgements categorised as acceptable, and column JR-JO shows the improvement or deterioration. The higher the score in JR-JO, the greater the reading ease of JR compared to JO. Given our requirement that the CL should not degrade source text readability, figures greater than or equal to 0.0 (%) in JR-JO are highlighted in bold.

As a whole, 23 out of 38 the CL rules improved or at least retained the quality of the source text. In particular, rules 2, 18 and 26 were effective with more than 40% improvement in readability. According to rule 18 (avoid using redundant word), for instance, we deleted redundant expressions such as ‘ものとする’ and ‘こととする’ After this rewrite human evaluators judged JR to be more readable than JO. These kinds of periphrastic expressions are commonly used in
Table 7: Improvement in Japanese readability

<table>
<thead>
<tr>
<th>No</th>
<th>JO</th>
<th>JR</th>
<th>JR-JO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>91.7</td>
<td>75.0</td>
<td>-16.7</td>
</tr>
<tr>
<td>2</td>
<td>50.0</td>
<td>91.7</td>
<td>41.7</td>
</tr>
<tr>
<td>3</td>
<td>58.3</td>
<td>91.7</td>
<td>33.3</td>
</tr>
<tr>
<td>4</td>
<td>83.3</td>
<td>58.3</td>
<td>-25.0</td>
</tr>
<tr>
<td>5</td>
<td>58.3</td>
<td>91.7</td>
<td>33.3</td>
</tr>
<tr>
<td>6</td>
<td>91.7</td>
<td>50.0</td>
<td>-41.7</td>
</tr>
<tr>
<td>7</td>
<td>83.3</td>
<td>100.0</td>
<td>16.7</td>
</tr>
<tr>
<td>8</td>
<td>75.0</td>
<td>75.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>75.0</td>
<td>58.3</td>
<td>-16.7</td>
</tr>
<tr>
<td>10</td>
<td>58.3</td>
<td>91.7</td>
<td>33.3</td>
</tr>
<tr>
<td>11</td>
<td>83.3</td>
<td>75.0</td>
<td>-8.3</td>
</tr>
<tr>
<td>12</td>
<td>83.3</td>
<td>91.7</td>
<td>8.3</td>
</tr>
<tr>
<td>13</td>
<td>91.7</td>
<td>75.0</td>
<td>-16.7</td>
</tr>
<tr>
<td>14</td>
<td>75.0</td>
<td>83.3</td>
<td>8.3</td>
</tr>
<tr>
<td>15</td>
<td>75.0</td>
<td>83.3</td>
<td>8.3</td>
</tr>
<tr>
<td>16</td>
<td>66.7</td>
<td>66.7</td>
<td>0.0</td>
</tr>
<tr>
<td>17</td>
<td>66.7</td>
<td>100.0</td>
<td>33.3</td>
</tr>
<tr>
<td>18</td>
<td>58.3</td>
<td>100.0</td>
<td>41.7</td>
</tr>
<tr>
<td>19</td>
<td>83.3</td>
<td>75.0</td>
<td>-8.3</td>
</tr>
<tr>
<td>20</td>
<td>75.0</td>
<td>50.0</td>
<td>-50.0</td>
</tr>
<tr>
<td>21</td>
<td>100.0</td>
<td>66.7</td>
<td>-33.3</td>
</tr>
<tr>
<td>22</td>
<td>91.7</td>
<td>75.0</td>
<td>-16.7</td>
</tr>
<tr>
<td>23</td>
<td>66.7</td>
<td>91.7</td>
<td>25.0</td>
</tr>
<tr>
<td>24</td>
<td>100.0</td>
<td>66.7</td>
<td>-33.3</td>
</tr>
<tr>
<td>25</td>
<td>50.0</td>
<td>91.7</td>
<td>41.7</td>
</tr>
<tr>
<td>26</td>
<td>66.7</td>
<td>91.7</td>
<td>25.0</td>
</tr>
<tr>
<td>27</td>
<td>83.3</td>
<td>91.7</td>
<td>8.3</td>
</tr>
<tr>
<td>28</td>
<td>66.7</td>
<td>66.7</td>
<td>0.0</td>
</tr>
<tr>
<td>29</td>
<td>66.7</td>
<td>66.7</td>
<td>0.0</td>
</tr>
<tr>
<td>30</td>
<td>91.7</td>
<td>83.3</td>
<td>-8.3</td>
</tr>
<tr>
<td>31</td>
<td>58.3</td>
<td>58.3</td>
<td>0.0</td>
</tr>
<tr>
<td>32</td>
<td>100.0</td>
<td>66.7</td>
<td>-33.3</td>
</tr>
<tr>
<td>33</td>
<td>75.0</td>
<td>100.0</td>
<td>25.0</td>
</tr>
<tr>
<td>34</td>
<td>83.3</td>
<td>100.0</td>
<td>16.7</td>
</tr>
<tr>
<td>35</td>
<td>83.3</td>
<td>100.0</td>
<td>16.7</td>
</tr>
<tr>
<td>36</td>
<td>50.0</td>
<td>91.7</td>
<td>41.7</td>
</tr>
<tr>
<td>37</td>
<td>66.7</td>
<td>66.7</td>
<td>0.0</td>
</tr>
<tr>
<td>38</td>
<td>100.0</td>
<td>41.7</td>
<td>-58.3</td>
</tr>
</tbody>
</table>

municipal documents in Japan.

In contrast, rules 6, 21 and 38 resulted in more than 40% degradation in readability. Avoid using suffix (rule 21) and wave dash ‘ʙ’ (rule 38) introduce redundancy by replacing the feature with a longer sequence of words, such as replacing ‘午後 1 時～4 時’ (‘1:00–4:00 PM’) with ‘午後 1 時から 4 時まで’ (‘from 1:00 to 4:00 PM’).

More detailed analysis revealed there are some rules for which the evaluations of their effectiveness differed depending on the more specific features of sentences. For example, rewriting ‘記念品代相当分’ (‘an appropriate amount of money toward the commemorative item’) as ‘記念品代に相当する分’ (‘an appropriate amount of money corresponding to the commemorative item’) according to rule 19 (avoid using compound noun) improves readability, while rewriting ‘市民提供資料’ (‘materials provided by residents’) as ‘市民が提供した資料’ (‘materials that residents provided’) based on the same rule degrades readability.

5.3 Compatibility of machine translatability and Japanese readability

Comparing the results of the MT quality (Table 5) and Japanese readability (Table 7), we now discuss the compatibility of the two requirements. Focusing on the generally applicable rules, we see that rules 2, 8, 10, 12, 25, 28 and 37 improved or retained Japanese readability. In particular, rule 2 and 10 greatly improved both machine translatability and Japanese readability. There are, however, some rules which are effective for MT quality in general but have an adverse effect on human readability, such as rules 4, 13, 20 and 35. Rule 20, for instance, produced a better MT output for systems A, B and C, but degraded the readability of the source text. We give an example below.

Rule 20: avoid omission

**JO:** 月・水・金曜日 の午前 9 時から午後 4 時まで開設しており、3 月末まで開設して います。

**JR:** 月曜日・水曜日・金曜日 の午前 9 時から午後 4 時まで開設しており、3 月末まで開設しています。
BO: It’s established from a month and 9:00am of water and Friday to 4:00pm and it’s established until the end of March.

BR: It’s established from 9:00am of Monday, Wednesday and Friday to 4:00pm and it’s established until the end of March.

HT: It will be open from 9:00 AM to 4:00 PM on Mondays, Wednesdays, and Fridays until the end of March.

System B failed to recognise ‘月・水・金曜日’ as an elliptic expression and literally translated ‘月’ into ‘month’, ‘水’ into ‘water’, and ‘金曜日’ into ‘Friday’ (BO). Restoring an omitted element ‘曜日’ according to rule 20 helped the system deal correctly with this expression (BR). In contrast, human evaluators preferred JO to JR in terms of readability. This is no doubt because too much complementation made the sentence longer and hindered reading. The point here is that we need to find a way to meet both requirements of the MT quality and human readability. In the case of rule 20, for instance, we should keep JO for human readers, but employ pre-processing to produce JR for MT purposes.

6 Conclusion and future work

We have proposed an empirical protocol for formulating CL rules with a view to improving the machine translatability of Japanese municipal documents. Focusing on Japanese to English translation and using 100 sentences from municipal websites, we derived 38 CL rules different from the 22 rules that had previously been formulated on the basis of collective wisdom about technical writing.

We assessed the efficacy of the rules on municipal documents, with respect to both MT quality and source text readability. We identified a total of 11 rules which are effective for at least three MT systems. Since previous studies could identify few ‘general’ rules (Hartley et al., 2012; Tatsumi et al., 2013), this result encourages us to pursue our CL formulation protocol. In addition, we consider the protocol described in Section 3.1 to be generalisable to other language pairs and text domains, even though the CL rules in this study are formulated particularly for Japanese-to-English translation of municipal documents.

Interestingly, the effectiveness of CL rules was not shown to align with architectural differences between RBMT and SMT. This implies that we need to tune CL rule sets at the level of particular MT systems rather than at the level of MT types. In addition, a preliminary selection of optimal rules for each system achieved a greater than 15% increase in the [MT–Useful] category.

On the other hand, the results of the Japanese readability assessment showed that about two thirds of the CL rules improved or at least maintained source text readability. To achieve both machine translatability and human readability, it is important to serve different texts to humans and machines. We identified that degradations in readability for humans often correlate with redundancy generated by the rules. Thus an effective solution would be, for instance, to unpack the elliptic expressions and insert linguistic elements such as subjects only for MT. Moreover, this pre-processing for MT can be automated to some extent by employing existing pre-editing methods (e.g., Shirai et al., 1998), which can reduce the cost of implementing CL rules. In a future study, we will assess the total productivity of controlled authoring and translation in combination with post-editing.

Acknowledgement

This work was supported by the Research Grant Program of KDDI Foundation, Japan. The MT system J-SERVER Professional TransGateway V3 was offered by Kodensha Co.
References


Abstract

The lack of parallel data for many language pairs is an important challenge to statistical machine translation (SMT). One common solution is to pivot through a third language for which there exist parallel corpora with the source and target languages. Although pivoting is a robust technique, it introduces some low quality translations especially when a poor morphology language is used as the pivot between rich morphology languages. In this paper, we examine the use of synchronous morphology constraint features to improve the quality of phrase pivot SMT. We compare hand-crafted constraints to those learned from limited parallel data between source and target languages. The learned morphology constraints are based on projected alignments between the source and target phrases in the pivot phrase table. We show positive results on Hebrew-Arabic SMT (pivoting on English). We get 1.5 BLEU points over a phrase pivot baseline and 0.8 BLEU points over a system combination baseline with a direct model built from parallel data.

1 Introduction

One of the main challenges in statistical machine translation (SMT) is the scarcity of parallel data for many language pairs especially when the source and target languages are morphologically rich. A common SMT solution to the lack of parallel data is to pivot the translation through a third language (called pivot or bridge language) for which there exist abundant parallel corpora with the source and target languages. The literature covers many pivoting techniques. One of the best performing techniques, phrase pivoting (Utiyama and Isahara, 2007), builds an induced new phrase table between the source and target. One of the main issues of this technique is that the size of the newly created pivot phrase table is very large. Moreover, many of the produced phrase pairs are of low quality which affects the translation choices during decoding and the overall translation quality.

In this paper, we focus on improving phrase pivoting. We introduce morphology constraint scores which are added to the log linear space of features in order to determine the quality of the pivot phrase pairs. We compare two methods of generating the morphology constraints. One method is based on hand-crafted rules relying on the authors knowledge of the source and target languages; while in the other method, the morphology constraints are induced from
available parallel data between the source and target languages which we also use to build a
direct translation model. We then combine both the pivot and direct models to achieve better
coverage and overall translation quality. We show positive results on Hebrew-Arabic SMT.
We get 1.5 BLEU points over a phrase-pivot baseline and 0.8 BLEU points over a system
combination baseline with a direct model built from given parallel data.

Next, we briefly discuss some related work. In Section 3, we review the best performing
pivoting strategy and how we use it. In Section 4, we discuss the linguistic differences among
Hebrew, Arabic, and the pivot language, English. This is followed by our approach to using
morphology constraints in Section 5. We finally present our experimental results in Section 6
and a case study in Section 7.

2 Related Work

Many researchers have investigated the use of pivoting (or bridging) approaches to solve the data
scarcity issue (Utiiyama and Isahara, 2007; Wu and Wang, 2009; Khalilov et al., 2008; Bertoldi
et al., 2008; Habash and Hu, 2009). The main idea is to introduce a pivot language, for which
there exist large source-pivot and pivot-target bilingual corpora. Pivoting has been explored for
closely related languages (Hajić et al., 2000) as well as unrelated languages (Koehn et al., 2009;
Habash and Hu, 2009). Many different pivot strategies have been presented in the literature. The
following three are the most common. The first strategy is the sentence translation technique in
which we first translate the source sentence to the pivot language, and then translate the pivot
language sentence to the target language (Khalilov et al., 2008). The second strategy is based
on phrase pivoting (Utiiyama and Isahara, 2007; Cohn and Lapata, 2007; Wu and Wang, 2009).
In phrase pivoting, a new source-target phrase table (translation model) is induced from source-
pivot and pivot-target phrase tables. Lexical weights and translation probabilities are computed
from the two translation models. The third strategy is to create a synthetic source-target corpus
by translating the pivot side of source-pivot corpus to the target language using an existing
pivot-target model (Bertoldi et al., 2008). In this paper, we use the phrase pivoting approach,
which has been shown to be the best with comparable settings (Utiiyama and Isahara, 2007).

There has been recent efforts in improving phrase pivoting. One effort focused on improv-
ing alignment symmetrization targeting pivot phrase systems (El Kholy and Habash, 2014). In
another recent effort, Multi-Synchronous Context-free Grammar (MSCFG) is leveraged to
triangulate source-pivot and pivot-target synchronous Context-free Grammar (SCFG) rule ta-
bles into a source-target-pivot MSCFG rule table that helps in remembering the pivot during
decoding. Also, pivot LMs are used to assess the naturalness of the derivation (Miura et al.,
2015).

In our own previous work, we demonstrated quality improvement using connectivity
strength features between the source and target phrase pairs in the pivot phrase table (El Kholy
et al., 2013). These features provide quality scores based on the number of alignment links
between words in the source phrase to words of the target phrase. In this work, we extend
on the connectivity scores with morphological constraints through which we provide quality
scores based on the morphological compatibility between the connected/aligned source and tar-
get words.

Since both Hebrew and Arabic are morphologically rich, we should mention that there
has been a lot of work on translation to and from morphologically rich languages (Yeniterzi and
Oflazer, 2010; Elming and Habash, 2009; El Kholy and Habash, 2010; Habash and Sadat, 2006;
Kathol and Zheng, 2008). Most of these efforts are focused on syntactic and morphological
processing to improve the quality of translation.

Until recently, there has not been much parallel Hebrew-English and Hebrew-Arabic data
(Tsvetkov and Wintner, 2010), and consequently little work on Hebrew-English and Hebrew-
<table>
<thead>
<tr>
<th>Translation Model</th>
<th>Training Corpora Size</th>
<th>Phrase Table # Phrase Pairs</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hebrew-English</td>
<td>≈1M words</td>
<td>3,002,887</td>
<td>327MB</td>
</tr>
<tr>
<td>English-Arabic</td>
<td>≈60M words</td>
<td>111,702,225</td>
<td>14GB</td>
</tr>
<tr>
<td>Pivot_Hebrew-Arabic</td>
<td>N/A</td>
<td>&gt; 30 Billion</td>
<td>≈2.5TB</td>
</tr>
</tbody>
</table>

Table 1: Translation Models Phrase Table comparison in terms of number of lines and sizes.

Arabic SMT. Lavie et al. (2004) built a transfer-based translation system for Hebrew-English and so did Shilon et al. (2012) for translation between Hebrew and Arabic. Our previous work discussed above (El Kholy et al., 2013) was demonstrated on Hebrew-Arabic with English pivoting.

3 Phrase Pivoting

In this section, we review the phrase pivoting strategy in detail as we describe how we built our baseline for Arabic-Hebrew via pivoting on English. We also discuss how we overcome the large expansion of source-to-target phrase pairs in the process of creating a pivot phrase table. In phrase pivoting (which is sometimes called triangulation or phrase table multiplication), we train a Hebrew-Arabic and an English-Arabic translation models, such as those used in the sentence pivoting technique. Based on these two models, we induce a new Hebrew-Arabic translation model. Since our models are based on a Moses phrase-based SMT system (Koehn et al., 2007), we use the standard set of phrase-based translation probability distributions. We follow Utiyama and Isahara (2007) in computing the pivot phrase pair probabilities. The following are the set of equations used to compute the lexical probabilities \( p_w \) and the phrase translation probabilities \( \phi \):

\[
\phi(h|a) = \sum_e \phi(h|e)\phi(e|a) \\
\phi(a|h) = \sum_e \phi(a|e)\phi(e|h) \\
p_w(h|a) = \sum_e p_w(h|e)p_w(e|a) \\
p_w(a|h) = \sum_e p_w(a|e)p_w(e|h)
\]

Above, \( h \) is the Hebrew source phrase; \( e \) is the English pivot phrase that is common in both Hebrew-English translation model and English-Arabic translation model; and \( a \) is the Arabic target phrase. We also build a Hebrew-Arabic reordering table using the same technique but we compute the reordering probabilities in a similar manner to Henriquez et al. (2010).

Filtering for Phrase Pivoting As discussed earlier, the induced Hebrew-Arabic phrase and reordering tables are very large. Table 1 shows the amount of parallel corpora used to train the Hebrew-English and the English-Arabic and the equivalent phrase table sizes compared to the induced Hebrew-Arabic phrase table.\(^2\) We follow the work of El Kholy et al. (2013) and filter the phrase pairs used in pivoting based on log-linear scores. The main idea of the filtering process is to select the top \([n]\) English candidate phrases for each Hebrew phrase from the Hebrew-English phrase table and similarly select the top \([n]\) Arabic target phrases for each English phrase from the English-Arabic phrase table and then perform the pivoting process described earlier to create a pivoted Hebrew-Arabic phrase table. To select the top candidates, we

\(^1\)Four different phrase translation scores are computed in Moses’ phrase tables: two lexical weighting scores and two phrase translation probabilities.

\(^2\)The size of the induced phrase table size is computed but not created.
first rank all the candidates based on the log linear scores computed from the phrase translation probabilities and lexical weights multiplied by the optimized decoding weights then we pick the top \([n]\) pairs. In our experiments, we pick the top 1000 pairs for pivoting.

4 Linguistic Comparison

In this section we present the challenges of preprocessing Arabic, Hebrew, and English, and how we address them. Both Arabic and Hebrew are morphologically complex languages. One aspect of Arabic’s complexity is its various attachable clitics and numerous morphological features (Habash, 2010). Clitics include conjunction proclitics, e.g., +ٍّ w\(^+\) ‘and’, prepositional proclitics, e.g., +ٍّ l\(+\) ‘to/for’, the definite article +ٍّ Al\(+\) ‘the’, and the class of pronominal enclitics, e.g., ُهح+ +hm ‘their/them’. All of these clitics are separate words in English. Beyond the clitics, Arabic words inflect for person, gender, number, aspect, mood, voice, state and case. Additionally, Arabic orthography uses optional diacritics for short vowels and consonant doubling. This, together with Arabic’s morphological richness, leads to a high degree of ambiguity: about 12 analyses per word, typically corresponding to two lemmas on average (Habash, 2010). We follow El Kholy and Habash (2010) and use the PATB tokenization scheme (Maamouri et al., 2004) in our experiments. The PATB scheme separates all clitics except for the determiner clitic Al+(DET). We use MADA v3.1 (Habash and Rambow, 2005; Habash et al., 2009) to tokenize the Arabic text. We only evaluate on detokenized and orthographically correct (enriched) output following the work of El Kholy and Habash (2010).

Similar to Arabic, Hebrew poses computational processing challenges typical of Semitic languages (Itai and Wintner, 2008; Shilon et al., 2012). Hebrew orthography also uses optional diacritics and its morphology inflects for gender, number, person, state, tense and definiteness. Furthermore, Similar to Arabic, Hebrew has a set of attachable clitics, e.g., conjunctions (such as +ٍّ w\(^+\) ‘and’), prepositions (such as +ٍّ b\(+\) ‘in’), the definite article (+ٍّ h\(+\) ‘the’), or pronouns (such as ُوٍّ h\(+\) ‘their’). These issues contribute to a high degree of ambiguity that is a challenge to translation from Hebrew to English or to any other language. We follow Singh and Habash (2012)’s best preprocessing setup which utilized a Hebrew tagger (Adler, 2007) and produced a tokenization scheme that separated all clitics.

English, our pivot language, is quite different from both Arabic and Hebrew. English is poor in morphology and barely inflects for number and tense, and for person in a limited context. English preprocessing simply includes down-casing, separating punctuation and splitting off “’s”.

5 Approach

One of the main challenges in phrase pivoting is the very large size of the induced phrase table. It becomes even more challenging if either the source or target language is morphologically rich. The number of translation candidates (fanout) increases due to ambiguity and richness which in return increases the number of combinations between source and target phrases. Since the only criteria of matching between the source and target phrase is through a pivot phrase, many of the induced phrase pairs are of low quality. These phrase pairs unnecessarily increase the search space and hurt the overall quality of translation. A basic solution to the combinatorial expansion is to filter the phrase pairs used in pivoting based on log-linear scores as discussed in Section 3, however, this doesn’t solve the low quality problem.

\(^3\)Arabic transliteration throughout the paper is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

\(^4\)The following Hebrew 1-to-1 transliteration is used (in Hebrew alphabetical order): abgdhwzxlklmnspcqršt. All examples are undiacritized and final forms are not distinguished from non-final forms.
Similar to factored translation models (Koehn and Hoang, 2007) where linguistic (morphology) features are augmented to the translation model to improve the translation quality, our approach to address the quality problem is based on constructing a list of synchronous morphology constraints between the source and target languages. These constraints are used to generate scores to determine the quality of pivot phrase pairs. However, unlike factored models, we do not use the morphology in generation and the morphology information comes completely from external resources. In addition, since we work in the pivoting space, we only apply the morphology constraints to the connected words between the source and target languages through the pivot language. This guarantees a fundamental level of semantic equivalence before applying the morphology constraints especially if there is distortion between source and target phrases.

We build on our approach in El Kholy et al. (2013) where we introduced connectivity strength features between the source and target phrase pairs in the pivot phrase table. These features provide quality scores based on the number of alignment links between words in the source phrase and words in the target phrase. The alignment links are generated by projecting the alignments of the source-pivot phrase pairs and the pivot-target phrase pairs used in pivoting. We use the same concept but instead of using the lexical mapping between source and target words, we compute quality scores based on the morphological compatibility between the connected source and target words.

To choose which morphological features to work with, we performed an automatic error analysis on the output of the phrase-pivot baseline system. We did the analysis using AMEANA (El Kholy and Habash, 2011), an open-source error analysis tool for natural language processing tasks targeting morphologically rich languages. We found that the most problematic morphological features in the Arabic output are gender (GEN), number (NUM) and determiner (DET). We focus on those features in addition to (POS) in our experiments.

Next, we present our approach to generating the morphology constraint features using hand-crafted rules and compare this approach with inducing these constraints from Hebrew-Arabic parallel data.

5.1 Rule-based Morphology Constraints

Our rule-based morphology constraint features are basically a list of hand-crafted mappings of the different morphological features between Hebrew and Arabic. Since both languages are morphologically rich as explained in Section 4, it is straightforward to produce these mappings for GEN, NUM and DET. Note, however, that we also account for ambiguous cases; e.g., feminine gender in Arabic can map to words with ambiguous gender in Hebrew. We additionally use different POS tag sets for Arabic (47 tags) and Hebrew (25 tags) and in many cases one Hebrew tag can map to more than one Arabic tag; for example, three Arabic noun tags abbrev, noun and noun_prop map to two Hebrew tags feminine, masculine noun.\(^5\) Table 2 shows a sample of the morphological mappings between Arabic and Hebrew.

After building the morphological features mappings, we use them to judge the quality of a given phrase pair in the phrase pivot model. We add two scores \(W_s\) and \(W_t\) to the log linear space. Given a source-target phrase pair \(s, t\) and a word projected alignment \(a\) between the source word positions \(i = 1, \ldots, n\) and the target word positions \(j = 1, \ldots, m\), \(W_s\) and \(W_t\) are defined in equations 1 and 2. \(F\) is the set of morphological features (we focus on GEN, NUM, DET and POS). \(M_f\) is the hand-crafted rules mapping between Arabic and Hebrew feature values of feature \(f \in F\). In case of ambiguity for a given feature; for example, a word’s gender being masculine or feminine, we use the maximum likelihood value of this feature given the word. \(MLE_f(i)\) is the maximum likelihood feature value of feature \(f\) for the source word at

\(^5\)Please refer to (Habash et al., 2009) for a complete set of Arabic POS tag set and (Adler, 2007) for Hebrew POS tag set.
Table 2: Rule-based mapping between Arabic and Hebrew morphological features. Each feature value in Arabic can map to more than one feature value in Hebrew.

<table>
<thead>
<tr>
<th>Features</th>
<th>Arabic</th>
<th>Hebrew</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEN</td>
<td>Feminine</td>
<td>Feminine / Both</td>
</tr>
<tr>
<td></td>
<td>Masculine</td>
<td>Masculine / Both</td>
</tr>
<tr>
<td>NUM</td>
<td>Singular</td>
<td>Singular / Singular-Plural</td>
</tr>
<tr>
<td></td>
<td>Dual</td>
<td>Dual / Dual-Plural</td>
</tr>
<tr>
<td></td>
<td>Plural</td>
<td>Plural / Dual-Plural / Singular-Plural</td>
</tr>
<tr>
<td>DET</td>
<td>No Determiner</td>
<td>No Determiner</td>
</tr>
<tr>
<td></td>
<td>Determiner</td>
<td>Determiner</td>
</tr>
</tbody>
</table>

| position \(i\), and \(MLE_f(j)\) is the maximum likelihood feature value of feature \(f\) for the target word at position \(j\). The maximum likelihood feature values for Hebrew were computed from the Hebrew side of the training data. As for Arabic, the maximum likelihood feature values were computed from the Arabic side of the training data in addition to Arabic Gigaword corpus, which was used in creating the language model (more details in Section 6.1).

\[
W_s = \frac{1}{|F|} \sum_{f \in F} \sum_{(s,j) \in a} \frac{1}{n} [ (MLE_f(i), MLE_f(j)) \in M_f ]
\]

(1)

\[
W_t = \frac{1}{|F|} \sum_{f \in F} \sum_{(s,j) \in a} \frac{1}{m} [ (MLE_f(i), MLE_f(j)) \in M_f ]
\]

(2)

5.2 Induced Morphology Constraints

In this section, we explain our approach in generating morphology constraint features from a given parallel data between source and target languages. Unlike the rule-based approach we build a translation model between the source and target morphological features and we use the morphology translation probabilities as metric to judge a given phrase pair in the pivot phrase table. For the automatically induced constraints, we jointly model mapping between conjunctions of features attached to aligned words rather than tallying each feature match independently. Writing good manual rules for such feature conjunction mappings would be more difficult. Table 3 shows some examples of mapping (GEN), number (NUM) and determiner (DET) in Hebrew to their equivalent in Arabic and their respective bi-directional scores.

| Hebrew (H) | Arabic (A) | \(P_{FC}(A|H)\) | \(P_{FC}(H|A)\) |
|------------|------------|-----------------|-----------------|
| [Fem+Dual+Det] | [Fem+Dual] | 0.0006          | 0.0833          |
| [Fem+Dual+Det] | [Fem+Dual+Det] | 0.0148          | 0.3333          |
| [Fem+Dual+Det] | [Fem+Singl] [Fem+Dual] | 0.0052          | 0.0833          |
| [Fem+Dual+Det] | [Masc+Dual+Det] | 0.0047          | 0.5000          |

Table 3: Examples of induced morphology constraints for (GEN), number (NUM) and determiner (DET) and their respective scores.

As in rule-based approach, we add two scores \(W_s\) and \(W_t\) to the log linear space which are defined in equations 3 and 4. \(P_{FC}\) is the conditional morphology probability of a given feature.
combination \((FC)\) value. Similar to rule-based morphology constraints, we resort to the maximum likelihood value of a feature combination when the values are ambiguous. \(MLE_{FC}(i)\) is the maximum likelihood feature combination \((FC)\) value for the source word at position \(i\) while \(MLE_{FC}(j)\) is the maximum likelihood feature combination \((FC)\) value for the target word at position \(j\).

\[
W_s = \frac{1}{n} \sum_{\forall (i,j) \in a} P_{FC}(MLE_{FC}(i)|MLE_{FC}(j))
\]

\[
W_t = \frac{1}{m} \sum_{\forall (i,j) \in a} P_{FC}(MLE_{FC}(j)|MLE_{FC}(i))
\]

5.3 Model Combinations

Since we use parallel data to induce the morphology constraints, it would make sense to measure the effect of combining (a) the pivot model with added morphology constraints, and (b) the direct model trained on the parallel data used to induce the morphology constraints. We perform the combination using Moses’ phrase table combination techniques. Translation options are collected from one table, and additional options are collected from the other tables. If the same translation option (in terms of identical input phrase and output phrase) is found in multiple tables, separate translation options are created for each occurrence, but with different scores (Koehn and Schroeder, 2007). We show results over a learning curve in Section 6.5.

6 Experiments

In this section, we present a set of experiments comparing the use of rule-based versus induced morphology constraint features in phrase-pivot SMT as well as model combination to improve Hebrew-Arabic pivot translation quality.

6.1 Experimental Setup

In our pivoting experiments, we build two SMT models; one model to translate from Hebrew to English, and another model to translate from English to Arabic. The English-Arabic parallel corpus is about \((\approx 60M \text{ words})\) and is available from LDC\(^6\) and GALE\(^7\) constrained data. The Hebrew-English corpus is about \((\approx 1M \text{ words})\) and is available from sentence-aligned corpus produced by Tsvetkov and Witnter (2010). For the direct Hebrew-Arabic SMT model, we use a TED parallel corpus of about \((\approx 2M \text{ words})\) (Cettolo et al., 2012).

Word alignment is done using GIZA++ (Och and Ney, 2003). For Arabic language modeling, we use 200M words from the Arabic Gigaword Corpus (Graff, 2007) together with the Arabic side of our training data. We use 5-grams for all language models (LMs) implemented using the SRILM toolkit (Stolcke, 2002).

All experiments are conducted using the Moses phrase-based SMT system (Koehn et al., 2007). We use MERT (Och, 2003) for decoding weight optimization. Weights are optimized using a tuning set of 517 sentences developed by Shilon et al. (2010).

We use a maximum phrase length of size 8 across all models. We report results on a Hebrew-Arabic development set (Dev) of 500 sentences with a single reference and an evaluation set (Test) of 300 sentences with three references developed by Shilon et al. (2010). We evaluate using BLEU-4 (Papineni et al., 2002).


\(^7\)Global Autonomous Language Exploitation, or GALE, was a DARPA-funded research project.
6.2 Baselines

We compare the performance of adding the connectivity strength features (+Conn) to the phrase pivoting SMT model (Phrase_Pivot) and building a direct SMT model using all parallel He-Ar corpus available. The results are presented in Table 4. Consistently with our previous effort (El Kholy et al., 2013), the performance of the phrase-pivot model improves with the connectivity strength features. While the direct system is better than the phrase pivot model in general, the combination of both models leads to a high performance gain of 1.7/4.4 BLEU points in Dev/Test over the best performers of both the direct and phrase-pivot models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct</td>
<td>9.7</td>
<td>20.4</td>
</tr>
<tr>
<td>Phrase_Pivot</td>
<td>8.3</td>
<td>19.8</td>
</tr>
<tr>
<td>Phrase_Pivot+Conn</td>
<td>9.1</td>
<td>20.1</td>
</tr>
<tr>
<td>Direct+Phrase_Pivot+Conn</td>
<td>11.4</td>
<td>24.5</td>
</tr>
</tbody>
</table>

Table 4: Comparing phrase pivoting SMT with connectivity strength features, direct SMT and the model combination. The results show that the best performer is the model combination in Dev and Test sets.

6.3 Rule-based Morphology Constraints

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single</td>
<td>Combined</td>
</tr>
<tr>
<td>Direct</td>
<td>9.7</td>
<td>n/a</td>
</tr>
<tr>
<td>Phrase_Pivot+Conn</td>
<td>9.1</td>
<td>11.4*</td>
</tr>
<tr>
<td>Phrase_Pivot+Conn+Morph_Rules</td>
<td>9.6</td>
<td>12.2*</td>
</tr>
<tr>
<td>Phrase_Pivot+Conn+Morph_Auto</td>
<td>9.6</td>
<td>12.4*</td>
</tr>
</tbody>
</table>

Table 5: Morphology constraints results. The “Single” columns show the results of a single model of either the direct model or the phrase pivoting models with additional morphological constraints features. The “Combined” show the results of system combination between the direct model and the different phrase pivoting models. In the first row, the “Combined” results are not applicable for the direct model. (*) marks a statistically significant result against both the direct and phrase-pivot baseline.

In this experiment, we show the performance of adding hand-crafted morphology constraints (+Morph_Rules) to determine the quality of a given phrase pair in the phrase-pivot translation model. The third row in Table 5 shows that although the rules are based on a one-to-one mapping between the different morphological features, the translation quality is improved over the baseline phrase-pivot model by 0.5/0.8 BLEU points in Dev/Test sets.

As expected, the system combination of the pivot model with the direct model improves the overall performance but the gain we get from the morphology constraints only appears in the Dev set with 0.8 BLEU points, and not much in the Test set.

6.4 Induced Morphology Constraints

In this experiment, we measure the effect of using induced morphology constraints (+Morph_Auto) on MT quality. The last row in Table 5 shows that the induced morphology constraints improve the results over the baseline phrase-pivot model by 0.5/1.5 BLEU points in
### Table 6: Learning curve results of 100% (2M words), 25% (500K words) and 6.25% (125K words) of the parallel Hebrew-Arabic corpus.

<table>
<thead>
<tr>
<th>Parallel Data Size</th>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Single</td>
<td>Combined</td>
</tr>
<tr>
<td>125K</td>
<td>Direct</td>
<td>2.7</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>Phrase_Pivot+Conn</td>
<td>9.1</td>
<td>10.4</td>
</tr>
<tr>
<td></td>
<td>Phrase_Pivot+Conn+Morph_Auto</td>
<td>9.2</td>
<td>10.6</td>
</tr>
<tr>
<td>500K</td>
<td>Direct</td>
<td>5.9</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>Phrase_Pivot+Conn</td>
<td>9.1</td>
<td>10.7</td>
</tr>
<tr>
<td></td>
<td>Phrase_Pivot+Conn+Morph_Auto</td>
<td>9.7</td>
<td>11.2</td>
</tr>
<tr>
<td>2M</td>
<td>Direct</td>
<td>9.7</td>
<td>n/a</td>
</tr>
<tr>
<td></td>
<td>Phrase_Pivot+Conn</td>
<td>9.1</td>
<td>11.4</td>
</tr>
<tr>
<td></td>
<td>Phrase_Pivot+Conn+Morph_Auto</td>
<td>9.6</td>
<td>12.4</td>
</tr>
</tbody>
</table>

Dev/Test sets and over the Rule-based morphology constraints by 0.7 BLEU points in the Test set.

Similar to the Rule-based constraints, the performance did not improve compared to the *direct model* in the Dev set; but, again, the Test set showed a great improvement of 1.5 and 1.2 BLEU points over the pivot and direct models, respectively. Also the system combination of the pivot model with the direct model improves the overall performance. The model using induced morphological features is the best performer with an increase in the performance gain by 1.0/0.8 BLEU points in Dev/Test sets. This shows that the benefit we get from the induced morphology constraints were not diluted when we do the model combination given the fact that the constraints were induced from the parallel data to start with.

It is important to note here that the induced morphology constraints outperformed the rule-based constraints across all settings. This shows that the complex morphology constraints extracted from the parallel data provide knowledge that can not be covered by simple linguistic rules. However, the simple rule-based approach comes in handy when there is no data between the source and target languages.

#### 6.5 Learning Curve

In this experiment, we examine the effect of using less data in inducing morphology constraints rules and the overall performance when we combine systems. Table 6 shows the results on a learning curve of 100% (2M words), 25% (500K words) and 6.25% (125K words) of the parallel Hebrew-Arabic corpus.

As expected, The system combination between the direct translation models and the phrase-pivot translation model leads to an improvement in the translation quality across the learning curve even when there is small amount of parallel corpora. Despite the weak performance (2.7 BLEU) of the direct system built on 6.25% of the parallel Hebrew-Arabic corpus, the system combination leads to 1.4 BLEU points gain.

An interesting observation from the results is that we always get a performance gain from the induced morphology constrains across all settings. This shows that the system combination helps in adding more lexical translation choices while the constraints help in a different dimension, which is selecting the best phrase pairs from the pivot system.
Table 7: Translation examples.

<table>
<thead>
<tr>
<th>Hebrew Source</th>
<th>the+middlemen and+the+traders refuse[m.p.] to+speak publicly about the+prices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic Reference</td>
<td>يرفض الوسطاء والتجار الحديث عن الأسعار</td>
</tr>
<tr>
<td>Phrase_Pivot+Conn</td>
<td>middlemen and+the+traders refused[m.p.] the+speaking publicly about the+prices</td>
</tr>
<tr>
<td>Direct</td>
<td>middlemen and+the+traders refused[m.p.] the+speaking publicly about the+prices</td>
</tr>
<tr>
<td>Phrase_Pivot+Conn+Morph_Auto</td>
<td>middlemen and+the+traders refused[m.p.] the+speaking publicly about the+prices</td>
</tr>
<tr>
<td>Direct+Phrase_Pivot+Conn+Morph_Auto</td>
<td>middlemen and+the+traders refused[m.p.] the+speaking publicly about the+prices</td>
</tr>
</tbody>
</table>

7 Case Study

In this section we consider an example from our Dev set that captures many of the patterns and themes in the evaluation. Table 7 shows a Hebrew source sentence and its Arabic reference. This is followed by the output from the pivot system, the direct system, the Phrase_Pivot+Conn+Morph_Auto system and the combined system.

Two particular aspects should be noted. First is the complementary lexical coverage of the direct and pivot systems. This is seen in how one of each covers half of the phrase middlemen and traders. The combined system captures both. Secondly, the gender, number and tense of the main verb prove challenging in many ways (and this is an issue for a majority of the sentences in the Dev set). The Hebrew verb in the present tense is masculine and plural; and naturally follows the subject. The Arabic reference verb appears at the beginning of the sentence, in which location it only agrees with the subject in gender (while number is singular). Arabic Verbs in SVO order agree in gender and number. All the MT systems we compare leave the verb after the subject. The direct, Phrase_Pivot+Conn+Morph_Auto, and combination systems get the number and gender correctly; however, the direct and combined system make the verb tense past. The Phrase_Pivot+Conn+Morph_Auto example highlights the value of morphology constraints; but the example points out that they sometimes are hard to evaluate automatically, since there are morphosyntactically allowable forms that do not match the translation references.

8 Conclusion and Future Work

In this paper, we presented the use of synchronous morphology constraint features based on hand-crafted rules compared to rules induced from parallel data to improve the quality of phrase-pivot based SMT. We show that the two approaches lead to an improvement in the translation quality. The induced morphology constraints approach is a better performer, however, it relies on the fact there is a parallel corpus between source and target languages. We show positive results on Hebrew-Arabic SMT. We get 1.5 BLEU points over phrase-pivot baseline and 0.8 BLEU points over system combination baseline with direct model built from given parallel data.

In the future, we plan to work on reranking experiments as a post-translation step based on morphosyntactic information between source and target languages. We also plan to work on word reordering between morphologically rich language to maintain the relationship between the word order and the morphosyntactic agreement in the context of phrase pivoting.
Acknowledgments

The work presented in this paper was possible thanks to a generous Google Research Award. We would like to thank Reshef Shilon and Shuly Winter for helpful discussions and support with processing Hebrew. We also thank the anonymous reviewers for their insightful comments.

References


Using Joint Models for Domain Adaptation in Statistical Machine Translation

Nadir Durrani  Hassan Sajjad  Shafiq Joty  Ahmed Abdelali  Stephan Vogel  
Qatar Computing Research Institute – Hamad Bin Khalifa University  
{ndurrani,hsajjad,sjoty,aabdelali,svogel}@qf.org.qa

Abstract

Joint models have recently shown to improve the state-of-the-art in machine translation (MT). We apply EM-based mixture modeling and data selection techniques using two joint models, namely the Operation Sequence Model or OSM — an ngram-based translation and reordering model, and the Neural Network Joint Model or NNJM — a continuous space translation model, to carry out domain adaptation for MT. The diversity of the two models, OSM with inherit reordering information and NNJM with continuous space modeling makes them interesting to be explored for this task. Our contribution in this paper is fusing the existing known techniques (linear interpolation, cross-entropy) with the state-of-the-art MT models (OSM, NNJM). On a standard task of translating German-to-English and Arabic-to-English IWSLT TED talks, we observed statistically significant improvements of up to +0.9 BLEU points.

1 Introduction

Parallel data required to train Statistical Machine Translation (SMT) systems is often inadequate, and is typically collected opportunistically from wherever it is available. The conventional wisdom is that more data improves the translation quality. Additional data however, may not be best suited for tasks such as translating TED talks (Cettolo et al., 2014) or patents (Fujii et al., 2010) or educational content (Abdelali et al., 2014), and often come with the challenges of dealing with word-sense ambiguities and stylistic variance of other domains. When additional data, later referred as out-domain, is much larger than in-domain, the resultant distribution can get biased towards out-domain, yielding a sub-optimal system. Domain adaptation aims to preserve the identity of the in-domain data while using the best of the out-domain data. This is done by selecting a subset from the out-domain data, which is closer to the in-domain (Matsoukas et al., 2009; Moore and Lewis, 2010), or by re-weighting the probability distribution in favor of the in-domain data (Foster and Kuhn, 2007; Sennrich, 2012).

Bilingual sequence models (Maríño et al., 2006) have shown to be effective in improving the quality of machine translation and have achieved state-of-the-art performance recently (Le et al., 2012; Durrani et al., 2013; Devlin et al., 2014). Their ability to capture non-local dependencies makes them superior to the traditional phrase-based models, which do not consider contextual information across phrasal boundaries. Two such models that we explore in this paper are (i) the Operation Sequence Model or OSM (Durrani et al., 2011) — a markov translation model that integrates reordering, and (ii) the Neural Network Joint Model or NNJM (Devlin et al., 2014) — a continuous space model that learns neural network over augmented streams of source and target sequences. Both models are used as additional language model (LM) features inside the SMT decoder.
The diversity of the two models, i.e., OSM with embedded reordering information and NNJM with continuous space modeling, makes them interesting to be explored for domain adaptation. The LM-like nature of the two models provides motivations to apply methods such as perplexity optimization for model weighting and cross-entropy-based ranking for data selection. In this paper, we explore both avenues. Firstly, we train models (OSM and NNJM) from each domain separately and then interpolate them (i) linearly using Expectation-Maximization or EM-based weighting, (ii) using log-linear model inside the SMT pipeline. Secondly, we use cross-entropy difference (Moore and Lewis, 2010) between in- and out-domain models to perform data selection for domain adaptation.

The bilingual property of the OSM and NNJM models gives them an edge over traditional LM-based methods, which do not capture source and target domain relevance jointly. The embedded reordering information modeled in OSM helps it to preserve reordering characteristic of the in-domain data. Capturing reordering variation across domains have been shown to be beneficial also by Chen et al. (2013a). NNJM adds a different dimension to it by semantically generalizing the data using distributed representation of words (Bengio et al., 2003).

We evaluated our systems on a standard task of translating IWSLT TED talks for German-to-English (DE-EN) and Arabic-to-English (AR-EN) language pairs. Below is a summary of our main findings:

**Model Weighting:**
- Linearly interpolating OSM models through EM-based weighting gave average BLEU (Papineni et al., 2002) improvements of up to +0.6 for DE-EN and +0.9 for AR-EN.
- Log-linear variant performed better in the case of NNJM giving an average improvements of +0.4 BLEU points for DE-EN and +0.5 for AR-EN.
- Linear interpolation for NNJM models was slightly behind its log-linear variant.

**Data Selection:**
- OSM-based selection performed better for AR-EN task giving an average improvement of +0.7
- NNJM performed better at the DE-EN task giving an average improvement of +0.6 points.
- Both OSM- and NNJM-based selection gave slightly better results than Modified-Moore-Lewis (MML) selection (Axelrod et al., 2011).

The rest of the paper is organized as follows. Section 2 briefly describes the OSM and the NNJM models. Section 3 describes mixture model and data selection techniques that we apply using the OSM and the NNJM models to carry out adaptation. Section 4 presents the results. Section 5 discusses related work and Section 6 concludes the paper.

2 Joint Sequence Models
In this section, we revisit Operation Sequence and Neural Network Joint models briefly.

2.1 Operation Sequence Model
The Operation Sequence Model (OSM) is a bilingual model that couples translation and reordering by representing them as a sequence of operations. An operation either generates source
and/or target word(s) or performs reordering by inserting gaps and jumping forward and backward. A bilingual sentence pair \((T, S)\) and its word-alignment \(A\) is transformed deterministically to a heterogeneous sequence of translation and reordering operations \((o_1, o_2, \ldots, o_J)\). A Markov model is then learned over these sequences:

\[
P_{osm}(T, S) = P(o_1, \ldots, o_J) \approx \prod_{j=1}^{J} P(o_j | o_{j-n+1} \ldots o_{j-1})
\]

For example, the German-English sentence pair shown in Figure 1 can be converted into the following sequence of operations:

- Generate (Wir, We)
- Generate (haben, have)
- Insert Gap
- Generate (genommen, taken)
- Jump Back (1)
- Generate (sie, them)
- Generate (aus, out)
- Generate (ihrër, of their)
- Generate (urspr"unglichen, natural)
- Generate (Pyramids, pyramid)

Figure 1: Sample German-English Sentence with Alignments

The generation is carried out in the order of target (English in this case). Gaps and jumps are inserted on the source side. Unaligned source and target words are handled through Generate Source Only and Generate Target Only operations, respectively. Discontinuous source and target units are handled through other operations; see Durrani et al. (2011) for details about the operations and the algorithm to convert a word-aligned corpus into sequences of operations.

Mixing lexical generation and reordering, each (translation or reordering) decision conditions on \(n - 1\) previous (translation or reordering) decisions. This allows the model to learn very rich translation and reordering patterns. Moreover, the model is based on minimal translation units (MTUs) and considers source and target contextual information across phrasal boundaries, thus addressing phrasal independence assumption and spurious segmentation problems in traditional phrase-based MT.

2.2 Neural Network Joint Model

In recent years, there has been a great deal of effort dedicated to neural networks (NNs) and word embeddings with applications to MT and other areas in NLP (Bengio et al., 2003; Auli et al., 2013; Kalchbrenner and Blunsom, 2013; Gao et al., 2014; Schwenk, 2012; Collobert et al., 2011; Mikolov et al., 2013; Socher et al., 2013; Hinton et al., 2012). A bilingual Neural Network Joint model for MT was recently proposed by Devlin et al. (2014). It learns a feed-forward neural network from augmented streams of source and target sequences. For a bilingual sentence pair \((S, T)\), NNJM defines a conditional probability distribution:

\[
P(T|S) \approx \prod_{i=1}^{[T]} P(t_i|t_{i-1} \ldots t_{i-n+1}, s_i)
\]

where, \(s_i\) is an \(m\)-word source window for a target word \(t_i\) based on the one-to-one alignment between \(T\) and \(S\). Each input word in the context has a \(D\) dimensional (continuous-valued)
vector representation in the shared look-up table $L \in \mathbb{R}^{|V_i| \times D}$, where $V_i$ is the input vocabulary. The context of the sequence is represented by a concatenated vector $x_n \in \mathbb{R}^{(m+n-1)D}$, which is then passed through non-linear hidden layers to learn a high-level representation. The output layer is a softmax over the output vocabulary $V_o$:

$$P(y_n = k | x_n, \theta) = \frac{\exp (w_k^T \phi(x_n))}{\sum_{m=1}^{|V_o|} \exp (w_m^T \phi(x_n))}$$

where $\phi(x_n)$ defines the non-linear transformations of $x_n$, and $w_k$ are the weights from the outermost hidden layer to the output layer. By setting $m$ and $n$ to be sufficiently large, NNJM can capture long-range cross-lingual dependencies between words.

### 3 Domain Adaptation

The ability to learn rich lexical and reordering patterns by OSM, the generalization power of NNJM, and their strong empirical results in MT gives us a strong motivation to use them for the problem of domain adaptation. However, the OSM and NNJM models trained on a plain concatenation of in-domain data with large and diverse multi-domain data are suboptimal. When other domains are sufficiently larger and/or different than the in-domain, the probability distribution can skew away from the target domain resulting in poor performance. The goal in domain adaptation is to do restrict this drift while still using the best of the available data.

We analyze the operation corpus as generated by the corpus conversion algorithm of Durani et al. (2011) in OSM training. It provides useful insights on the amount of reordering, number of (source word) insertions and (target word) deletions that are carried out in the bilingual corpus. We use this information to motivate our study. Table 1 shows some statistics about the operations in several datasets. We report probabilities of Jumps (Jump Forward and Jump Back (*) operations), Gaps (Insert Gap operation), Insertions of source words (Generate Source Only (X) operation to handle unaligned source words) and Deletions of target words (Generate Target Only (Y) operation to handle unaligned target words) in each domain.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Jumps</th>
<th>Gaps</th>
<th>Deletions</th>
<th>Insertions</th>
</tr>
</thead>
<tbody>
<tr>
<td>German-to-English</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iwslt</td>
<td>0.17</td>
<td>0.09</td>
<td>0.06</td>
<td>0.04</td>
</tr>
<tr>
<td>news</td>
<td>0.21</td>
<td>0.13</td>
<td>0.05</td>
<td>0.07</td>
</tr>
<tr>
<td>europarl</td>
<td>0.22</td>
<td>0.14</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>common crawl</td>
<td>0.19</td>
<td>0.11</td>
<td>0.12</td>
<td>0.11</td>
</tr>
</tbody>
</table>

| Arabic-to-English |       |        |           |            |
| iwslt            | 0.17  | 0.09   | 0.07      | 0.05       |
| UN               | 0.21  | 0.12   | 0.07      | 0.08       |

Table 1: Probabilities of Jumps, Gaps, Insertion and Deletion operations in each domain.

The probabilities of Jumps and Gaps in the in-domain IWSLT data are lower than other domains in both German-to-English and Arabic-to-English language pairs. This indicates that lesser amount of reordering is required in the in-domain data. Because other domains are significantly larger than the in-domain data, the resulting distribution would get biased towards doing more reordering than desired. For example Insert Gap operation in Europarl and UN data is much probable than IWSLT (compare column Gaps in Table 1). Similarly the probability of insertions carried out in the in-domain data is less than the other domains. Therefore, the resulting models
would favor more insertions than preferred by the in-domain data. Table 1 does not show statistics on different vocabularies, but lexical variance between domains is obviously another cause of divergence from the in-domain data, which previous methods have also tackled. In this work, we additionally address the reordering variance across domains. These statistics, although, collected from the operation corpus on which the OSM model is trained, can be reflected on the NNJM training as well which uses same word-alignments to generate the stream of source and target n-grams.

In this paper we study two directions to perform domain adaptation in MT. We apply mixture modeling, a well-established model weighting technique, to re-weight the models in favor of the in-domain data. More specifically, we first train OSM and NNJM models on different domains and then use an EM-based interpolation to optimize the weights based on an in-domain tuning set. We also use the two models to rank sequences for data selection using cross entropy difference. In the next two subsections we discuss these in detail.

### 3.1 Model Weighting

We use both OSM and NNJM models as an additional language model feature inside the decoder. A domain-adapted version of the model, biased towards the in-domain data, can help assigning higher scores to the hypotheses that represent lexical choices and reordering patterns preferred by the in-domain data. We train OSM and NNJM models from each domain separately and learn the relative weights of the models using linear and log-linear interpolation methods. For linear interpolation, we compute weights by optimizing perplexity on in-domain tuning set using a standard EM-based algorithm as described below:

**Model Weighting by EM:** Let \( \theta_d \in \{ \theta_1, \ldots, \theta_D \} \) represent a model (e.g., OSM, NNJM) trained on domain \( d \), where \( D \) is the total number of domains. The probability of a sequence \( x_n \) can be written as a mixture of \( D \) probability densities, each coming from a different model:

\[
P(x_n|\theta, \lambda) = \sum_{d=1}^{D} P(x_n|z_n=d, \theta_d) \lambda_d
\]

where \( P(x_n|z_n=d, \theta_d) \) represents the probability of \( x_n \) assigned by model \( \theta_d \), and the mixture weights \( \lambda_d \) satisfy \( 0 \leq \lambda_d \leq 1 \) and \( \sum_{d=1}^{D} \lambda_d = 1 \). In our setting, \( \theta = \{ \theta_1, \ldots, \theta_D \} \) is known, and we can use EM to learn the mixture weights. The expected complete data log likelihood is given by:

\[
E[L(\lambda)] = \sum_{n=1}^{N} \sum_{d=1}^{D} r_{nd} \log [P(x_n|z_n=d, \theta_d)\lambda_d]
\]

where \( r_{nd} = P(z_n=d|x_n, \theta_d, \lambda_{d}^{t-1}) \) is the responsibility that domain \( d \) takes for data point \( n \) given the mixing weight in the previous step \( \lambda_{d}^{t-1} \). In the E-step, we compute \( r_{nd} \) and we update \( \lambda \) in the M-step. More specifically:

**E-step:** Compute \( r_{nd}^{t} = \frac{\lambda_{d}^{t-1} P(x_n|z_n=d, \theta_d)}{\sum_{d'=1}^{D} \lambda_{d'}^{t-1} P(x_n|z_n=d', \theta_{d'})} \)

**M-step:** Update \( \lambda_{d}^{t} = \frac{1}{N} \sum_{n=1}^{N} r_{nd}^{t} \)

Once we have learned the relative weights of the models based on the in-domain tuning data, we can linearly interpolate the models as:

\[1\]The tuning-set is required to be word-aligned and then converted into a sequence of operations (for OSM) and augmented streams of source and target strings (for NNJM) to compute model-wise perplexities.
An alternative way to combine the models is through log-linear interpolation by optimizing weights, directly on BLEU, along with other features inside of the SMT pipeline.

### 3.2 Data Selection

An alternative to model weighting is data selection, which attempts to filter out harmful data from the training corpus rather than down weighting it. Data selection could be useful in a scenario with memory constraints. However, a down-side of this approach is that it requires extensive amount of experimentation to find an optimal cut-off point.

In this paper, we select data using differences in cross entropy as proposed by Moore and Lewis (2010). More specifically, we first train a model (OSM or NNJM) on the in-domain corpus, and then train another model on the out-domain data of equal size. Then we score the out-domain data using:

\[
\text{score}(x) = H_D(x) - H_O(x)
\]

where \(x\) is a sequence of operations \((o_1, \ldots, o_n)\) in the case of OSM and an augmented stream of source and target sequences \((t_1, \ldots, t_n, s_i)\) in the case of NNJM. \(H_D\) is the cross-entropy between a model and the empirical n-gram distribution in the domain \(D\). We train a 5-gram OSM and a 14-gram NNJM with 5-grams on target-side and 4-grams on each side of the source word that is aligned with the target word \(t_i\). The bilingual characteristic of the models makes it comparable to the MML method which trains source- and target-side language models from in- and out-domains separately and take a sum of cross-entropy differences over each side of the corpus:

\[
\text{score}(s, t) = [H_{I-\text{src}}(s) - H_{O-\text{src}}(s)] + [H_{I-tgt}(t) - H_{O-tgt}(t)]
\]

where \(s\) and \(t\) are sequences of source and target strings respectively. Out-domain models are trained by randomly selecting corpora of same size as that of the in-domain data.

### 4 Experiments

**Data:** We used TED talks (Cettolo et al., 2014) as our in-domain corpus. For German-to-English (DE-EN), we used the data made available for WMT’14.\(^2\) This contains News, Europarl and Common Crawl as out-domain data. For Arabic-English (AR-EN), we used the UN corpus as out-domain data. We concatenated dev- and test-2010 for tuning and used test2011-2013 for evaluation. Table 2 shows the size of the training and test data used.

**NNJM Settings:** The NNJM models were trained using NPLM\(^3\) toolkit (Vaswani et al., 2013) with the following settings. We used a target context of 5 words and an aligned source window of 9 words, forming a joint stream of 14-grams for training. We restricted source and target side vocabularies to 20K and 40K most frequent words. We used an input embedding layer of 150

---

\(^2\)http://www.statmt.org/wmt14/
\(^3\)http://nlg.isi.edu/software/nplm/
and an output embedding layer of 750. Only one hidden layer is used with NCE\(^4\) to allow faster training and decoding. Training was done using mini-batch size of 1000 and using 100 noise samples. We train the out-domain NNJM models using the same vocabulary as the in-domain vocabulary. All models were trained for 25 epochs.

**Machine Translation Settings:** We followed Birch et al. (2014) to train a Moses system Koehn et al. (2007) with the following settings: maximum sentence length of 80, Fast-Align (Dyer et al., 2013) for word-alignments, an interpolated Kneser-Ney smoothed 5-gram language model (Schwenk and Koehn, 2008) with KenLM (Heafield, 2011) for querying, lexicalized re-ordering (Galley and Manning, 2008) and other default parameters. We used Moses implementations of OSM and NNJM as a part of their respective baseline systems. Arabic OOVs were translated using an unsupervised transliteration module (Durrani et al., 2014b) in Moses. We used k-best batch MIRA (Cherry and Foster, 2012) for tuning.\(^5\)

### 4.1 Results: Model Weighting

We first discuss the results of applying mixture modeling approach. The MT systems are trained on a concatenation of all in- and out-domain data. The OSM and NNJM models used in baseline MT systems were also trained on the concatenated data.

Linear interpolation (OSM\(_{ln}\)) based on EM-weighting shows significant improvements with average BLEU gains of +0.6 in DE-EN and +0.9 in AR-EN over the baseline system B\(_{cat}\) (see Table 3).\(^6\) One reason for better gains in AR-EN is the fact that the out-domain UN data

---

\(^4\)Training NNJM with backpropagation could be prohibitively slow because for each training instance, the softmax layer requires a summation over the entire output vocabulary. One way to avoid this repetitive computation is to use a Noise Contrastive Estimation or NCE (Gutmann and Hyvärinen, 2010) of the loss function. NCE has been recently used in neural language models (Vaswani et al., 2013; Mnih and Teh, 2012).

\(^5\)All systems were tuned three times.

\(^6\)We carried out additional experiments by linearly interpolating class-based OSM models Durrani et al. (2014a). We used the mkcls utility in GIZA to cluster source and target vocabularies into 50 classes. Class-based OSM models were trained on each domain and interpolated in the same way as we did for the word forms. This however, did not yield any significant improvements on top of what was already achieved from the interpolation of word-based OSM. We also tried interpolating POS, morph and lemma-based OSM-models but did not gain any further improvement. Results are omitted from the paper.
### OSM Interpolation (German-English)

<table>
<thead>
<tr>
<th>System</th>
<th>test11</th>
<th>test12</th>
<th>test13</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_{cat}</td>
<td>35.8</td>
<td>31.1</td>
<td>27.6</td>
<td>31.5</td>
</tr>
<tr>
<td>OSM_{ln}</td>
<td>36.6 +0.8</td>
<td>31.9 +0.8</td>
<td>27.7 +0.1</td>
<td>32.1 +0.6</td>
</tr>
<tr>
<td>OSM_{lg}</td>
<td>35.4 -0.4</td>
<td>31.1 ± 0.0</td>
<td>27.4 -0.2</td>
<td>31.3 -0.2</td>
</tr>
</tbody>
</table>

### OSM Interpolation (Arabic-English)

<table>
<thead>
<tr>
<th>System</th>
<th>test11</th>
<th>test12</th>
<th>test13</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_{cat}</td>
<td>26.4</td>
<td>29.2</td>
<td>29.9</td>
<td>28.5</td>
</tr>
<tr>
<td>OSM_{ln}</td>
<td>27.3 +0.9</td>
<td>30.0 +0.8</td>
<td>30.8 +0.9</td>
<td>29.4 +0.9</td>
</tr>
<tr>
<td>OSM_{lg}</td>
<td>25.8 -0.6</td>
<td>28.7 -0.5</td>
<td>29.4 -0.5</td>
<td>28.0 -0.5</td>
</tr>
</tbody>
</table>

Table 3: OSM Interpolation OSM_{ln} = Linear, OSM_{lg} = Log-linear

### NNJM Interpolation (German-English)

<table>
<thead>
<tr>
<th>System</th>
<th>test11</th>
<th>test12</th>
<th>test13</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_{cat}</td>
<td>35.6</td>
<td>31.3</td>
<td>27.4</td>
<td>31.4</td>
</tr>
<tr>
<td>NNJM_{ln}</td>
<td>36.2 +0.6</td>
<td>31.8 +0.5</td>
<td>27.1 -0.3</td>
<td>31.7 +0.3</td>
</tr>
<tr>
<td>NNJM_{lg}</td>
<td>36.1 +0.5</td>
<td>32.1 +0.8</td>
<td>27.2 -0.2</td>
<td>31.8 +0.4</td>
</tr>
</tbody>
</table>

### NNJM Interpolation (Arabic-English)

<table>
<thead>
<tr>
<th>System</th>
<th>test11</th>
<th>test12</th>
<th>test13</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_{cat}</td>
<td>26.6</td>
<td>29.4</td>
<td>30.1</td>
<td>28.7</td>
</tr>
<tr>
<td>NNJM_{ln}</td>
<td>26.7 +0.1</td>
<td>30.2 +0.8</td>
<td>30.3 +0.2</td>
<td>29.1 +0.4</td>
</tr>
<tr>
<td>NNJM_{lg}</td>
<td>26.8 +0.2</td>
<td>30.2 +0.8</td>
<td>30.5 +0.4</td>
<td>29.2 +0.5</td>
</tr>
</tbody>
</table>

Table 4: NNJM Interpolation NNJM_{ln} = Linear, NNJM_{lg} = Log-linear

is much harmful for the task at hand. On the contrary additional data in DE-EN is helpful (see also the results in next section for more information). Log-linear interpolation of OSM models (OSM_{lg}) performs much worse than B_{cat} in both language pairs. In the log-linear model, all sub-models are queried separately. An operation sequence from the out-domain data that is unknown to the in-domain OSM, gets high probability\(^7\) and is ranked higher in the search space. On the contrary, the same gets down-weighted in a linearly interpolated global model.

Both linear and log-linear interpolation of the NNJM models showed improvements over the baseline system B_{cat} (refer to Table 4). Log-linear interpolation (NNJM_{lg}) performed slightly better in both cases. Notice that NNJM_{lg} does not face the same problem as OSM_{lg} because all NNJM models are trained using the in-domain vocabulary with a low probability assigned to the out-domain UNKs.\(^8\) See Joty et al. (2015) for more details on our novel handling

---

\(^7\) Due to probability mass assigned to UNK sequences.

\(^8\) In order to reduce the training time and to learn better word representations, neural models are trained on most frequent vocabulary words only and low frequency words are represented under a class of unknown words, unk. This results in a large number of n-gram sequences containing at least one unk word and thereby, makes unk a highly probable word for the model. As a result of this discrepancy, sentences with more number of unk words will be selected. To solve this problem we created a separate class for out-domain unk_o words. We train the in-domain model by adding a few dummy sequences containing unk_o occurring on both source and target sides ensuring that out-domain unknown words get minimal probabilities.
Table 5: MML, OSM and NNJM-based data selection, evaluated using test2011

<table>
<thead>
<tr>
<th>Percentage</th>
<th>German-English</th>
<th>Arabic-English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MML</td>
<td>OSM</td>
</tr>
<tr>
<td>0%</td>
<td>35.4</td>
<td>35.4</td>
</tr>
<tr>
<td>5%</td>
<td>36.0</td>
<td>36.0</td>
</tr>
<tr>
<td>10%</td>
<td>36.2</td>
<td>36.3</td>
</tr>
<tr>
<td>20%</td>
<td>36.4</td>
<td>36.8</td>
</tr>
<tr>
<td>40%</td>
<td>36.3</td>
<td>36.6</td>
</tr>
<tr>
<td>100%</td>
<td>35.6</td>
<td>35.6</td>
</tr>
</tbody>
</table>

Table 6: Data Selection

Data Selection (German-English)

<table>
<thead>
<tr>
<th>System</th>
<th>test11</th>
<th>test12</th>
<th>test13</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_{100%}</td>
<td>35.8</td>
<td>31.1</td>
<td>27.6</td>
<td>31.5</td>
</tr>
<tr>
<td>B_{0%}</td>
<td>35.4</td>
<td>31.3</td>
<td>25.5</td>
<td>30.7</td>
</tr>
<tr>
<td>MML_{20%}</td>
<td>36.4 +0.6</td>
<td>31.4 +0.3</td>
<td>27.7 +0.1</td>
<td>31.8 +0.3</td>
</tr>
<tr>
<td>OSM_{20%}</td>
<td>36.8 +1.0</td>
<td>31.5 +0.4</td>
<td>27.7 +0.1</td>
<td>32.0 +0.5</td>
</tr>
<tr>
<td>NNJM_{20%}</td>
<td>36.9 +1.1</td>
<td>31.6 +0.5</td>
<td>27.7 +0.1</td>
<td>32.1 +0.6</td>
</tr>
</tbody>
</table>

Data Selection (Arabic-English)

<table>
<thead>
<tr>
<th>System</th>
<th>test11</th>
<th>test12</th>
<th>test13</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>B_{100%}</td>
<td>26.4</td>
<td>29.2</td>
<td>29.9</td>
<td>28.5</td>
</tr>
<tr>
<td>B_{0%}</td>
<td>27.2</td>
<td>30.0</td>
<td>30.2</td>
<td>29.1</td>
</tr>
<tr>
<td>MML_{5%}</td>
<td>27.6 +0.4</td>
<td>30.5 +0.5</td>
<td>31.0 +0.8</td>
<td>29.7 +0.6</td>
</tr>
<tr>
<td>OSM_{5%}</td>
<td>27.7 +0.5</td>
<td>30.6 +0.6</td>
<td>31.0 +0.8</td>
<td>29.8 +0.7</td>
</tr>
<tr>
<td>NNJM_{5%}</td>
<td>27.6 +0.4</td>
<td>30.5 +0.5</td>
<td>31.1 +0.9</td>
<td>29.7 +0.6</td>
</tr>
</tbody>
</table>

Table 6: Data Selection

of UNK words in the NNJM model.

### 4.2 Results: Data Selection

We selected 0\%, 2.5\%, 5\%, 10\%, 20\%, 40\% and 100\% out-domain data and evaluated on test2011 to select the best percentage. See Table 5 for results on each selected percentage. Table 6 shows that the out-domain data is helpful in the case of DE-EN and harmful in the case of AR-EN; compare B_{100\%} (all data) versus B_{0\%} (in-domain data only). MML-selection improves the baseline by +0.3 and +0.6 in case of DE-EN and AR-EN respectively. OSM and NNJM-based selection gave similar improvements with slightly better results than MML. We found that the amount of overlap in data selected by the three models is roughly 63\% in DE-EN and 71\% in AR-EN.

### 5 Related Work

Previous work on domain adaptation in MT can be broken down broadly into two main categories namely data selection and model adaptation.
5.1 Data Selection

Data selection has shown to be an effective way to discard poor quality or irrelevant training instances, which when included in the MT systems, hurts its performance. The idea is to score the out-domain data using model trained from the in-domain data and apply a cut-off based on the resulting scores. The MT system can then be trained on a subset of the out-domain data that is closer to in-domain. Selection based methods can be helpful to reduce computational cost when training is expensive and also when memory is constrained. Data selection was earlier done for language modeling using information retrieval techniques (Hildebrand et al., 2005) and using perplexity measure (Moore and Lewis, 2010). Axelrod et al. (2011) further extended the work of Moore and Lewis (2010) to translation model adaptation by using both source side and target side language models. Duh et al. (2013) used recurrent neural network language model instead of an ngram-based language model to do the same. Translation model features were used recently by Liu et al. (2014); Hoang and Sima’an (2014) to do data selection.

5.2 Model Adaptation

The downside of data selection is that finding an optimal cut-off threshold is a time consuming process. Therefore rather than filtering less useful data, an alternative way is to down-weight it and boost the data closer to the in-domain. It is robust than selection since it takes advantage of the complete out-domain data with intelligent weighting towards the in-domain. Matsoukas et al. (2009) proposed a classification-based sentence weighting method for adaptation. Foster et al. (2010) extended this by weighting phrases rather than sentence pairs. Other researchers have carried out weighting by merging phrase-tables through linear interpolation (Finch and Sumita, 2008; Nakov and Ng, 2009) or log-linear combination (Foster and Kuhn, 2009; Bisazza et al., 2011; Sennrich, 2012) and through phrase training based adaptation (Mansour and Ney, 2013). Chen et al. (2013b) used vector space model for adaptation at phrase level. Every phrase pair is represented as a vector where every entry in the vector reflects its relatedness with each domain. Chen et al. (2013a) also applied mixture model adaptation for reordering model. Joty et al. (2015) performed model weighting by regularizing the loss function towards the in-domain model directly inside neural network training. They also used NNJM model as their basis.

Other work on domain adaptation includes but not limited to studies that focus on topic modeling (Eidelman et al., 2012; Hasler et al., 2014), dynamic adaptation where no in-domain data is available (Sennrich et al., 2013; Mathur et al., 2014) and sense disambiguation (Carpuat et al., 2013).

6 Conclusion

We targeted an unexplored area of using bilingual language models for domain adaptation. We applied model weighting and data selection techniques using OSM and NNJM models. Both methods were shown to be effective in the target translation tasks. Interpolating multi-domain models gave an average improvement of up to +0.9 BLEU points using OSM and +0.5 using NNJM. We also used NNJM and OSM models for data selection using differences in cross entropy and showed improvements of up to +0.6 BLEU points. The code will be contributed to Moses git repository.

References


Machine translation evaluation made fuzzier:
A study on post-editing productivity and evaluation metrics in commercial settings

Carla Parra Escartín
Manuel Arcedillo
Hermes Traducciones
C/ Cólquide 6, portal 2, 3. I
28230 Las Rozas, Madrid, Spain

Abstract
In this paper, we report on an experiment carried out in the context of a translation company. Ten translators, with diverse degrees of experience in translation and machine translation post-editing (MTPE), were assigned the same task, involving translation from scratch, fuzzy-match post-editing, and MTPE. We evaluate the MT output using traditional evaluation metrics such as BLEU and TER, correlate these measures with productivity values and study whether a fuzzy score stands up against them. Our main goal was to evaluate whether fuzzy scores can be used for evaluating MTPE, thus incorporating its familiarity and TM matching analogies to an MTPE workflow. The results of our experiment seem to support this hypothesis.

1 Introduction
In the last years, machine translation post-editing (MTPE) tasks have become ever more common in the translation industry. Translators and translation companies of all sizes use machine translation (MT) to increase their productivity and reduce production costs. However, it is still unclear how to assess MT output in order to verify that those goals have been met and, if applicable, determine a fair compensation for the post-editor.

Traditional automatic evaluation metrics such as BLEU (Papineni et al., 2001) and TER (Snover et al., 2006) have shortcomings such as unproven correlation with productivity gains (Callison-Burch et al., 2006), technical difficulties for their estimation by general users and lack of intuitiveness. Meanwhile, the translation industry has a well-established way of evaluating text similarity and establishing discount rates: translation memory (TM) fuzzy match scores (FMS). Based on text similarity with segments stored in the TM, each sentence to be translated receives a fuzzy score, ranging from 0% (no similar sentence was found in the TM) to 100% (an exact match was found). In the translation industry, TM matches above 75% are normally assigned a rate discount, while those segments below this threshold are paid the full rate following the general assumption that they do not yield any productivity increase. Since translators and other parties involved in a translation project are familiar with this system and accept it as a valid business model, we designed an experiment to verify whether target-side FMS stand up

1This is an industry general practice. Plitt and Masselot (2010) also report that in a typical localization scenario TM matches below 75% are considered no-matches. However, lack of research and the peculiarities of fuzzy score calculation in each tool may allow for variations of this model.
against traditional methods of MTPE evaluation.

In the experiment, ten professional translators with diverse experience in the industry were assigned the same file to translate, involving translation from scratch, TM match post-editing, and MTPE. We recorded the time spent by each translator in each segment in order to obtain productivity values for all TM match bands, including no match segments and exact matches. We then computed several automated metrics (namely, BLEU and TER) and correlated them with productivity values. Finally, we compared the performance of these automated metrics with the performance of the target-side FMS.

The remainder of this paper is structured as follows: Section 2 summarizes traditional MT evaluation metrics (c.f. 2.1 and 2.2) and presents the metric we use in our experiments, explaining why we deem it appropriate for MT evaluation (c.f. 2.3). The details of the experiment itself are explained in Section 3. In Section 4 we present the results and analyze them. Finally, in Section 5 we summarize the findings and discuss possible future research.

2 Background

A common practice when negotiating MTPE discounts is to either annotate a sample of the MT output according to any of the methods described below or similar ones, or post-edit it in order to generate a reference for automatic evaluation, while possibly gathering other data such as time spent or keystrokes. Since rates are normally set before handing off a project, a good MT evaluation at this stage is critical to avoid underpayments and the mistrust this situation may cause.

MT evaluation is a research field in itself. As stated by King et al. (2003), Yorick Wilks has been credited with the famous remark that “more has been written about MT evaluation than about MT itself”. While it is not the purpose of this paper to revise all the proposed MT evaluation metrics, we deem it necessary to acknowledge the most commonly used metrics and assess their usability from the translation industry point of view. Subsection 2.1 summarizes human evaluation and Subsection 2.2 focuses in automatic evaluation metrics. Subsection 2.3 explains the shortcomings of traditional evaluation metrics and presents the metric we additionally used in our experiments: the target-side FMS.

2.1 Human evaluation

As explained by Koehn (2010), two main strategies are used in evaluation campaigns: fluency and adequacy, and ranking of translations. When human judges are asked to assess the fluency and adequacy of MT output, the task consists of assigning a score from one to five on these two criteria. Fluency assesses whether the text is fluent in the target language. It refers to grammaticality, correctness and idiomatic word choices. Adequacy, on the other hand, assesses whether the output sentence conveys the same meaning as the input sentence.

As pointed out by Koehn (2010), “these definitions are very vague, and it is difficult for evaluators to be consistent in their application”. Callison-Burch et al. (2007) also point this out: “No instructions are given to evaluators in terms of how to quantify meaning, or how many grammatical errors (or what sort) separates the different levels of fluency. Because of this many judges either develop their own rules of thumb, or use the scales as relative rather than absolute.” To overcome these issues, the “translation ranking” approach may be taken. In this kind of evaluation, human judges are given the output of different systems and are asked to choose the best one. This approach is based on the judges’ perception of usefulness in terms of savings and productivity, rather than on usefulness itself. In a translation and productivity test carried out by Autodesk in 2011, a clear mismatch between perception of productivity and actual productivity was found.

http://langtech.autodesk.com/productivity.html
An alternative evaluation methodology is the one proposed by Hurtado Albir (1995). This methodology is widely used in translator training courses. It distinguishes different types of errors such as orthotypographical, grammatical, or semantic errors. More recent proposals are the TAUS Dynamic Quality Framework (DQF), the Multidimensional Quality Metrics (MQM) Framework proposed by the QTLaunchPad project (Burchardt and Lommel, 2014), and their harmonized version, recently released (Lommel, 2015). However, since human evaluation (as opposed to automatic evaluation) is costly and time consuming, these methods are not always feasible.

2.2 Automatic evaluation

Throughout the years, several automatic evaluation metrics have been proposed. These can be grouped in lexical, syntactic and semantic evaluation metrics depending on the linguistic level at which they operate.

Lexical evaluation metrics assess the lexical similarity between the MT output and one or several references. These metrics assume that the usefulness of MT is related to its proximity or similarity to the reference(s). This assumption is most informative when such reference(s) are post-edits of the MT output, rather than previous translations.

They can be further classified into metrics measuring the edit distance, the lexical precision, the lexical recall and the F-Measure. Edit distance is measured by WER (Word Error Rate) (Nießen et al., 2000) and TER (Translation Edit Rate) (Snover et al., 2006, 2009). Lexical precision is measured by BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2001) and NIST (Doddington, 2002). ROUGE (Lin and Och, 2004) assesses the lexical recall and PER (Position-independent Word Error Rate) (Tillmann et al., 1997) assesses the lexical precision and recall. Finally, $GM_r$ (Melamed et al., 2003) and METEOR (Lavie and Agarwal, 2005; Denkowski and Lavie, 2010) are based on the F-Measure.

MT evaluation tools such as Asiya (Giménez and Márquez, 2010) additionally offer syntactic and semantic similarity evaluation metrics. The syntactic metrics capture similarities over shallow-syntactic structures, dependency relations and constituent parse trees. The semantic metrics intend to capture similarities over named entities, semantic roles and discourse representations. Both the syntactic and the semantic metrics require the usage of additional Language Resources and Tools (LRT), such as parsers, corpora, and specific packages.

All these metrics share the advantage of being fast and cheap to obtain. However, they still remain obscure for laymen and professional translators not familiarized with MT evaluation research, and typically are not designed with MTPE in mind.

2.3 Fuzzy match scores for MT evaluation

Although BLEU is a well-established evaluation metric and it is extensibly used in MT engine development cycles and the evaluation of raw MT output as a final product, it has also received criticism (Callison-Burch et al., 2006; Koehn, 2010; Bechara, 2013). A general rule of thumb in MT evaluation is that BLEU scores above 30 reflect understandable translations, while scores over 50 can be considered good and fluent translations (Lavie, 2010). However, the usefulness of “understandable” translations is questionable in the case of MTPE, since post-editors do not depend on MT to understand the meaning of the source text. Also, how should one interpret a BLEU score improvement from 45 to 50 in terms of productivity? Taking another widely-used metric, does a TER value of 40 justify any discount? The vast majority of translators (and even MT researchers) would probably be unable to answer these questions. And yet, they would probably instantly acknowledge that TM fuzzy matches of 60% are not worth editing, while

---

3https://evaluate.taus.net/evaluate/about
4See, for instance, Section 3, "Costs", in Burchardt and Lommel (2014).
they would probably consider it fair to accept discounts for 80% matches. We believe MTPE
tasks would benefit greatly from the familiarity of source-side FMS applied to MTPE evaluation
as a target-side FMS.\textsuperscript{5}

Organizations such as TAUS have already proposed alternative models of MT evaluation
which use FMS, such as the “MT Reversed Analysis”.\textsuperscript{6} Also, Zhechev (2012) created a met-
ic combining character and word-based FMS and reported it as having the greatest correlation
with productivity gain. In this experiment, bearing in mind that each CAT tool uses a different
(and generally unknown) algorithm for their source-side FMS, we use the fuzzy value com-
puted by Okapi Rainbow\textsuperscript{7} in its Translation Comparison feature. We use this tool because it is
freely available and it natively supports the most common bilingual formats, hence potentially
improving transparency and usability for all parties involved in a translation project. This FMS
is based on the \textit{Sørensen-Dice} coefficient (Sørensen, 1948; Dice, 1945) using 3-grams. Equation
1 illustrates how the FMS is computed by Okapi Rainbow. An important difference with
other automatic metrics is that it does not depend on traditional tokenization: instead of word
n-grams, it computes character n-grams. Segments are split into a list of characters, which is
then used to compute 3-grams and string size.\textsuperscript{8}

\begin{equation}
\frac{2 \ast (MT\text{\_output} \cap PE\text{\_output})}{\text{size}\_MT\text{\_output} + \text{size}\_PE\text{\_output}} \ast 100
\end{equation}

The MT output (or TM output in the case of TM matches) was compared against the post-
edited output of each translator. This means that each segment has ten separate scores, one for
each translator. It is also worth noting here that translators faced either MT output or TM output
—or no suggestion at all in segments selected for translation from scratch—, so the original output
is unambiguous.

\section{Experiment settings}

The experiment was based on a pilot experiment (Parra Escartín and Arcedillo, 2015) which
seemed to suggest that fuzzy scores might be as good as or even better than BLEU and TER
scores for evaluating MTPE. Similarly to what Federico et al. (2012) did, we replicated a real
production environment.

Ten in-house translators were asked to translate the same file from English into Spanish.
We asked them to translate the text using one of their most common CAT tools, memoQ.\textsuperscript{9}
This tool was chosen because it allows keeping track of the time spent in each segment. We
disregarded using other tools which also record time and other useful segment-level indicators,
such as keystrokes in PET (Aziz et al., 2012) and iOmegaT (Moran et al., 2014), or MTPE
effort in MateCat (Federico et al., 2012) because they are not part of the everyday tools of the
translators involved. Translators were only allowed to use the TM, the terminology database and
the MT output included in the translation package. We disabled all other memoQ’s productivity
enhancing features such as predictive text, sub-segment leverage and automatic fixing of TM
matches to allow for better comparisons with translation environments which may not offer
similar features.

\textsuperscript{5}Hereinafter, FMS will refer exclusively to target-side FMS unless otherwise specified.
\textsuperscript{7}http://okapi.opentag.com/
\textsuperscript{8}For further information, see the open source code available at: https://bitbucket.org/okapiframework/.
\textsuperscript{9}The version used was memoQ 2015 build 3.
3.1 Test set selection

The selection criteria for the text to be translated were the following:

1. Belong to a real translation request.
2. Originate from a client for which our company owned a customized MT engine.
3. Have a word volume capable of engaging translators for several hours.
4. Include significant word counts for each TM match band (i.e., exact matches, fuzzy matches and no-match segments).

We used the word count analysis already available in our company’s archive to find a translation project which met our criteria. In this case, the existing word count analysis came from SDL Trados Studio. The original source text had over 8,000 words and was part of a software user guide. To avoid skewing due to the inferior typing and cognitive effort required to translate the second of two similar segments, repetitions and internal leverage segments were filtered out.

When transferring the project to memoQ (the CAT tool chosen for the experiment), we encountered a quite different word count. We expected some discrepancies in this new analysis, but they turned out to be bigger than we preliminary thought. Table 1 shows the word count distribution reported by both tools. As can be observed in Table 1, memoQ’s word count for the 95–99% fuzzy band ended up being significantly smaller despite our efforts to select a text with a more balanced distribution. This is evidence of a well-known problem in the translation industry: the different algorithms used by each tool for computing word counts and calculating fuzzy scores.

<table>
<thead>
<tr>
<th>TM match</th>
<th>SDL Trados Studio</th>
<th>memoQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Words</td>
<td>Segments</td>
</tr>
<tr>
<td>100%</td>
<td>1243</td>
<td>95</td>
</tr>
<tr>
<td>95–99%</td>
<td>1044</td>
<td>55</td>
</tr>
<tr>
<td>85–94%</td>
<td>747</td>
<td>43</td>
</tr>
<tr>
<td>75–84%</td>
<td>608</td>
<td>42</td>
</tr>
<tr>
<td>No Match</td>
<td>3388</td>
<td>233</td>
</tr>
<tr>
<td>Total</td>
<td>7030</td>
<td>468</td>
</tr>
</tbody>
</table>

Table 1: Word count as computed by SDL Trados Studio and memoQ.

The differences in word count unfortunately created a less informative 95–99% fuzzy band. On the other hand, the increased number of no-match words (from 3388 to 3804) provided us with a more solid sample for comparing MTPE and translation throughputs. In order to make this comparison, we randomly divided the no-match band in two halves using the test set generator included in the m4loc package. One half was translated from scratch, and the other half was machine translated and subsequently post-edited.

3.2 Machine translation system used

To generate the raw MT output we used a customized Systran’s RBMT engine. This system was selected because it is the one we normally use for MTPE for this client. It can be

10 The version used was SDL Trados Studio 2014 SP1.
11 m4loc is an open source project initiative to “combine investments to efficiently leverage the potential that Moses promises for localization” (Ruopp, 2010) For more information and access to the code see: https://code.google.com/p/m4loc
12 Systran 7 Premium Translator was used.
considered a mature engine since at the time of the experiment it had been subject to ongoing in-house customization for over three years and boasted a consistent record for productivity enhancement. The customization includes dictionary entries, software settings, and pre- and post-editing scripts. Although Systran includes a statistical machine translation (SMT) component, this was not used in our experiment. This decision was taken based on our past experience. In our experiments using this component, it seemed to negate the strengths of RBMT engines (such as predictability and orthotipographic consistency), while not incorporating enough improvements to compensate for them. While we are aware of recent advances in this topic, this component is not part of our usual MTPE preparation workflow and thus we chose not to use it in our experiment.

3.3 Translators participating in the experiment

Ten translators were engaged in the experiment. Two carried out the task as part of a pilot experiment (Parra Escartín and Arcedillo, 2015) and eight more were engaged with the aim of verifying the initial findings. A preliminary survey was sent to all translators to gather information on their experience. Translators were asked to provide their years of experience in translation and MTPE, their experience translating and/or MTPE texts from this client, their opinion on MT (positive or negative), and their opinion on CAT tools (positive or negative). Table 2 summarizes the results of our survey. Translators are sorted in descending order according to their combined experience in translation and MTPE.

<table>
<thead>
<tr>
<th>Trans. exp.</th>
<th>MTPE exp.</th>
<th>Client exp.</th>
<th>MT opinion</th>
<th>CAT opinion</th>
</tr>
</thead>
<tbody>
<tr>
<td>TR1</td>
<td>5</td>
<td>3</td>
<td>Yes</td>
<td>Positive</td>
</tr>
<tr>
<td>TR2</td>
<td>5</td>
<td>3</td>
<td>Yes</td>
<td>Positive</td>
</tr>
<tr>
<td>TR3</td>
<td>5.5</td>
<td>2</td>
<td>Yes</td>
<td>Positive</td>
</tr>
<tr>
<td>TR4</td>
<td>5</td>
<td>1</td>
<td>Yes</td>
<td>Negative</td>
</tr>
<tr>
<td>TR5</td>
<td>5</td>
<td>0</td>
<td>No</td>
<td>Positive</td>
</tr>
<tr>
<td>TR6</td>
<td>5</td>
<td>0</td>
<td>No</td>
<td>Positive</td>
</tr>
<tr>
<td>TR7</td>
<td>2</td>
<td>1</td>
<td>Yes</td>
<td>Positive</td>
</tr>
<tr>
<td>TR8</td>
<td>1.5</td>
<td>1.5</td>
<td>Yes</td>
<td>Positive</td>
</tr>
<tr>
<td>TR9</td>
<td>2</td>
<td>0</td>
<td>No</td>
<td>Negative</td>
</tr>
<tr>
<td>TR10</td>
<td>1</td>
<td>0</td>
<td>No</td>
<td>Positive</td>
</tr>
</tbody>
</table>

Table 2: Overview of translator’s experience (measured in years) and opinion on MT and CAT.

As some translators may be biased against MT, we deemed it necessary to gather their view on it as positive or negative, to see if this may also have an impact in the results. Only two translators expressed that they did not like working with MT or MTPE, even though they acknowledged that high quality MT output generally enhances their productivity.

4 Results and discussion

The ten packages received from the translators were analyzed individually in order to extract the time spent in each segment and calculate automated evaluation metrics. Segments with exceptionally long times (above 10 minutes) and those with unnaturally high productivity (above 5 words per second) were considered outliers and were discarded for further analysis. The few exceptionally long times registered can be explained by the translator not “closing” the segment during long pauses or lunch breaks. Since the two segments with highest recorded times (above one hour) belong to the translators who openly hold a grudge against MT, it is tempting to interpret this as an attempt to pervert the results. However, since the rest of their samples showed no other data anomalies and the time at which those exceptions were edited matches
translators’ normal lunch breaks, it can be safely assumed that those two cases were exceptions
due to carelessness rather than deliberate data tampering.

A possible explanation for the segments with unnaturally high productivity might be that
the translator actually proofread the translation when having the cursor placed in the previous
segment. Some of these segments were 100% matches and thus it seems logical that they only
opened the segment to confirm it and move to the next one. However, we lack a way to confirm
this hypothesis.

Data series of Translators 5 and 8 also showed questionable results. Translator 5’s MT
post-edited output quality showed clear signs of under-editing, which would not have reached
the minimum quality standards required by the client in a real project. This under-editing can
also be seen when comparing MTPE sample’s automated metrics across translators. As table
3 shows, values for Translator 5 differ significantly from the rest. Closer analysis of the fuzzy
and translation-from-scratch final text also showed significantly more errors than in the rest of
translators. Although it may prove an interesting starting point for further research on MTPE
and translation errors, we have omitted the analysis of this data series for the purpose of this
paper.

<table>
<thead>
<tr>
<th></th>
<th>TR-1</th>
<th>TR-2</th>
<th>TR-3</th>
<th>TR-4</th>
<th>TR-5</th>
<th>TR-6</th>
<th>TR-7</th>
<th>TR-8</th>
<th>TR-9</th>
<th>TR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLEU</td>
<td>64.24</td>
<td>66.37</td>
<td>64.23</td>
<td>63.40</td>
<td><strong>78.16</strong></td>
<td>69.57</td>
<td>66.42</td>
<td>63.56</td>
<td>65.88</td>
<td>68.43</td>
</tr>
<tr>
<td>TER</td>
<td>22.14</td>
<td>20.98</td>
<td>22.32</td>
<td>24.52</td>
<td><strong>12.82</strong></td>
<td>18.89</td>
<td>19.74</td>
<td>21.40</td>
<td>19.77</td>
<td>18.84</td>
</tr>
<tr>
<td>FMS</td>
<td>87.29</td>
<td>87.74</td>
<td>87.05</td>
<td>86.30</td>
<td><strong>93.06</strong></td>
<td>88.53</td>
<td>88.40</td>
<td>87.62</td>
<td>88.85</td>
<td>89.09</td>
</tr>
</tbody>
</table>

Table 3: Automated metrics for the MTPE sample per translator.

Translator 8’s anomaly was detected when studying correlation values between automated
metrics and productivity. In this case, it was significantly lower than for the rest of translators
(see Table 9 in Section 4.3). Upon further enquiries, Translator 8 admitted to have enabled the
predictive text feature which had been originally disabled from the hand-off package. Again,
although extremely interesting for future research, for the purpose of this paper Translator 8’s
data series was discarded.

One last remark about possible anomalies is the generally low productivity of the 95–99%
TM match band when compared to contiguous TM match bands (Table 4). In the vast majority
of these segments there were no textual differences between the match stored in the TM and the
content to be translated. This is illustrated in Example 2.

(2) **Source in TM:** If protection has been disabled manually, you can enable it in the
application settings window.

**Target in TM:** Si se ha desactivado la protección manualmente, puede activarla en la
ventana de configuración de la aplicación.

**New Source:** If protection has been disabled manually, you can `<1>` enable it in the
application settings window `</1>`.

**New Target:** Si se ha desactivado la protección manualmente, puede `<1>` activarla en la
ventana de configuración de la aplicación `</1>`.

The results obtained show that the time needed to perform these inline tag edits does not
match what would be expected of an equivalent 95% TM match due to text differences. Even
though it is possible to adjust the fuzzy score assigned by CAT tools to these segments via
formatting penalties (as in SDL Trados Studio) or by using tag weights (as in memoQ), the
impact of inline tag editing in translation throughput has not yet been extensively researched.
For this reason, we have omitted the 95–99% band in all following analysis.
4.1 Productivity report

Although a standard reference of 313–375 words per hour (2500–3000 words per day) is usually taken as a baseline for productivity comparisons, we calculated each translator’s translation-from-scratch throughput individually. As expected, it varied greatly, ranging from 473 to 1701 words per hour (Table 4). It is remarkable that Translator 1 was even able to translate from scratch faster than when post-editing 75–84% TM matches. In a real project, a rate discount would have been applied to those matches, which apparently would have been unfair for this translator.

Not depending on standard reference values also avoided misleading interpretations of MTPE throughputs. For example, comparing the average 1329 words per hour with a standard reference of 375 words per hour would show productivity gains of around 250%. However, based on the translation-from-scratch throughputs obtained, none of our translators reached such numbers, as can be seen in Table 5, which shows a more realistic average gain of “just” 24%. This data confirms the importance of having a translation reference for each sample instead of relying on standard values (Federico et al., 2012).

Table 4 reports the productivity (words per hour) obtained for each translator in each band. The words per hour average is provided in the last column.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>94</td>
<td>1226</td>
<td>3277</td>
<td>2942</td>
<td>1894</td>
<td>1767</td>
<td>1579</td>
<td>4039</td>
<td>2798</td>
</tr>
<tr>
<td>95–99%</td>
<td>21</td>
<td>231</td>
<td>2642</td>
<td>2625</td>
<td>1476</td>
<td>1299</td>
<td>963</td>
<td>2011</td>
<td>1133</td>
</tr>
<tr>
<td>85–94%</td>
<td>48</td>
<td>1062</td>
<td>2960</td>
<td>2248</td>
<td>1660</td>
<td>1678</td>
<td>1232</td>
<td>2164</td>
<td>2429</td>
</tr>
<tr>
<td>75–84%</td>
<td>42</td>
<td>696</td>
<td>1592</td>
<td>1592</td>
<td>1372</td>
<td>1140</td>
<td>1019</td>
<td>1342</td>
<td>1576</td>
</tr>
<tr>
<td>MTPE</td>
<td>131</td>
<td>1890</td>
<td>1804</td>
<td>1743</td>
<td>1369</td>
<td>1141</td>
<td>922</td>
<td>1481</td>
<td>1433</td>
</tr>
<tr>
<td>Trans.</td>
<td>132</td>
<td>1914</td>
<td>1701</td>
<td>1319</td>
<td>993</td>
<td>916</td>
<td>933</td>
<td>1236</td>
<td>1223</td>
</tr>
</tbody>
</table>

Table 4: Productivity achieved per translator and match band in words per hour.

We additionally computed the productivity gain achieved by each translator in each band and the average gain across translators. Equation 3 shows the formula used for calculating productivity gains, where \( PE_{Throughput} \) is the productivity achieved by one translator in words per hour when post-editing MT output or TM matches, and \( TRA_{Throughput} \) is the productivity achieved by that same translator when translating from scratch (taken from productivity values in Table 4). Table 5 reports the productivity gains obtained.

\[
\left( \frac{PE_{Throughput}}{TRA_{Throughput}} - 1 \right) \times 100
\]

(3)

<table>
<thead>
<tr>
<th>MTPE gains</th>
<th>TR-1</th>
<th>TR-2</th>
<th>TR-3</th>
<th>TR-4</th>
<th>TR-6</th>
<th>TR-7</th>
<th>TR-9</th>
<th>TR-10</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.06</td>
<td>32.15</td>
<td>37.78</td>
<td>24.62</td>
<td>-1.23</td>
<td>19.80</td>
<td>17.20</td>
<td>56.34</td>
<td>24.09</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Productivity gain percentage due to MTPE for each translator.

All translators except one (Translator 6, who experienced 1.2% productivity loss when facing MTPE) were faster in MTPE than translating from scratch, although the productivity gain varied greatly among them (from 6.06% to 56.34%). A possible explanation for the slight productivity loss experienced by Translator 6 might be that this translator had experienced an extensive period of inactivity and had barely used memoQ. The biggest productivity gain was achieved by the least experienced translator (Translator 10), while the smallest productivity gain corresponds to the most experienced translator (Translator 1), although this trend cannot be confirmed by the rest of results. Furthermore, all translators who had MTPE throughput...
higher than the 75–84% band (Translators 1, 2 and 7) had previous experience in MTPE. On the contrary, both translators (6 and 9) who worked faster with 75–84% TM matches than with MT segments had no MTPE experience. This seems to point out that mature MT engines allow experienced post-editors to post-edit MT faster than the lowest TM matches.

Also, it is worth noting that the productivity in Table 4 reflects only the time spent by the translator inside the editor. Other tasks such as reading of instructions, file management, escalation of queries, self-review, quality assurance and communication with project managers and/or other parties involved in the project are not directly reflected here, but are all part of a regular project. Care should also be taken when transferring productivity gains into rate discounts, as translation rates often include tasks which are not affected by the introduction of MTPE (such as project management, file engineering or review by a second linguist). In any case, it is interesting to see that MTPE can improve productivity even when the fastest translators work with content which allow for higher-than-usual translation throughputs.

4.2 MT evaluation metrics

We calculated document-level values for BLEU, TER and FMS\(^{13}\) (Tables 6, 7 and 8 respectively) taking the segments belonging to each band as separate documents. To allow for a more direct comparison, next to each band average we attach its corresponding average productivity gain, calculated as the average of the eight values (one for each translator) obtained according to Equation 3 above.

<table>
<thead>
<tr>
<th>TM Match Band</th>
<th>TR-1</th>
<th>TR-2</th>
<th>TR-3</th>
<th>TR-4</th>
<th>TR-6</th>
<th>TR-7</th>
<th>TR-9</th>
<th>TR-10</th>
<th>Avg. Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>93.17</td>
<td>92.38</td>
<td>85.57</td>
<td>92.33</td>
<td>91.96</td>
<td>96.28</td>
<td>94.25</td>
<td>95.51</td>
<td>92.68</td>
</tr>
<tr>
<td>95–99%</td>
<td>86.51</td>
<td>89.35</td>
<td>78.96</td>
<td>81.85</td>
<td>87.91</td>
<td>85.83</td>
<td>80.32</td>
<td>92.10</td>
<td>85.35</td>
</tr>
<tr>
<td>85–94%</td>
<td>82.22</td>
<td>81.00</td>
<td>76.15</td>
<td>80.70</td>
<td>80.19</td>
<td>85.62</td>
<td>84.39</td>
<td>88.23</td>
<td>82.31</td>
</tr>
<tr>
<td>75–84%</td>
<td>70.42</td>
<td>71.98</td>
<td>66.57</td>
<td>70.34</td>
<td>73.94</td>
<td>71.50</td>
<td>70.52</td>
<td>70.36</td>
<td>70.70</td>
</tr>
<tr>
<td>MTPE</td>
<td>64.24</td>
<td>66.37</td>
<td>64.23</td>
<td>63.40</td>
<td>69.57</td>
<td>66.42</td>
<td>65.88</td>
<td>68.43</td>
<td>66.07</td>
</tr>
</tbody>
</table>

Table 6: BLEU scores for each TM match band and average productivity gain (%).

<table>
<thead>
<tr>
<th>TM Match Band</th>
<th>TR-1</th>
<th>TR-2</th>
<th>TR-3</th>
<th>TR-4</th>
<th>TR-6</th>
<th>TR-7</th>
<th>TR-9</th>
<th>TR-10</th>
<th>Avg. Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>4.38</td>
<td>3.99</td>
<td>8.43</td>
<td>4.35</td>
<td>4.64</td>
<td>2.00</td>
<td>3.10</td>
<td>1.94</td>
<td>4.10</td>
</tr>
<tr>
<td>95–99%</td>
<td>8.24</td>
<td>6.37</td>
<td>14.02</td>
<td>11.50</td>
<td>6.91</td>
<td>8.31</td>
<td>12.83</td>
<td>5.30</td>
<td>9.19</td>
</tr>
<tr>
<td>75–84%</td>
<td>21.65</td>
<td>21.03</td>
<td>23.74</td>
<td>21.18</td>
<td>19.08</td>
<td>20.07</td>
<td>20.86</td>
<td>20.20</td>
<td>20.98</td>
</tr>
<tr>
<td>MTPE</td>
<td>22.14</td>
<td>20.98</td>
<td>22.32</td>
<td>24.52</td>
<td>18.89</td>
<td>19.74</td>
<td>19.77</td>
<td>18.84</td>
<td>20.90</td>
</tr>
</tbody>
</table>

Table 7: TER scores for each TM match band and average productivity gain (%).

The three metrics agree that the four less experienced translators (6–10) performed less edits to the MT output and the TM matches than the four more experienced translators (1–5). The issue reported in the beginning of Section 4 about the 95–99% band not achieving the expected productivity can also be seen in these metric tables. It would be logical to expect the productivity of this band being somewhere in between the 100% and the 85–94% bands, and the three metrics indeed agree with this intuition, but in terms of productivity it was not the case.

\(^{13}\)BLEU and TER were computed using Asiya (Giménez and Márquez, 2010) and the FMS was computed using the Okapi framework. For each translator and in each segment, the reference was only the post-edited MT or TM output delivered by that same translator. In no case multiple references were used.
As explained earlier, this can be explained by the fact that the vast majority of edits required in the 95–99% band involved dealing with inline tags. Although the impact of these operations has not been researched enough, these results show that they can have a big impact in terms of productivity, slowing down the translator more than would be expected. Moreover, when calculating automated metrics, these inline tags are generally deleted to avoid their division in different tokens, thus potentially skewing the results. Instead of deleting them, when calculating the automated metrics reported in this paper we converted each tag into a unique token. This operation was clearly not enough to compensate all the effort put into tag handling, as hinted by the productivity values. More research is needed in this area to find out the appropriate weight or penalty that automated metrics should assign to inline tags. As stated earlier, we opted to treat the 95–99% band as an outlier and ignore it from further analysis.

### 4.3 Productivity vs. MT evaluation metrics correlation

To find out if the proposed FMS stands up against traditional methods of MT evaluation, we correlated the metrics in Tables 6, 7 and 8 with the productivity values in Table 4. We also added two alternatives to the FMS calculated by Okapi Rainbow: the FMS provided by SDL Trados Studio and memoQ, two of the most popular CAT tools worldwide. As can be observed in Table 9, the results indicate that BLEU, TER and the three FMS have a strong correlation with productivity. Therefore, all of them seem suitable for evaluating MTPE or TM match productivity. The anomaly created by Translator 8 using the predictive text feature is also reflected in Table 9.

<table>
<thead>
<tr>
<th>TR-1</th>
<th>TR-2</th>
<th>TR-3</th>
<th>TR-4</th>
<th>TR-5</th>
<th>TR-6</th>
<th>TR-7</th>
<th>TR-8</th>
<th>TR-9</th>
<th>TR-10</th>
<th>Avg</th>
<th>Avg gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.934</td>
<td>0.952</td>
<td>0.996</td>
<td>0.934</td>
<td>0.999</td>
<td>0.925</td>
<td>0.816</td>
<td>0.995</td>
<td>0.933</td>
<td>0.953</td>
<td>0.961</td>
<td></td>
</tr>
<tr>
<td>0.973</td>
<td>0.995</td>
<td>0.987</td>
<td>0.945</td>
<td>0.989</td>
<td>0.968</td>
<td>0.725</td>
<td>0.984</td>
<td>0.973</td>
<td>0.977</td>
<td>0.953</td>
<td></td>
</tr>
<tr>
<td>0.967</td>
<td>0.987</td>
<td>0.955</td>
<td>0.930</td>
<td>0.942</td>
<td>0.968</td>
<td>0.749</td>
<td>0.987</td>
<td>0.969</td>
<td>0.963</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.929</td>
<td>0.962</td>
<td>0.903</td>
<td>0.838</td>
<td>0.934</td>
<td>0.998</td>
<td>0.579</td>
<td>0.910</td>
<td>0.999</td>
<td>0.933</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9: Pearson correlation between productivity and evaluation measures. Translator 8’s results are omitted from the average.

### 4.4 Discussion

Analyzing the results further, there seems to be benefits of using FMS over BLEU or TER to evaluate MTPE. Table 10 shows the average BLEU, TER, FMS and productivity gain values for MTPE and the 75–84% fuzzy band. The average gain due to MTPE was higher than the 75–84% band. However, according to BLEU scores, the situation should have been the opposite,
while according to TER both values should have been more or less the same. The FMS, on the other hand, accurately reflects the productivity gains obtained.\footnote{This pattern has also been observed in other MTPE projects performed in our company.}

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>TER</th>
<th>FMS</th>
<th>Prod. gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>75–84%</td>
<td>70.70</td>
<td>20.98</td>
<td>85.60</td>
<td>22.52%</td>
</tr>
<tr>
<td>MTPE</td>
<td>66.07</td>
<td>20.90</td>
<td>87.91</td>
<td>24.09%</td>
</tr>
</tbody>
</table>

Table 10: Average BLEU, TER, FMS and productivity gain for MTPE and 75–84% bands.

Another useful application of FMS is the insight they provide when setting the threshold between TM matching and segments sent to the MT engine. Normally, TM matches below 75% are assumed to yield no productivity increase. Therefore, when applying MTPE, the general approach is to process all segments in the 0–74% range via MT. However, if the MT output quality is high enough, it may be more productive to post-edit MT output rather than some of the TM matches, as the results of this experiment actually show. This issue was first brought to our attention by the post-editors themselves, who frequently suggested increasing the threshold to 85%. After internal testing, it was verified that the post-editors’ intuition was correct and a new threshold of 85% was established for this client.\footnote{As explained in Section 3.1, for the purposes of the experiment reported here, this higher threshold was not used, so all segments below 75% were considered no-match segments. Also, we do not imply that MTPE can always be faster than post-editing 75–84% TM matches. With poor output, that threshold should still be set at 75% or even lower, while better engines may benefit from higher thresholds.} Interestingly enough, that value is also the average FMS obtained with this client’s projects over a three-year period. In our experience, this method can also be applied successfully to other language pairs and higher FMS ranges. For example, the threshold of Spanish-Catalan engines, which generally perform better than more distant language pairs, is set at 95%, as suggested by post-editors, confirmed by testing and reflected in FMS. These experiences suggest that the FMS applied to MTPE evaluation can be used to adjust the threshold between TM matching and MT segments on a client, domain or engine basis.

Since FMS seems to perform as well as other widely used metrics and incorporates other benefits, it may be reasonable to use FMS as a metric for analyzing both TM leverage and MT output. Relying on two different scores for these technologies creates contradictory situations, where a client may implicitly acknowledge that TM fuzzy matches below 75% do not yield productivity gains, and yet consider MT output with 60% FMS as liable for discounts based on their own assumptions on BLEU, TER or other metric. A unified framework for combining evaluation and quality estimation of TM and MT technologies has been recently proposed by Forcada and Sánchez-Martínez (2015). Even though they do not consider MT quality indicators and TM matching scores comparable, fuzzy scores still play an important role in this unification. In any case, these efforts may help to provide a better integration of translation technologies and spread best practices for all parties involved in a translation project.

## 5 Conclusion and future work

In this paper, we reported an experiment where ten professional translators with diverse experience in translation and MTPE complete the same translation and post-editing job within their everyday work environment using files from a real translation request. The objective was to analyze the relation between productivity and automated metrics in a commercial setting, and verify if the target-side FMS could be an adequate method of evaluating MT output for PE.
The more than 7,000 words of the file to be translated from English into Spanish was analyzed with the usual CAT tools in our company to ensure it contained a significant amount of fuzzy matches, exact matches and no-match segments. Half of the no-match segments was randomly selected for MTPE, while the other half was translated from scratch. The MT output for the MTPE sample was generated using one of our customized RBMT engines.

The results show that the FMS applied to MTPE evaluation has a strong correlation with productivity, comparable to the correlation obtained with more traditional evaluation metrics such as BLEU and TER. Moreover, the FMS was the only metric able to reflect the higher productivity obtained by the MTPE sample over the 75–84% TM match band. Another advantage shown is that it can also be used to set the TM matching threshold at which post-editing MT is more productive than post-editing TM matches.

Another finding is that MT output from a mature engine increases translators’ productivity when compared to translation from scratch and can even surpass the throughput of TM matches. Only one of the ten translators involved did not experience any productivity gain. The productivity gain achieved by the rest of translators varied greatly between them, with the most experienced translator having the least productivity increase and the least experienced translator obtaining the biggest gain. However, this trend cannot be confirmed by the performance of the rest of translators.

It was also revealed that inline tags have a big impact on productivity, a fact which is not reflected in any of the known metrics and which has not yet received much attention in research. Other areas for future work include using this data set to perform a study on quality estimation to avoid depending on reference translations and further analysis of automated metrics in order to reveal the threshold at which each metric ceases to consider a segment useful in terms of productivity.

The performance of FMS applied to MTPE evaluation has been shown to be as solid as any of the traditional metrics. Its correlation with productivity is as strong as BLEU or TER and it can be used to determine thresholds between TM matching and MTPE segments. Moreover, FMS is easier to calculate for general users and we believe it would be more readily embraced by the industry due to its analogy with the well-established TM match bands.

References


A Distributed Inflection Model for Translating into Morphologically Rich Languages

Ke Tran  m.k.tran@uva.nl
Arianna Bisazza  a.bisazza@uva.nl
Christof Monz  c.monz@uva.nl
Informatics Institute, University of Amsterdam,
Science Park 904, 1098 XH Amsterdam, The Netherlands

Abstract
Lexical sparsity is a major challenge for machine translation into morphologically rich languages. We address this problem by modeling sequences of fine-grained morphological tags in a bilingual context. To overcome the issue of ambiguous word analyses, we introduce soft tags, which are under-specified representations retaining all possible morphological attributes of a word. In order to learn distributed representations for the soft tags and their interactions we adopt a neural network approach. This approach allows for the combination of source and target side information to model a wide range of inflection phenomena. Our re-inflection experiments show a substantial increase in accuracy compared to a model trained on morphologically disambiguated data. Integrated into an SMT decoder and evaluated for English-Italian and English-Russian translation, our model yields improvements of up to 1.0 BLEU over a competitive baseline.

1 Introduction
In morphologically rich languages (MRLs), words can have many different surface forms depending on the grammatical context. When translating into MRLs, standard statistical machine translation (SMT) models such as phrase translation models and n-gram language models (LMs) often fail to select the right surface form due to the sparsity of observed word sequences (Minkov et al., 2007; Green and DeNero, 2012). While neural LMs (Bengio et al., 2003; Schwenk, 2007) address lexical sparsity to a certain degree by projecting word sequences to distributed vector representations, they still suffer from the problem of rare words which is particularly exacerbated in MRLs (Botha and Blunsom, 2014; Jean et al., 2015; Luong et al., 2015).

A potential solution to overcome data sparsity in MRLs, is to use word representations that separate the grammatical aspects of a word, i.e. inflection, from the lexical ones. Such word representations already exist for many languages in the form of morphological analyzers or lexicons. However, using these resources for statistical language modeling is far from trivial due to the issue of ambiguous word analyses. Table 1 illustrates this problem in Italian, for which a fine-grained morphological lexicon but no sizable disambiguated corpus exists. These morphological analyses1 clearly contain information that is useful to encourage grammatical agreement and, in this case, detect the highlighted error. Unfortunately, though, the needed

1In this work we use the terms analysis and tag interchangeably to denote fine-grained word annotations provided by a morphological analyzer or lexicon.
Table 1: Example of morphological error in Italian SMT output: the verb form should be plural (circolano) and not singular (circola) to agree in number with the subject. Most of the words have multiple analyses according to our morphological lexicon of reference (Zanchetta and Baroni, 2005). The correct one in context is highlighted.

information is difficult to access because each word can have multiple analyses. Performing contextual disambiguation during translation is an ill-posed problem because the SMT decoder produces large numbers of ungrammatical word sequences but gold tagged training data is naturally composed of grammatical sentences. Moreover, searching for the optimal tag sequence introduces spurious ambiguity into the SMT decoder. Finally, training a disambiguator requires manually disambiguated data, which is not available in many languages and costly to produce.

In this paper, we address this problem with a novel inflection model that is based on two main ideas: First, morphological ambiguity does not need to be resolved for SMT. Instead, we map words to a space where all possible morphological attributes of a word are retained. Rather than enforcing hard tagging decisions, we let the model operate on soft word representations. The resulting tag set is larger than the original one, but still effective at reducing the lexical sparsity of purely word-based LM. Second, learning distributed representations for soft morphological tags can help share statistical strength among overlapping tags, i.e. tags that have some attributes in common. To achieve this, we train a neural network that predicts sequences of soft tags conditioned on rich contextual features.

We show that: (i) our soft representation model achieves higher accuracies in re-inflecting translations than a model performing contextual disambiguation, and (ii) our model significantly improves translation quality on two different target MRLs, including a language for which no sizable disambiguated corpora exist.

The paper is organized as follows: after reviewing the previous work (Section 2), we present our distributed inflection model based on soft morphological representations (Section 3). In Section 4 we introduce the general experimental setup, followed by a detailed description of the re-inflection experiments (Section 5) and the end-to-end SMT experiments (Section 6). We conclude with a discussion of SMT output examples and an outlook of future work.

2 Previous Work

Previous work on inflection modeling for translation into MRLs has mostly relied on the availability of morphologically disambiguated data to choose the most probable analysis of each word in either a context-independent (Minkov et al., 2007) or context-dependent (Green and DeNero, 2012; Koehn and Hoang, 2007; Subotin, 2011) way. While the former irrevocably discards potentially useful attributes of the words, the latter tasks the inflection model with disambiguating the word sequence under construction, which is difficult given the ill-formedness of SMT output and a cause of spurious ambiguity.

---

2 This issue has also been shown to affect syntactic parsing of SMT output (Post and Gildea, 2008; Carter and Monz, 2011).
Considerably less work has focused on MRLs where disambiguated data does not exist, with few exceptions where ambiguity is solved by randomly selecting one analysis per word type (Minkov et al., 2007; Toutanova et al., 2008; Jeong et al., 2010).

As for how inflection models are integrated into the STM system, different strategies have been proposed. Minkov et al. (2007); Toutanova et al. (2008); Fraser et al. (2012) treat inflection as a post-processing task: the SMT model is trained to produce lemmatized target sentences (possibly enhanced with some form of morphological annotation) and afterwards the best surface form for each lemma is chosen by separate inflection models. Some work has focused on the generation of new inflected phrases given the input sentence (Chahuneau et al., 2013) or given the bilingual context during decoding (Koehn and Hoang, 2007; Subotin, 2011). Other inflection models have been integrated to SMT as additional feature functions: e.g. as an additional lexical translation score (Jeong et al., 2010; Tran et al., 2014) or as an additional target language model score (Green and DeNero, 2012). We follow this last strategy, rather than generating new inflections, motivated by previous observations that, when translating into MRLs, a large number of reference inflections are already available in the SMT models but are not selected for Viterbi translation (Green and DeNero, 2012; Tran et al., 2014).

More in general, our work is related to class-based language modeling (Brown et al., 1992) with the major difference that we also condition on source-side context and that we use explicit morphological representations instead of data-driven word clusters (Uszkoreit and Brants, 2008), word suffixes (Müller et al., 2012; Bisazza and Monz, 2014) or coarse-grained part-of-speech tags (Koehn et al., 2008).

Modeling morphology using neural networks has recently shown promising results: in the context of monolingual neural language modeling, Luong et al. (2013); Botha and Blunsom (2014) obtain the vectorial representation of a word by composing the representations of its morphemes. Tran et al. (2014) model translation stem and suffix selection in SMT with a bilingual neural network. Soricut and Och (2015) discover morphological transformation rules from word embeddings learned by a shallow network. We are not aware of work that leveraged fine-grained morphological tags for neural language or translation modeling.

3 A Distributed Inflection Model

In MRLs, the surface form of a word is heavily determined by its grammatical features, such as number, case, tense etc. Choosing the right target word form during translation is a complex problem since some of these features depend on the source context while others depend on the target context (agreement phenomena). We model target language inflection by a Markov process generating a sequence of abstract word representations based on source and target context. This complements previous work focusing on either the former (Avramidis and Koehn, 2008; Chahuneau et al., 2013; Tran et al., 2014) or the latter (Green and DeNero, 2012; Fraser et al., 2012; Botha and Blunsom, 2014; Bisazza and Monz, 2014).

3.1 Soft Morphological Representations

As previously stated, it is common for words in MRLs to admit multiple morphological analyses out of context. Rather than trying to disambiguate the analyses in context using for instance conditional random fields (Green and DeNero, 2012; Fraser et al., 2012), we modify the tagging scheme so that each word corresponds to only one tag. To also avoid the loss of useful information incurred when arbitrarily selecting one analysis per word type (Minkov et al., 2007; Jeong et al., 2010), we introduce soft morphological representations, or simply soft tags.

Assume that a morphological analysis $\mu$ is a set of morphological attributes $S(\mu)$ such as masculine or plural. Given a word $w$, a morphological analyzer or lexicon LEX returns a list
Our inflection model$^3$, Inf-NN, is trained on word-aligned bilingual data to predict sequences of target soft tags given a fixed-size target history and the input source sentence (see Figure 1). We adopt a neural LM approach as learning distributed representations for the soft tags can help to share statistical information among overlapping tags (i.e. tags that share some morphological attributes). Moreover, compared to Maximum Entropy models that use lexical features, neural networks can better exploit sparse input features such as lexicalized source context and target lemma features, as well as their interactions, in high dimensional spaces.

We learn distributed representations for both source words and target soft tags. The source word representations are initialized from pre-trained embeddings, which has been shown to encode certain morphological regularities (Soricut and Och, 2015), whereas target tag representations are initialized randomly.

Inf-NN is a feed-forward neural network whose output is a conditional probability distribution over a set of morphological tags given target history and source context. Formally, let $h_i = (r_{i-1}, \ldots, r_{i-n+1})$ be the $n-1$ tag history of the target word $w_j$, and $c_j = (s_{j-k}, \ldots, s_{j+k})$ the source context centering at the word $s_j$ aligned to $w_j$ by an automatic aligner. We use simple heuristics similar to the approach by Devlin et al. (2014) to handle null and multiple alignments so that each target word $w_i$ can be mapped to exactly one source word $s_j$. Let $s_j \in \mathbb{R}^D$ and $r_i \in \mathbb{R}^D$ denote the distributed representations of source $s_j$ and target tag $r_i$ respectively. Then, the conditional probability $p_{\text{Inf-NN}}(r_i|h_i, c_j)$ is computed at the output layer $y$ of the network as follows:

$$z_i = \phi(W^c c_j + W^h h_i + b_z)$$

$$y = \text{softmax}(W^y z_i + b_y)$$

$^3$The implementation is available at https://bitbucket.org/ketran/soft-tags
Figure 1: Graphical representation of the Inf-NN model: the current target word’s soft tag, \( r_i \), is predicted based on a fixed-size target tag history and a source side context centered around \( s_j \), the translation of \( w_i \). Each target word \( w_i \) can be deterministically mapped to a soft tag \( r_i \).

where \( W^c \), \( W^h \), and \( W^m \) are weight matrices, \( c_j \) and \( h_i \) are shorthands for \([s_{j-k}; \ldots; s_{j+k}]\) and \([r_{i-1}; \ldots; r_{i-n+1}]\) respectively, \([v; v']\) denotes vector concatenation, and \( \phi \) is a non-linear transfer prelu. As \( \phi \), we use in all experiments the channel-shared parametric rectified linear unit (PReLU) introduced by He et al. (2015). PReLU \( \phi(x) \) is defined as:

\[
\phi(x) = \begin{cases} 
  x & \text{if } x > 0 \\
  ax & \text{otherwise}
\end{cases}
\]

where \( a \) is a parameter learned during training. To speed up decoding, we train the Inf-NN model with a self-normalized objective (Devlin et al., 2014; Andreas and Klein, 2015). More specifically, we adopt the objective function proposed by Andreas and Klein (2015):

\[
\ell(\theta) = -\mathbb{E} \left[ \ln p(r_i|h_i, c_j) \right] + \eta \|\theta\|^2_2 + \frac{\gamma}{p} \mathbb{E} \left[ \ln^2 Z(h_i, c_j) | (h_i, c_j) \in \mathcal{H} \right]
\]

where \( \mathcal{H} \) is a set of random samples on which self-normalization is performed, \( \theta = \{\{s_j\}, \{r_i\}, W^c, W^h, W^m, b_z, a\} \) are the parameters of the networks, and \( Z(h_i, c_j) \) is the partition function of the input \((h_i, c_j)\). In practice, we obtain \( \mathcal{H} \) by sampling from a Bernoulli distribution \( \text{Bern}(p) \). This is equivalent to applying dropout (Srivastava et al., 2014) on the loss gradient \( \ell_2 \) norm.

4 Experimental Setup

We evaluate our approach on two related tasks: re-inflecting reference translations and end-to-end translation from English into MRLs. With the first task, we test the effectiveness of soft morphological representations against (i) a model that randomly assigns one tag per word type (among its possible tags) and (ii) a model that admits multiple tags per word and requires a pre-disambiguated corpus to be trained. With the second task, we measure translation quality when our inflection model is integrated into a state-of-the-art phrase-based SMT decoder, showing its applicability to languages where no disambiguated data exists.
4.1 Data

As target languages, we choose two MRLs belonging to different language families and displaying different inflectional patterns: Russian has very rich nominal, adjectival and verbal inflection, while Italian has moderate nominal and adjectival inflection, but extremely rich verbal inflection. Experiments are performed on the following tasks:

- English-Russian WMT (Bojar et al., 2013): translation of news commentaries with large-scale training data.
- English-Italian IWSLT (Cettolo et al., 2014): translation of speeches with either small-scale training data (TED talks only) or large-scale training data (TED talks and European proceedings).

SMT training data statistics are reported in Table 2. The Russian Inf-NN model is trained on a 1M-sentence subset of the bilingual data, while the Italian one is trained on all the data available in each setting. For each data set, we create automatic word alignments using GIZA++ (Och and Ney, 2003).

<table>
<thead>
<tr>
<th></th>
<th>En-Ru</th>
<th>En-It</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>large</td>
<td>small</td>
</tr>
<tr>
<td>Bilingual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#sentences</td>
<td>2.4M</td>
<td>180K</td>
</tr>
<tr>
<td>src/trg #tokens</td>
<td>49.2/47.2M</td>
<td>3.6/3.4M</td>
</tr>
<tr>
<td>src/trg dict.size</td>
<td>774K/1100K</td>
<td>55K/80K</td>
</tr>
<tr>
<td>Monoling.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#sentences</td>
<td>21.0M</td>
<td>2.1M</td>
</tr>
<tr>
<td>trg #tokens</td>
<td>390M</td>
<td>58.4M</td>
</tr>
<tr>
<td>src/trg dict.size</td>
<td>2.7M</td>
<td>199K</td>
</tr>
</tbody>
</table>

Table 2: Training corpora statistics.

The ambiguous morphological analyses are obtained from the Russian OpenCorpora lexicon⁴ (Bocharov et al., 2013) and from the Italian Morph-it!⁵ lexicon (Zanchetta and Baroni, 2005). Table 3 shows the number of tags and soft tags occurring in our training data, as well as the expected counts of analyses per word $E_w[t]$, words per lemma $E_t[w]$ and analyses per lemma $E_l[t]$.

<table>
<thead>
<tr>
<th>Language</th>
<th>#tags</th>
<th>#soft-tags</th>
<th>$E_w[t]$</th>
<th>$E_t[w]$</th>
<th>$E_l[t]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Russian</td>
<td>892</td>
<td>4431</td>
<td>3.8</td>
<td>7.2</td>
<td>27.4</td>
</tr>
<tr>
<td>Italian</td>
<td>450</td>
<td>901</td>
<td>1.9</td>
<td>12.7</td>
<td>24.3</td>
</tr>
</tbody>
</table>

Table 3: Morphological characteristics of the Inf-NN training data: number of tags and soft tags, expected counts of analyses per word $E_w[t]$, words per lemma $E_t[w]$ and analyses per lemma $E_l[t]$.

We find that the Russian tag set and, consequently, the soft tag set are considerably larger than the Italian ones. The average morphological ambiguity is also larger in Russian (3.8 versus 1.9 per lemma) and increases to 27.4 in the case of analyses per word.

⁴opencorpora.org
⁵sslmitdev-online.sslmit.unibo.it/linguistics/morph-it.php
1.9 tags per word). However, somewhat surprisingly, morphological richness is higher in Italian (12.7 versus 7.2 words per lemma). At a closer inspection, we find that most of this richness is due to verbal inflection which goes up to 50 forms for frequently observed verbs.

### 4.2 Neural network training

The Inf-NN models are trained on a history of 4 target tags and source context of 7 words with the following configuration: Embedding size is set to 200 and the number of hidden units to 768. Target word and soft-tag embeddings are initialized randomly from a Gaussian distribution with mean zero and standard deviation 0.01. Source word embeddings are initialized from pre-trained Glove vectors (Pennington et al., 2014) and rescaled by a factor of 0.1. Weight matrices of linear layers are initialized from a zero-mean Gaussian distribution with standard deviation \( \sqrt{2/n_i} \) where \( n_i \) is the number of input units (He et al., 2015). We set self-normalization strength \( \gamma = 0.02 \), Bernoulli parameter \( p = 0.1 \), and regularization parameter \( \eta = 10^{-4} \). All models are trained with a mini-batch size of 128 for 30 epochs. Our stochastic objective functions are optimized using the first-order gradient-based optimizer Adam (Kingma and Ba, 2015). We use the default settings suggested by the authors: \( \alpha = 0.001 \), \( \beta_1 = 0.9 \), \( \beta_2 = 0.999 \), \( \epsilon = 10^{-8} \) and \( \lambda = 1 - 10^{-8} \).

### 5 Re-inflection Experiments

The purpose of this experiment is to simulate the behavior of the inflection model during SMT decoding: Given a reference translation and its corresponding source sentence, we re-inflect the former using a simple beam search and count how many times the model recovers the correct surface word form on a 10K-sentence held-out data set.

Since we do not assume the availability of a disambiguator, we also have to deal with lemma ambiguity. While this issue does not affect the definition and training of our Inf-NN, we do need lemmas to determine the set of candidate surface forms \( I_w \) for each word \( w \) that is being re-inflected. As a solution, we define \( I_w \) as the union of the surface forms of each possible lemma of \( w \) or, more formally, as:

\[
I_w = \{ w_i \mid \text{lem}(w_i) \cap \text{lem}(w) \neq \emptyset \}
\]

where \( \text{lem}(w) \) denotes the set of lemmas returned by the lexicon for word \( w \). For example, the Italian form \( baci \) has two possible lemmas: \( bacio \) (noun: kiss) and \( baciare \) (verb: to kiss). Its candidate set \( I_w \) will then include all the forms of the noun \( bacio \) and all the forms of the verb \( baciare \): that is, \( bacio, baci, baciamo, baciate, baciano, \) etc.

We compare the proposed soft-tag Inf-NN against an Inf-NN trained on randomly assigned tag per type and to another one trained on tag sequences disambiguated by TreeTagger (Schmid, 1994; Sharoff et al., 2008). The latter model must search through a much larger space of morphological tag sequences. Therefore, to allow for a fair comparison, we set a higher beam size when re-inflecting with this model. As another difference from the other models, the TreeTagger-based inflection model relies on the lemmatization performed by TreeTagger to define the candidate set \( I_w \).

To validate the effectiveness of the neural network approach, we also compare Inf-NN to a simpler MaxEnt model trained on a similar configuration. Finally, we evaluate the importance of source-side context features by experimenting with a series of Inf-NN models that are only conditioned on the target tag history.

Since no morphological disambiguator is available for Italian, we perform this experiment only for Russian. As shown in Table 4, soft tags perform best in all settings and become even more effective when moving from MaxEnt to neural network, demonstrating the impor-
Table 4: Token-level re-inflection accuracy (%) on a 10K-sentence English-Russian held-out set. The last column indicates the beam size used when searching for the optimal re-inflected sequence.

<table>
<thead>
<tr>
<th>Method</th>
<th>MaxEnt with src</th>
<th>MaxEnt w/o src</th>
<th>Inf-NN with src</th>
<th>beam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree-Tagger: all analyses</td>
<td>56.33</td>
<td>61.19</td>
<td>69.68</td>
<td>200</td>
</tr>
<tr>
<td>Random: 1 analysis per word</td>
<td>66.08</td>
<td>72.32</td>
<td>79.92</td>
<td>5</td>
</tr>
<tr>
<td>Soft-Reps: 1 soft tag per word</td>
<td><strong>66.95</strong></td>
<td><strong>75.43</strong></td>
<td><strong>81.93</strong></td>
<td>5</td>
</tr>
</tbody>
</table>

tance of learning distributed representations for the soft tags. The notably lower accuracy of the TreeTagger-based model confirms our intuition that morphological disambiguation is not needed to model inflection in SMT, but can actually make the task more difficult. This result can be explained by the fact that, when fixing one tag per word type either by random assignment or with soft tags, the number of tags per lemma becomes substantially smaller (cf. Table 3) and classification easier. On the other hand, the Tree-Tagger based model operating on all word analyses has to deal with spurious ambiguity: that is, a correct sequence of inflected words can correspond to multiple tag sequences that are competing with one another. Solving this problem by marginalizing over the ambiguous analyses (cf. Equation 1) can lead to intractable decoding (Sima’an, 1996; Li et al., 2009).

The model using soft-tags, which capture all possible morphological attributes of words, performs the best. Even without using source context features, our Inf-NN outperforms the MaxEnt model by 8.5% absolute because of the high dimensional space used to capture complex morphological regularities. By adding source context, we further increase accuracy by 6.5%, leading to an overall gain of 15% over the MaxEnt baseline.

Next, we investigate the impact of our most accurate re-inflection model (Soft-Reps Inf-NN) in an end-to-end SMT setting without relying on any disambiguated data.

6 End-to-end SMT Experiments

We integrated our Inf-NN model into a phrase-based SMT decoder similar to Moses (Koehn et al., 2007) as an additional log-probability feature function ($\log p_{\text{Inf-NN}}$).

When a new target phrase $\bar{w}$ is produced by the decoder, the Inf-NN model returns a probability for each word $w_i$ that composes it, given the previously translated words’ soft tags and the source context centered around the source word $s_j$ aligned to $w_i$. To detect $s_j$ we store phrase-internal word alignments in the phrase table and use simple heuristics to map each target index $i$ to exactly one source index $j$, as done for the Inf-NN training (Section 3.2). Since every target word corresponds to one soft tag, obtaining the representation of $w_i$ is trivial (by lookup in a word-tag map) and so is maintaining the target tag history. This crucially differs from previous approaches that distinguish between hypotheses with equal surface forms but different morphological analyses (Koehn et al., 2007), thereby introducing spurious ambiguity into what is already a huge search space. As a result, the integration of our Inf-NN does not affect decoding speed.

6 Green and DeNero (2012) also tag each target phrase in context as it is produced. However, they avoid the spurious ambiguity problem by only preserving the most probable tag sequence for each phrase (incremental greedy decoding).
6.1 Baseline

Our SMT baseline is a competitive phrase-based SMT system including hierarchical lexicalized reordering models (Galley and Manning, 2008) and a 5-gram target LM trained with modified Kneser-Ney smoothing (Chen and Goodman, 1999). Since the large English-Italian data comes from very different sources (TED talks and European proceedings), we construct phrase table and reordering models for this experiment using the fillup technique (Bisazza et al., 2011). Note that our baseline does not include previously proposed inflection models because the main goal of our experiment is to demonstrate the effectiveness of the proposed approach for languages where no sizable disambiguated data exists, which is indeed the case for Italian.

Feature weights are tuned with pairwise ranking optimization (Hopkins and May, 2011) on the union of IWSLT’s dev10 and test10 in Italian, and on the first 2000 lines of wmt12 benchmark in Russian (Callison-Burch et al., 2012). During tuning, 14 PRO parameter estimation runs are performed in parallel on different samples of the n-best list after each decoder iteration. The weights of the individual PRO runs are then averaged and passed on to the next decoding iteration. Performing weight estimation independently for a number of samples corrects for some of the instability that can be caused by individual samples.

6.2 Results

Translation quality is measured by case-insensitive BLEU (Papineni et al., 2002) on IWSLT’s test12 and test14 in Italian, and on wmt13 and wmt14 for Russian, all provided with one reference translation. To see whether the differences between the approaches we compared in our experiments are statistically significant, we apply approximate randomization (Noreen, 1989).

<table>
<thead>
<tr>
<th>Data</th>
<th>Test</th>
<th>Baseline</th>
<th>Inf-NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>en→ru</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>large</td>
<td>wmt13</td>
<td>19.0</td>
<td>19.3(^*)(+0.3)</td>
</tr>
<tr>
<td></td>
<td>wmt14</td>
<td>26.1</td>
<td>26.7(^*)(+0.6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>small</td>
<td>iwslt12</td>
<td>24.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25.6(^*)(+1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iwslt14</td>
<td>20.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20.9(^*)(+0.5)</td>
</tr>
<tr>
<td>en→it</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>large</td>
<td>iwslt12</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>25.8(^*)(+0.8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iwslt14</td>
<td>20.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>21.4(^*)(+0.5)</td>
</tr>
</tbody>
</table>

Table 5: Impact on translation quality of the Inf-NN model. \(^*\) marks significance level \(p < .01\).

Results are presented in Table 5. Our Inf-NN model consistently leads to significant improvements over a competitive baseline, for both language pairs and all test sets, without affecting decoding speed. By comparing the two data conditions in English-Italian, we see that most of the BLEU gain is preserved even after adding a large amount of parallel training data. This suggests that morphological phenomena are not sufficiently captured by phrases and stresses the importance of specifically modeling word inflection. It is possible that adding even more training data would reduce the impact of our inflection model, but currently we do not have access to other data sets that would be relevant to our translation tasks.

To put these results into perspective, our improvements are comparable to those achieved by previous work that generated new phrase inflections using a morphological disambiguator (Chahuneau et al., 2013) on the same large-scale English-Russian task.

\(^3\)Riezler and Maxwell (2005) have shown that approximate randomization is less sensitive to Type-I errors, i.e. less likely to falsely reject the null hypothesis, than bootstrap resampling (Koehn, 2004) in the context of SMT.
Table 6: Examples of SMT output drawn from IWSLT English-Italian test12 showing the effect of our inflection model on lexical selection.

6.3 Examples

As previously mentioned, most previous approaches to inflection modeling for SMT may not be applied to Italian due to the lack of morphological disambiguated data. It is then particularly interesting to analyze how our model affects baseline translations. Table 6 presents a number of English-Italian SMT output examples where the use of our soft-tag Inf-NN either resulted in a better inflection choice (1-3) or not (4-5). Out of the ‘good’ examples, only (1) resulted in a complete match with the reference translation, while in (2) and (3) the system preferred an equally appropriate lexical choice, showing that automatically evaluating inflection models in an SMT setting is far from trivial.

The usefulness of source-side features is demonstrated by example (3): here, the translation of broken should agree in gender with the subject he but the baseline system chose instead a feminine form (infranta). Since the subject pronoun can be dropped in Italian, this error cannot be detected by the target language model and may only be fixed by translating the sequence ‘he died broken’ as a single phrase, which was never observed in the training data. By contrast, Inf-NN successfully exploited the source-side context and preferred a masculine form (devastato).

Next are two unsuccessful examples: in (4) Inf-NN encouraged the system to translate the whole phrase ‘the classic asian student’ as masculine whereas the baseline translation used...
an incoherent mix of masculine and feminine. Unfortunately, though, the student in question, i.e., the speaker, happened to be a woman, but this could not be inferred in any way from this sentence. In (5) Inf-NN failed to fix the agreement between adjective and subject pronoun. By inspecting the parallel data we found that the word *enmeshed* always occurred with plural forms of Italian adjectives. This example shows that improving the scoring of the existing translation options is not always sufficient. While we do not address generation of new inflected forms in this work, this is an interesting direction for future work.

7 Conclusions

We have proposed a novel morphological representation scheme combined with a neural network for modeling translation into morphologically rich languages (MRLs). Our approach successfully circumvents the problem of ambiguous word analyses and makes it possible to improve translation into MRLs where morphological lexica but no manually disambiguated corpora exist.

Evaluated in a re-inflection task, the proposed soft tags achieve significantly higher accuracy than (i) a model using standard tags and trained on morphologically disambiguated data and (ii) a Maximum Entropy model that does not learn distributed representations for source words and target tags. When integrated into a state-of-the-art SMT decoder, our inflection model significantly improves translation quality in two different language pairs, without having to disambiguate during decoding. In particular, our positive English-Italian results under both small- and large-scale data conditions demonstrate the applicability of our approach to languages where no disambiguator exists.

As future work, we will consider learning distributed morphology representation directly from the corpus jointly with the inflection model as well as generating unseen word inflections during translation.

Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers 639.022.213 and 612.001.218.

References


Bandit Structured Prediction for Learning from Partial Feedback in Statistical Machine Translation

Artem Sokolov  sokolov@cl.uni-heidelberg.de
Stefan Riezler*  riezler@cl.uni-heidelberg.de
Computational Linguistics and IWR*
Heidelberg University, 69120 Heidelberg, Germany
Tanguy Urvoy  tanguy.urvoy@orange.com
Orange Labs, 2 Avenue Pierre Marzin, 22307 Lannion, France

Abstract
We present an approach to structured prediction from bandit feedback, called Bandit Structured Prediction, where only the value of a task loss function at a single predicted point, instead of a correct structure, is observed in learning. We present an application to discriminative reranking in Statistical Machine Translation (SMT) where the learning algorithm only has access to a \(1 - \text{BLEU}\) loss evaluation of a predicted translation instead of obtaining a gold standard reference translation. In our experiment bandit feedback is obtained by evaluating BLEU on reference translations without revealing them to the algorithm. This can be thought of as a simulation of interactive machine translation where an SMT system is personalized by a user who provides single point feedback to predicted translations. Our experiments show that our approach improves translation quality and is comparable to approaches that employ more informative feedback in learning.

1 Introduction
Learning from bandit\(^1\) feedback describes an online learning scenario, where on each of a sequence of rounds, a learning algorithm makes a prediction, and receives partial information in terms of feedback to a single predicted point. In difference to the full information supervised scenario, the learner does not know what the correct prediction looks like, nor what would have happened if it had predicted differently. This scenario has (financially) important real world applications such as online advertising (Chapelle et al., 2014) that showcases a tradeoff between exploration (a new ad needs to be displayed in order to learn its click-through rate) and exploitation (displaying the ad with the current best estimate is better in the short term). Crucially, in this scenario it is unrealistic to expect more detailed feedback than a user click on the displayed ad. Similar to the online advertising scenario, there are many potential applications of bandit learning to NLP situations where feedback is limited for various reasons. For example, online learning has been applied successfully in interactive statistical machine translation (SMT) (Bertoldi et al., 2014; Denkowski et al., 2014; Green et al., 2014). Post-editing feedback clearly is limited by its high cost and by the required expertise of users, however, current approaches force the full information supervised scenario onto the problem of learning from user post-edits.

\(^1\)The name is inherited from a model where in each round a gambler pulls an arm of a different slot machine (“one-armed bandit”), with the goal of maximizing his reward relative to the maximal possible reward, without apriori knowledge of the optimal slot machine.
Bandit learning would allow to learn from partial user feedback that is easier and faster to obtain than full information. An example where user feedback is limited by a time constraint is simultaneous translation of a speech input stream (Cho et al., 2013). Clearly, it is unrealistic to expect user feedback that goes beyond a one-shot user quality estimate of the predicted translation in this scenario. Another example is SMT domain adaptation where the translations of a large out-of-domain model are re-ranked based on bandit feedback on in-domain data. This can also be seen as a simulation of personalized machine translation where a given large SMT system is adapted to a user solely by single-point user feedback to predicted structures.

The goal of this paper is to develop algorithms for structured prediction from bandit feedback, tailored to NLP problems. We investigate possibilities to “banditize” objectives such as expected loss (Och, 2003; Smith and Eisner, 2006; Gimpel and Smith, 2010) that have been proposed for structured prediction in NLP. Since most current approaches to bandit optimization rely on a multiclass classification scenario, the first challenge of our work is to adapt bandit learning to structured prediction over exponentially large structured output spaces (Taskar et al., 2004; Tsochanaridis et al., 2005). Furthermore, most theoretical work on online learning with bandit feedback relies on convexity assumptions about objective functions, both in the non-stochastic adversarial setting (Flaxman et al., 2005; Shalev-Shwartz, 2012) as well as in the stochastic optimization framework (Spall, 2003; Nemirovski et al., 2009; Bach and Moulines, 2011). Our case is a non-convex optimization problem, which we analyze in the simple and elegant framework of pseudogradient adaptation that allows us to show convergence of the presented algorithm (Polyak and Tsypkin, 1973; Polyak, 1987).

The central contributions of this paper are:

- An algorithm for minimization of expected loss for structured prediction from bandit feedback, called Bandit Structured Prediction.
- An analysis of convergence of our algorithm in the stochastic optimization framework of pseudogradient adaptation.
- An experimental evaluation on structured learning in SMT. Our experiment follows a simulation design that is standard in bandit learning, namely by simulating bandit feedback by evaluating task loss functions against gold standard structures without revealing them to the learning algorithm.

As a disclaimer, we would like to note that improvements over traditional full-information structured prediction cannot be expected from learning from partial feedback. Instead, the goal is to investigate learning situations in which full information is not available. Similarly, a comparison between our approach and dueling bandits (Yue and Joachims, 2009) is skewed towards the latter approach that has access to two-point feedback instead of one-point feedback as in our case. While it has been shown that querying the loss function at two points leads to convergence results that closely resemble bounds for the full information case (Agarwal et al., 2010), such feedback is clearly twice as expensive and, depending on the application, might not be elicitable from users.

2 Related Work

Stochastic Approximation. Online learning from bandit feedback dates back to Robbins (1952) who formulated the task as a problem of sequential decision making. His analysis, as ours, is in a stochastic setting, i.e., certain assumptions are made on the probability distribution of feedback and its noisy realization. Stochastic approximation covers bandit feedback as noisy observations which only allow to compute noisy gradients that equal true gradients in expectation. While the stochastic approximation framework is quite general, most theoretical analyses....
of convergence and convergence rate are based on (strong) convexity assumptions (Polyak and Juditsky, 1992; Spall, 2003; Nemirovski et al., 2009; Bach and Moulines, 2011, 2013) and thus not applicable to our case.

Non-Stochastic Bandits. Auer et al. (2002) initiated an active area of research on non-stochastic bandit learning, i.e., no statistical assumptions are made on the distribution of feedback, including models of feedback as a malicious choice of an adaptive adversary. The adversarial bandit setting has been extended to take context or side information into account, using models based on general linear classifiers (Auer et al., 2002; Langford and Zhang, 2007; Chu et al., 2011). However, they formalize a multi-class classification problem that is not easily scalable to general exponentially large structured output spaces. Furthermore, most theoretical analyses rely on online (strongly) convex optimization (Flaxman et al., 2005; Shalev-Shwartz, 2012) thus limiting the applicability to our case.

Neurodynamic Programming. Bertsekas and Tsitsiklis (1996) cover optimization for neural networks and reinforcement learning under the name of “neurodynamic programming”. Both areas are dealing with non-convex objectives that lead to stochastic iterative algorithms. Interestingly, the available analyses of non-convex optimization for neural networks and reinforcement learning in Bertsekas and Tsitsiklis (1996), Sutton et al. (2000), or Bottou (2004) rely heavily on Polyak and Tsypkin (1973)’s pseudogradient framework. We apply their simple and elegant framework directly to give asymptotic guarantees for our algorithm.

NLP Applications. In the area of NLP, recently algorithms for response-based learning have been proposed to alleviate the supervision problem by extracting supervision signals from task-based feedback to system predictions. For example, Goldwasser and Roth (2013) presented an online structured learning algorithm that uses positive executability of a semantic parse against a database to convert a predicted parse into a gold standard structure for learning. Riezler et al. (2014) apply a similar idea to SMT by using the executability of a semantic parse of a translated database query as signal to convert a predicted translation into gold standard reference in structured learning. Sokolov et al. (2015) present a coactive learning approach to structured learning in SMT where instead of a gold standard reference a slight improvement over the prediction is shown to be sufficient for learning. Saluja and Zhang (2014) present an incorporation of binary feedback into an latent structured SVM for discriminative SMT training. NLP applications based on reinforcement learning have been presented by Branavan et al. (2009) or Chang et al. (2015). Their model differs from ours in that it is structured as a sequence of states at which actions and rewards are computed, however, the theoretical foundation of both types of models can be traced back to Polyak and Tsypkin (1973)’s pseudogradient framework.

3 Expected Loss Minimization under Full Information

The expected loss learning criterion for structured prediction is defined as a minimization of the expectation of a task loss function with respect to the conditional distribution over structured outputs (Gimpel and Smith, 2010; Yuille and He, 2012). More formally, let \( \mathcal{X} \) be a structured input space, let \( \mathcal{Y}(x) \) be the set of possible output structures for input \( x \), and let \( \Delta_y : \mathcal{Y} \rightarrow [0,1] \) quantify the loss \( \Delta_y(y') \) suffered for making errors in predicting \( y' \) instead of \( y \); as a rule, \( \Delta_y(y') = 0 \) if \( y = y' \). Then, for a data distribution \( p(x,y) \), the learning criterion is defined as minimization of the expected loss

\[
\mathbb{E}_{p(x,y)\rho_w(y'|x)} [\Delta_y(y')] = \sum_{x,y} p(x,y) \sum_{y' \in \mathcal{Y}(x)} \Delta_y(y') \rho_w(y'|x).
\]
Assume further that output structures given inputs are distributed according to an underlying Gibbs distribution (a.k.a. conditional exponential or log-linear model)

\[ p_w(y|x) = \exp(w^\top \phi(x, y))/Z_w(x), \]

where \( \phi: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d \) is a joint feature representation of inputs and outputs, \( w \in \mathbb{R}^d \) is a weight vector, and \( Z_w(w) \) is a normalization constant.

The natural rule for prediction or inference is according to the minimum Bayes risk principle

\[ \hat{y}_w(x) = \arg \min_{y \in \mathcal{Y}(x)} \Delta_y(y)p_w(y|x). \]

(2)

This requires an evaluation of \( \Delta_y(y') \) over the full output space, which is standardly avoided in practice by performing inference according to a maximum a posteriori (MAP) criterion (which equals criterion (2) for the special case of \( \Delta_y(y') = 1[y \neq y'] \) where \( 1[s] \) evaluates to 1 if statement \( s \) is true, 0 otherwise)

\[ \hat{y}_w(x) = \arg \max_{y \in \mathcal{Y}(x)} p_w(y|x) \]

(3)

\[ = \arg \max_{y \in \mathcal{Y}(x)} w^\top \phi(x, y). \]

Furthermore, since it is unfeasible to take expectations over the full space \( \mathcal{X} \times \mathcal{Y} \) to perform minimization of objective (1), in the full information case the data distribution \( p(x, y) \) is approximated by the empirical distribution \( \hat{p}(x, y) = \frac{1}{T} \sum_{t=0}^{T} 1[x = x_t]1[y = y_t] \) for i.i.d. training data \( \{(x_t, y_t)\}_{t=0}^{T} \). This yields the objective

\[ \mathbb{E}_{\hat{p}(x,y)p_w(y'|x)}[\Delta_y(y')] = \frac{1}{T} \sum_{t=0}^{T} \sum_{y' \in \mathcal{Y}(x_t)} \Delta_{y_t}(y')p_w(y'|x_t). \]

(4)

While being continuous and differentiable, the expected loss criterion is typically non-convex. For example, in SMT, expected loss training for the standard task loss BLEU leads to highly non-convex optimization problems. Despite of this, most approaches rely on gradient-descent techniques for optimization (see Och (2003), Smith and Eisner (2006), He and Deng (2012), Auli et al. (2014), Wuebker et al. (2015), inter alia) by following the opposite direction of the gradient of (4):

\[ \nabla \mathbb{E}_{\hat{p}(x,y)p_w(y'|x)}[\Delta_y(y')] = \mathbb{E}_{\hat{p}(x,y)}\left[ \mathbb{E}_{p_w(y'|x)}[\Delta_y(y')\phi(x, y') - \mathbb{E}_{p_w(y'|x)}[\Delta_y(y')][\mathbb{E}_{p_w(y'|x)}[\phi(x, y')]] \right] \]

\[ = \mathbb{E}_{\hat{p}(x,y)p_w(y'|x)}\left[ \Delta_y(y') \phi(x, y') - \mathbb{E}_{p_w(y'|x)}[\phi(x, y')] \right]. \]

4 Bandit Structured Prediction

Bandit feedback in structured prediction means that the gold standard output structure \( y \), with respect to which the objective function is evaluated, is not revealed to the learner. Thus we can neither calculate the gradient of the objective function (4) nor evaluate the task loss \( \Delta \) as in the full information case. A solution to this problem is to pass the evaluation of the loss function to the user, i.e, we access the loss directly through user feedback without assuming existence of a fixed reference \( y \). We indicate this by dropping the subscript \( y \) in \( \Delta(y') \). Assuming a fixed,
unknown distribution \( p(x) \) over input structures, we can formalize the following new objective for expected loss minimization in a bandit setup

\[
J(w) = \mathbb{E}_{p(x)p_w(y'|x)} [\Delta(y')]
\]

(5)

\[
= \sum_x p(x) \sum_{y' \in Y(x)} \Delta(y') p_w(y'|x).
\]

Optimization of this objective is then as follows:

1. We assume a sequence of input structures \( x_t, t = 1, \ldots, T \) that are generated by a fixed, unknown distribution \( p(x) \).
2. We use a Gibbs distribution estimate as a sampling distribution to perform simultaneous exploration / exploitation on output structures (Abernethy and Rakhlin, 2009).
3. We use feedback to the sampled output structures to construct a parameter update rule that is an unbiased estimate of the true gradient of objective (5).

4.1 Algorithm

Algorithm 1 implements these ideas as follows: We assume as input a given learning rate schedule (line 1) and a deterministic initialization \( w_0 \) of the weight vector (line 2). For each random i.i.d. input structure \( x_t \), we calculate the expected feature count (line 5). This can be done exactly, provided the underlying graphical model permits a tractable calculation, or for intractable models, with MCMC sampling. We then sample an output structure \( \tilde{y}_t \) from the Gibbs model (line 6). If the number of output options is small, this is done by sampling from a multinomial distribution. Otherwise, we use a Perturb-and-MAP approach (Papandreou and Yuille, 2011), restricted to unary potentials, to obtain an approximate Gibbs sample without waiting for the MC chain to mix. Finally, an update in the negative direction of the instantaneous gradient, evaluated at the input structure \( x_t \) (line 8), is performed.

Intuitively, the algorithm compares the sampled feature vector to the average feature vector, and performs a step into the opposite direction of this difference, the more so the higher the loss of the sampled structure is. In the extreme case, if the sampled structure is correct (\( \Delta(\tilde{y}_t) = 0 \)), no update is performed.

4.2 Stochastic Approximation Analysis

The construction of the update in Algorithm 1 as a stochastic realization of the true gradient allows us to analyze the algorithm as a stochastic approximation algorithm. We show how our case can be fit in the pseudogradient adaptation framework of Polyak and Tsypkin (1973) which gives asymptotic guarantees for non-convex and convex objectives. They characterize an
iterative process
\[ w_{t+1} = w_t - \gamma_t s_t \] (6)
where \( \gamma_t \geq 0 \) is a learning rate, \( w_t \) and \( s_t \) are vectors in \( \mathbb{R}^d \) with fixed \( w_0 \), and the distribution of \( s_t \) depends on \( w_0, \ldots, w_t \). For a given lower bounded and differentiable function \( J(w) \) with Lipschitz continuous gradient \( \nabla J(w) \), that is, for all \( w, w' \), for all \( L \geq 0 \),
\[ \| \nabla J(w + w') - \nabla J(w) \| \leq L \| w' \| , \] (7)
the vector \( s_t \) in process (6) is said to be a pseudogradient of \( J(w) \) if
\[ \nabla J(w_t)^\top \mathbb{E}[s_t] \geq 0, \] (8)
where the expectation is taken over all sources of randomness. Intuitively, the pseudogradient \( s_t \) is on average at an acute angle with the true gradient, meaning that \( -s_t \) is on average a direction of decrease of the functional \( J(w) \).

In order to show convergence of the iterative process (6), besides conditions (7) and (8), only mild conditions on boundedness of the pseudogradient
\[ \mathbb{E}[\| s_t \|^2] < \infty, \] (9)
and on the use of a decreasing learning rate satisfying
\[ \gamma_t \geq 0, \sum_{t=0}^{\infty} \gamma_t = \infty, \sum_{t=0}^{\infty} \gamma_t^2 < \infty, \] (10)
are necessary. Under the exclusion of trivial solutions such as \( s_t = 0 \), the following convergence assertion can be made:

**Theorem 1 (Polyak and Tsypkin (1973), Thm. 1)** Under conditions (7)–(10), for any \( w_0 \) in process (6):
\[ J(w_t) \to J^* \text{ a.s., and } \lim_{t \to \infty} \nabla J(w_t)^\top \mathbb{E}[s_t] = 0. \]

The significance of the theorem is that its conditions can be checked easily, and it applies to a wide range of cases, including non-convex functions, in which case the convergence point \( J^* \) is a critical point of \( J(w) \).

The convergence analysis of Theorem 1 can be applied to Algorithm 1 as follows: First note that we can define our functional \( J(w) \) with respect to expectations over the full space of \( \mathcal{X} \) as \( J(w) = \mathbb{E}_{p(x)p_{w}(y'|x)}[\Delta(y')] \). This means, convergence of the algorithm can be understood directly as a generalization result that extends to unseen data. In order to show this result, we have to verify conditions (7)–(10). It is easy to show that condition (7) holds for our functional \( J(w) \). Next we match the update in Algorithm 1 to a vector
\[ s_t = \Delta(\hat{y}_t)(\phi(x_t, \hat{y}_t) - \mathbb{E}_{p_{w_t}(y'|x_t)}[\phi(x_t, y')]). \]

Taking the expectation of \( s_t \) yields \( \mathbb{E}_{p(x)p_{w_t}(y'|x)}[s_t] = \nabla J(w_t) \) such that condition (8) is satisfied by
\[ \nabla J(w_t)^\top \mathbb{E}_{p(x)p_{w_t}(y'|x)}[s_t] = \| \nabla J(w_t) \|^2 \geq 0. \]
Assuming \( \| \phi(x, y') \| \leq R \) and \( \Delta(y') \in [0, 1] \) for all \( x, y' \), condition (9) is satisfied by
\[ \mathbb{E}_{p(x)p_{w_t}(y'|x)}[\| s_t \|^2] \leq 4R^2. \]

For a decreasing learning rate, e.g., \( \gamma_t = 1/t \), condition (10) holds, such that convergence to a critical point of the expected risk follows according to Theorem 1.
Algorithm  Structured Dueling Bandits

1: Input: $\gamma, \delta, w_0$
2: for $t = 0, \ldots, T$ do
3: Observe $x_t$
4: Sample unit vector $u_t$ uniformly
5: Set $w'_t = w_t + \delta u_t$
6: Compare $\Delta(\hat{y}_{w_t}(x_t))$ to $\Delta(\hat{y}_{w'_t}(x_t))$
7: if $w'_t$ wins then
8: $w_{t+1} = w_t + \gamma u_t$
9: else
10: $w_{t+1} = w_t$

5 Structured Dueling Bandits

For purposes of comparison, we present an extension of Yue and Joachims (2009)’s dueling bandits algorithm to structured prediction problems. The original algorithm is not specifically designed for structured prediction problems, but it is generic enough to be applicable to such problems when the quality of a parameter vector can be proxied through loss evaluation of an inferred structure.

The Structured Dueling Bandits algorithm compares a current weight vector $w_t$ with a neighboring point $w'_t$ along a direction $u_t$, performing exploration (controlled by $\delta$, line 5) by probing random directions, and exploitation (controlled by $\gamma$, line 8) by taking a step into the winning direction. The comparison step in line 6 is adapted to structured prediction from the original algorithm of Yue and Joachims (2009) by comparing the quality of $w_t$ and $w'_t$ via an evaluation of the losses $\Delta(\hat{y}_{w_t}(x_t))$ and $\Delta(\hat{y}_{w'_t}(x_t))$ of the structured arms corresponding to MAP prediction (3) under $w_t$ and $w'_t$, respectively.

Further, note that the Structured Dueling Bandit algorithm requires access to a two-point feedback instead of a one-point feedback as in case of Bandit Structured Prediction (Algorithm 1). It has been shown that two-point feedback leads to convergence results that are close to those for learning from full information Agarwal et al. (2010). However, two-point feedback is twice as expensive as one-point feedback, and most importantly, such feedback might not be elicitable from users in real-world situations where feedback is limited by time- and resource-constraints. This limits the range of applications of Dueling Bandits to real-world interactive scenarios.

6 Experiments

Our experimental design follows the standard of simulating bandit feedback by evaluating task loss functions against gold standard structures without revealing them to the learner. We compare the proposed Structured Bandit Prediction algorithm to Structured Dueling Bandits, and report results by test set evaluations of the respective loss functions under MAP inference. Furthermore, we evaluate models at different iterations according to their loss on the test set in order to visualize the empirical convergence behavior of the algorithms.

All experiments with bandit algorithms perform online learning for parameter estimation, and apply early stopping to choose the last model in a learning sequence for online-to-batch conversion at test time. Final results for bandit algorithms are averaged over 5 independent runs.

In this experiment, we present bandit learning for the structured $1 - \text{BLEU}$ loss used in SMT. The setup is a reranking approach to SMT domain adaptation where the $k$-best list of an out-of-domain model is re-ranked (without re-decoding) based on bandit feedback from in-
Table 1: Corpus BLEU (under MAP decoding) on test set for SMT domain adaptation from Europarl to NewsCommentary by $k$-best reranking.

<table>
<thead>
<tr>
<th></th>
<th>full information</th>
<th>bandit information</th>
</tr>
</thead>
<tbody>
<tr>
<td>in-domain SMT</td>
<td>0.2854</td>
<td>0.2731 ± 0.001</td>
</tr>
<tr>
<td>out-domain SMT</td>
<td>0.2579</td>
<td>0.2705 ± 0.001</td>
</tr>
</tbody>
</table>

Figure 1: Corpus-BLEU on test set for early stopping at different iterations for the SMT task.

We use the data from the WMT 2007 shared task for domain adaptation experiments in a popular benchmark setup from Europarl to NewsCommentary for French-to-English (Koehn and Schroeder, 2007; Daumé and Jagarlamudi, 2011). We tokenized and lowercased our data using the moses toolkit, and prepared word alignments by fast_align (Dyer et al., 2013). The SMT setup is phrase-based translation using non-unique 5,000-best lists from moses (Koehn et al., 2007) and a 4-gram language model (Heafield et al., 2013).

The out-of-domain baseline SMT model is trained on 1.6 million parallel Europarl data and includes the English side of Europarl and in-domain NewsCommentary in the language model. The model uses 15 dense features (6 lexicalized reordering features, 1 distortion, 1 out-of-domain and 1 in-domain language model, 1 word penalty, 5 translation model features) that are tuned with MERT (Och, 2003) on a dev set of Europarl data (dev2006, 2,000 sentences). The full-information in-domain SMT model gives an upper bound by MERT tuning the out-of-domain model on in-domain development data (nc-dev2007, 1,057 sentences). MERT runs for both baseline models were repeated 7 times and median results are reported.

Learning under bandit feedback started at the learned weights of the out-of-domain median model. It uses the parallel NewsCommentary data (news-commentary, 43,194 sentences) to simulate bandit feedback, by evaluating the sampled translation against the gold standard reference using as loss function $\Delta$ a smoothed per-sentence $1 - \text{BLEU}$ (by flooring zero $n$-gram

domain data. This can also be seen as a simulation of personalized machine translation where a given large SMT system is adapted to a user solely by single-point user feedback to predicted structures.
counts to 0.01). The meta-parameters of Dueling Bandits and Bandit Structured Prediction were adjusted by online optimization of cumulative per-sentence 1 – BLEU on a small in-domain dev set (nc-devtest2007, 1,064 parallel sentences). The final results are obtained by online-to-batch conversion where the model trained for 100 epochs on 43,194 in-domain training data is evaluated on a separate in-domain test set (nc-test2007, 2,007 sentences).

Table 1 shows that the results for Bandit Structured Prediction and Dueling Bandits are very close, however, both are significant improvements over the out-of-domain SMT model that even includes an in-domain language model. We show the standard evaluation of the corpus-BLEU metric evaluated under MAP inference. The range of possible improvements is given by the difference of the BLEU score of the in-domain model and the BLEU score of the out-of-domain model – nearly 3 BLEU points. Bandit learning can improve the out-of-domain baseline by about 1.26 BLEU points (Bandit Structured Prediction) and by about 1.52 BLEU points (Dueling Bandits). All result differences are statistically significant at a p-value of 0.0001, using an Approximate Randomization test (Riezler and Maxwell, 2005; Clark et al., 2011). Figure 1 shows that per-sentence BLEU is a difficult metric to provide single-point feedback, yielding a non-smooth progression of loss values against iterations for Bandit Structured Prediction. The progression of loss values is smoother and empirical convergence speed is faster for Dueling Bandits since it can exploit preference judgements instead of having to trust real-valued feedback.

7 Discussion

We presented an approach to Bandit Structured Prediction that is able to learn from feedback in form of an evaluation of a task loss function for single predicted structures. Our experimental evaluation showed promising results, both compared to Structured Dueling Bandits that employ two-point feedback, and compared to full information scenarios where the correct structure is revealed.

Our approach shows its strength where correct structures are unavailable and two-point feedback is infeasible. In future work we would like to apply bandit learning to scenarios with limited human feedback such as the interactive SMT applications discussed above. In such scenarios, per-sentence BLEU might not be the best metric to quantify feedback. We will instead investigate feedback based on HTER (Snover et al., 2006), or based on judgements according to Likert scales (Likert, 1932).

Acknowledgements

This research was supported in part by DFG grant RI-2221/2-1 “Grounding Statistical Machine Translation in Perception and Action”.

References


Chu, W., Li, L., Reyzin, L., and Schapire, R. E. (2011). Contextual bandits with linear payoff functions. In International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA.


An Empirical Study of Segment Prioritization for Incrementally Retrained Post-Editing-Based SMT

Jinhua Du
jdu@computing.dcu.ie
Ankit Srivastava
asrivastava@computing.dcu.ie
Andy Way
away@computing.dcu.ie
ADAPT Centre, School of Computing, Dublin City University, Ireland
Alfredo Maldonado-Guerra
maldonaa@tcd.ie
David Lewis
dave.lewis@cs.tcd.ie
ADAPT Centre, Trinity College Dublin, Ireland

Abstract
Post-editing the output of a statistical machine translation (SMT) system to obtain high-quality translation has become an increasingly common application of SMT, which henceforth we refer to as post-editing-based SMT (PE-SMT). PE-SMT is often deployed as an incrementally retrained system that can learn knowledge from human post-editing outputs as early as possible to augment the SMT models to reduce PE time. In this scenario, the order of input segments plays a very important role in reducing the overall PE time. Under the active learning-based (AL) framework, this paper provides an empirical study of several typical segment prioritization methods, namely the cross entropy difference (CED), n-grams, perplexity (PPL) and translation confidence, and verifies their performance on different data sets and language pairs. Experiments in a simulated setting show that the confidence of translations performs best with decreases of 1.72-4.55 points TER absolute on average compared to the sequential PE-based incrementally retrained SMT.

1 Introduction
In recent years, SMT systems have been widely deployed into the translator’s workflow in the localization and translation industry to improve productivity, referred to as post-editing-based SMT. However, in most cases, current SMT systems cannot generate high-quality translations, so human effort is usually required. With the help of incrementally improved SMT systems, the productivity of translators/post-editors can be significantly increased due to the early learning of knowledge from the previously post-edited segments (Guerberof, 2009; Plitt and Masselot, 2010; Carl et al., 2011; OBrien, 2011; Zhechev, 2012; Guerberof, 2013). Furthermore, the order of input segments has been found to have a significant impact on the overall PE-time, i.e., an optimized sequence of input segments can reduce the overall PE-time compared to the typical chronological sequence (Dara et al., 2014).

Regarding the PE-SMT, the incremental retraining can be roughly categorized into two different scenarios, namely the segment-level online incremental retraining (segment mode) (Levenberg et al., 2010; Denkowski et al., 2014) and batch-level incremental retraining (batch mode) (Hardt and Elming, 2010; Henriquez Q. et al., 2011; Mathur et al., 2013; Simard and Foster, 2013; Dara et al., 2014; Bertoldi et al., 2014). The former takes one post-edited segment per retraining cycle to immediately update the models, which requires rapid incremental processing of
the word alignment, phrase/rule generation, language model and parameters tuning etc., while
the latter firstly accumulates a batch of segments, and then performs the incremental retraining
process to update the system. The batch-level mode can perform the incremental retraining
process in the background while the translators/post-editors continue to work on the next batch
of segments. From the point of view of parameter estimation, the former can promptly adapt its
feature weights to the newly post-edited segment and learn the translator’s knowledge, but the
frequent change of weights might make the system unstable; the latter adapts the parameters
on an average level of segments in a batch, which can relatively keep the system more robust,
however, it cannot learn the knowledge as early as possible and cannot demonstrate a quick re-
sponse to translator’s practice and preference. In our task, in order to better show the impact of
the order of the input segments on the PE time, we select the batch-level incremental retraining
SMT as our experimental platform.

The order in which post-editors review and correct machine-translated segments has an
impact on the evaluation score (PE time in our case) of the incrementally retrained PE-SMT
systems. That is, assuming that post-editors work on batches, and after post-editing each batch
the SMT system is dynamically retrained, the order of segments in these batches will have an
impact on how quickly the overall translation performance grows. The expectation is that if
post-editors work first on the segments that are most informative or most difficult to translate
for SMT, the SMT system will learn most from the corrections, and as a consequence, trans-
lation quality will increase more steeply in the following retraining iterations. In doing so, it
is possible to devise a process in which the most experienced and potentially more expensive
post-editors/translators tackle the first few batches of segments, leaving the rest of the segments
to either be worked upon by less experienced and potentially cheaper post-editors/translators,
or to be left completely unedited, depending on the quality vs. cost requirements of the actual
translation project at hand. Therefore, in this paper, we carry out an empirical study on sev-
eral different mainstream segment prioritization strategies, and then investigate the factors that
closely correlate to the effectiveness of the methods.

The main contributions of this paper include:

- Confidence of translation and perplexity methods are proposed to reorder the input seg-
  ments in the AL-based dynamically retrained SMT.

- A deep comparison and investigation of different segment prioritization methods for PE-
  SMT using different data sets and language pairs.

- A detailed data and results analysis of the correlation between the reordering score and the
  factors.

- Our experiments show that the unnormalized confidence of translations performs best in
  all tasks and gains around 1.72 to 4.55 TER (Snover et al., 2006) absolute on average.

2 Related Work

The purpose of the input segment prioritization is to reduce the overall PE time to improve
productivity and to reduce the cost. In this scenario, the involvement of human effort implies
that the segment prioritization process can be regarded as AL framework-based PE-SMT. In this
framework, the input segments are ranked based on the information or uncertainty contained
therein. In this section, we will introduce the related work in terms of two aspects: AL-based
framework for PE-SMT; and the incrementally retrained PE-SMT.

The practical active learning framework for SMT was firstly proposed in Haffari et al.
(2009) where a number of high-quality parallel data are acquired from large-scale monolingual
data. Relatively inexpensive human costs are iteratively used to translate information-rich sentences. Experimental results show that generally the translation unit-based selection strategies, namely phrases and $n$-grams, performed best compared to other methods such as random selection, translation confidence, inverse model etc. However, in their work, the AL framework is used for low-resource SMT rather than the PE-SMT scenario. Furthermore, it is a static retraining process in which the test set is constant per iteration, and the retraining procedure is not incremental.

Gonzalez-Rubio et al. (2012) apply AL to the interactive MT in which AL techniques are used to select the most informative sentences to reduce human effort for a given translation quality. Experimental results show that applying AL techniques in an interactive MT setting can prove a better tradeoff between required human effort and final translation quality.

To the best of our knowledge, the most relevant previous work is that of Dara et al. (2014), which proposes a Cross Entropy Difference (CED) criterion to prioritize input segments in an AL framework for PE-based incremental MT update applications. The fundamental goal is to reduce the overall PE time rather than aiming at reducing human effort. The proposed CED method calculates the rank score by the entropy difference of a sentence $s$ in the untranslated corpus (or the incremental data) $U$ and the current training corpus $L$. The higher the score, the more informative the sentence is and the greater the possibility of the sentence being more highly ranked. Experimental results on the industrial data in a simulated setting show that the proposed method significantly reduces the TER score compared to the random and sequential order. In their work, Dara et al. (2014) used batch mode for the incrementally retrained PE-SMT with the CED only considering the information of the source side of the data in order to keep the costs to a minimum for the commercial PE MT applications. However, in the practical scenario, we can take the information of the target side (e.g. translations) into account in batch mode without a significant increase in extra time and human costs by pre-translating the remaining batches in the background while post-editing the current batch. In doing so, we propose to use the confidence of MT translations to rank the segments.

Regarding the incrementally retrained SMT, the most challenging and time-consuming steps are the word alignment and the phrase/rule generation. Ortiz-Martinez et al. (2010) incrementally update the feature values of the phrase table by extracting new phrases from the new sentence pairs based on the pre-stored statistics related to the feature scores. Hardt and Elming (2010) propose a sentence-level retraining scheme in which a local phrase table is created and incrementally updated as a file is translated and post-edited. In their work, a modified revision of GIZA++ (Och and Ney, 2003) is used to approximate word alignments of a newly translated sentence to reduce the incremental training time, and then an additional phrase table is produced from the newly aligned sentences with higher priority. The experiments show the efficiency of the incremental retraining system.

In the incrementally retrained PE-SMT system, suffix arrays (Callison-Burch et al., 2005; Zhang and Vogel, 2005) are a very efficient technique for the incremental retraining process. Levenberg et al. (2010) introduce a dynamic suffix array to incorporate new training text to the current training data. Denkowski et al. (2014) propose an online model adaptation for PE-SMT in which three methods are used for incremental model adaptation: adding new data to a suffix array-indexed bitext from which grammars are extracted, updating a Bayesian language model with incremental data, and using an online MIRA (Crammer and Singer, 2003) to update the parameters. The simulated experiments show that significant improvement in MT quality is achieved when these methods are used individually and in tandem. Germann (2014) proposes a dynamic phrase table strategy for an interactive PE-SMT that computes phrase table entries on demand by sampling a suffix array-indexed bitext. Experiments show that without loss of translation quality, the sampling phrase table achieves good performance in terms of speed.
our task, we use this dynamic phrase table for incremental retraining in Moses (Koehn et al., 2007).

3 The Incrementally Retrained PE-SMT Paradigm

In the post-editing scenario, humans are involved to continuously edit MT outputs into high-quality translations. As discussed in Dara et al. (2014), the fundamental goal of input segment prioritization for PE-SMT is to reduce the overall PE time taken to complete a translation job. The crucial step is to first select the most uncertain sentences or most informative sentences for post-editing in order to learn as much knowledge as possible from these sentences. The workflow of an AL-based incrementally retrained PE-SMT system is as shown in Figure 1.

![Figure 1: The workflow of active learning-based incrementally retrained PE-SMT](image)

In Figure 1, the translations of the input segments are post-edited and the corrected translations are used for incremental update of the models. The process is repeated until the incremental data (or translation job) is finished. In a typical PE scenario, post-editors are presented with SMT outputs in chronological order (i.e. sequentially) of the input segments. However, an optimized order of the input segments in a translation job can significantly reduce the overall PE time.

In our scenario, the PE time is simulated using TER score between the MT output and the reference translations for the sentences in each batch. The overall performance of the segment prioritization method is evaluated by the average TER score for all the batches.

4 Methodology

To prioritize the input segments, an importance score or uncertainty score for the sentence \( s \) must be calculated under some metric, which can be formalized as follows:

Given an initial parallel training corpus \( L := \{(f_i, e_i)\} \) and a monolingual corpus (translation job) \( U := \{f_j\} \), the goal of the segment prioritization is to rank a sentence \( s \) with the score \( \phi(s) \) under a scoring metric \( F \). This process can be defined as a triple in (1):

\[
\phi(s) = F(s, U, L)
\]

Clearly, we can see that the scoring metric \( F \) is most important in a prioritization algorithm.

We use the sequential order of input segments as our baseline.\(^1\) In the following sections,

\(^1\)The random and sequential methods have similar performance in Dara et al. (2014), so we only use sequential as the baseline.
we carry out an empirical study on different information-driven prioritization methods.

4.1 Confidence of Translations (Confidence)

In the decoding process, a translation output $\hat{e}$ is produced with the probability $p(e|f)$ that is calculated by different features, such as bidirectional lexical probabilities, language model etc. It can be treated as a confidence score for the translation because it reflects the translation difficulty or uncertainty of the source segment in some sense.

Generally, the probability $p(e|f)$ is influenced by two aspects, namely the out-of-vocabulary (OOV) words and the sentence length (c.f. Section 6). For human translators, these two aspects are often more time-consuming. That is, a long sentence with many OOVs will take much more time to post-edit. Therefore, intuitively, the unnormalized confidence score of translations can better measure the uncertainty of a sentence.

Based on the confidence of translations, we rank the input segments in an inverted order, i.e. those segments with the lowest MT confidence scores are at the top and those with higher confidence scores are at the bottom.

4.2 Geometry $n$-gram (Geom $n$-gram)

$n$-grams are often used as an information unit to measure the importance score of a sentence. Dara et al. (2014) used an “$n$-gram Overlap method” that computes the unseen score of a sentence $s$ in $U$ by the ratio of $n$-grams not seen in the training data. Particularly, $n$-grams that are seen fewer than $V$ times in the training data are defined as ‘unseen’. However, the “$n$-gram Overlap” method does not consider the information in the incremental data $U$. In our experiments, we utilize the “Geometry $n$-gram” method in Haffari et al. (2009) to calculate the sentence score as in (2):

$$
\phi(s) = \sum_{n=1}^{N} \sum_{x \in X_n} \log \frac{P(x|U, n)}{P(x|L, n)}
$$

where $X_n\{n = 1, \ldots, N\}$ denotes $n$-grams in the sentence $s$, and $P(x|U, n)$ and $P(x|L, n)$ are the probability of $x$ occurring in the set of $n$-grams in $U$ and $L$, respectively, which can be computed via maximum likelihood estimation. $\omega_n$ is the weight that adjusts the importance of the scores of $n$-grams with different lengths. The weights for $\omega_n$ are same as in (Haffari et al., 2009).

From the equation, we can see that “Geom $n$-gram” takes into account the training corpus $L$ and the untranslated corpus $U$ at the same time.

4.3 Perplexity of Sentences (PPL)

In NLP tasks, the perplexity (PPL) is closely related to the concept of entropy, which reflects the degree of uncertainty of the information in a sentence: the larger the entropy, the greater the perplexity, and the more informative the sentence. Thus, we use PPL to calculate the importance score of a sentence $s$ in $U$ as in (3):

$$
\phi(s) = 10^{-\frac{\log p(s)}{N}}
$$

where $N$ is the number of words in the sentence $s$. In our experiments, the language model is trained by SRILM (Stolcke, 2002) using the source side of the parallel data with trigrams.

4.4 Cross Entropy Difference (CED)

This metric is proposed in Dara et al. (2014) for the sentence reranking in the incrementally retrained PE-SMT scenario. In this scenario, given the training corpus $L$ and an incremental
corpus $U$, language models (3-grams) are built from both, and each sentence $s$ in $U$ is scored according to the entropy $H$ difference as in (4):

$$\phi(s) = H_U(s) - H_L(s)$$

(4)

where $H_U(s)$ is the entropy of the sentence $s$ in $U$ and the $H_L(s)$ is the entropy of $s$ in $L$.

The higher the score given to a sentence, the more useful it is to $L$. That is, CED selects sentences from $U$ that are different from $L$ and similar to the overall corpus $U$.

5 Experiments

5.1 Data Settings

In order to have a full and fair study of the prioritization methods, we run our incremental retraining experiments on two open data sets, namely the Europarl² and DGT³ corpora. For DGT data, we use four language pairs, namely English–German (En-De), English–Spanish (En-Es), English–French (En-Fr) and English–Polish (En-Pl), in one direction, i.e. the source language is English. For Europarl data, we use two language pairs bidirectionally, namely English–German and English–Spanish.

For each language pair, we extract 50k pairs of sentences as the parallel training data $L$ for the initial SMT systems, and 10k pairs of sentences as the incremental data $U$ that will be translated, (quasi-) post-edited⁴ and added into the parallel training data iteratively in the retraining cycle. For the Europarl data, we use Newswire 2012 set as the development set (devset) to tune the initial SMT systems. For the DGT data, we extract 2,000 pairs of sentences as the devset to tune the initial SMT systems.

5.2 PE-SMT System Settings

The work flow of our incrementally retrained PE-SMT is shown in Figure 2.

![Figure 2: The work flow of the incrementally retrained PE-SMT in our experiments](http://www.statmt.org/wmt15/translation-task.html)

²http://www.statmt.org/wmt15/translation-task.html
⁴As mentioned before, we use the references of the translated segments as the PEs.
Figure 2 shows a simulated post-editing workflow in which we use the references of the translated segments instead of post-edited MT output per se. $S_i$ indicates the batch of sentences for translation, which is determined by the segment prioritization algorithm. $Model_i$ represents the initial SMT system and the incrementally updated SMT system by the newly post-edited data. $T_i$ indicates the MT outputs of the input segment batch $S_i$, and $T^*_i$ the post-edited MT translations that return to the SMT system for dynamic retraining. This process is repeated until the incremental data set or the translation job is finished. At each retraining cycle, the incremental batch contains 500 segments that are sorted and selected from the incremental data set according to the prioritization method.

A practical incremental retrained PE-SMT system requires a quick update for its related components at each retraining cycle, such as the translation model, language model and parameter weights. In our experiments, we use Moses to build up an incrementally retrained PE-SMT system as:

- the word alignment is performed using incremental GIZA++;
- the translation model is implemented by the dynamic phrase tables based on sampling word-aligned bitexts (Germann, 2014);
- the language model is updated by appending the newly post-edited data to the training data;
- in our experiments, the weights for the features are kept unchanged. The update of parameter weights is time-consuming and is not suitable for real-time incremental retraining. From the viewpoint of system stability, the parameters can perform robustly in a limited range when the data changes. Experiments are conducted on DGT data sets to verify our assumption. The results in terms of BLEU (Papineni et al., 2002) score are shown in Table 1.

<table>
<thead>
<tr>
<th>Pair</th>
<th>Static (%)</th>
<th>Incremental-Seq. (%)</th>
<th>Incremental-Confidence (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>En–De</td>
<td>32.72</td>
<td>32.85</td>
<td>32.88</td>
</tr>
<tr>
<td>En–Es</td>
<td>47.29</td>
<td>47.18</td>
<td>47.14</td>
</tr>
<tr>
<td>En–Fr</td>
<td>44.94</td>
<td>44.68</td>
<td>44.76</td>
</tr>
<tr>
<td>En–Pl</td>
<td>36.15</td>
<td>35.91</td>
<td>36.04</td>
</tr>
</tbody>
</table>

Table 1: Robustness test of parameter weights for PE-SMT (BLEU score)

In Table 1, the numbers are BLEU scores evaluated on a constant test set (or progress set) that contains 2,000 sentence pairs. “Static” indicates that the system is built by adding all incremental data into the initial training data, tuned on the devset and tested on the progress set. “Incremental-Seq.” and “Incremental-Confidence” indicate that the parameter weights are tuned by the initial training data, and kept unchanged during the whole retraining process. The BLEU scores for these two systems are obtained at the last iteration.

We can see that the differences between the incremental systems and the static system are not significant in terms of BLEU score, which show that for the same domain data, the weights are robust in a limited data scale so that it is not necessary for them to be updated per iteration.

5.3 Statistics of Experimental Data

The statistics of entries in two data sets are shown in Table 2 and Table 3. It can be seen that we have similar and consistent distributions of entries for all language pairs and the data sets.

http://code.google.com/p/inc-giza-pp/
### 5.4 Prioritization Experiments

The prioritization experiments are mainly to simulate PE time by the TER score per iteration. The test set at each retraining cycle is dynamic, and contains 500 segments selected from the incremental data according to the prioritisation criteria. The average TER score of the incremental test sets for different language pairs and data sets are shown in Tables 4 and 5.

<table>
<thead>
<tr>
<th>Pair</th>
<th>Sequential</th>
<th>Geom n-gram</th>
<th>PPL</th>
<th>CED</th>
<th>Confidence</th>
<th>Gains</th>
</tr>
</thead>
<tbody>
<tr>
<td>En–De</td>
<td>55.94</td>
<td>56.50</td>
<td>56.58</td>
<td>55.23</td>
<td>51.39</td>
<td>4.55</td>
</tr>
<tr>
<td>En–Es</td>
<td>41.65</td>
<td>42.40</td>
<td>42.48</td>
<td>41.58</td>
<td>38.69</td>
<td>2.96</td>
</tr>
<tr>
<td>En–Fr</td>
<td>44.86</td>
<td>46.92</td>
<td>46.75</td>
<td>44.62</td>
<td>41.42</td>
<td>3.44</td>
</tr>
<tr>
<td>En–Pl</td>
<td>51.38</td>
<td>51.53</td>
<td>51.63</td>
<td>51.16</td>
<td>48.09</td>
<td>3.29</td>
</tr>
</tbody>
</table>

Table 4: Incremental results on DGT data set (TER Score)

<table>
<thead>
<tr>
<th>Pair</th>
<th>Sequential</th>
<th>Geom n-gram</th>
<th>PPL</th>
<th>CED</th>
<th>Confidence</th>
<th>Gains</th>
</tr>
</thead>
<tbody>
<tr>
<td>De–En</td>
<td>73.73</td>
<td>73.87</td>
<td>72.60</td>
<td>72.48</td>
<td>70.56</td>
<td>3.17</td>
</tr>
<tr>
<td>En–De</td>
<td>80.12</td>
<td>80.00</td>
<td>79.38</td>
<td>79.02</td>
<td>77.83</td>
<td>2.29</td>
</tr>
<tr>
<td>Es–En</td>
<td>84.14</td>
<td>84.20</td>
<td>83.75</td>
<td>83.25</td>
<td>81.82</td>
<td>2.32</td>
</tr>
<tr>
<td>En–Es</td>
<td>64.10</td>
<td>64.10</td>
<td>63.37</td>
<td>63.11</td>
<td>62.38</td>
<td>1.72</td>
</tr>
</tbody>
</table>

Table 5: Incremental results on Europarl data set (TER Score)

In Tables 4 and 5, the “Gains” are computed by the best result and the baseline (Sequential). We can see that the best result is obtained by the “Confidence” criterion for all tasks. The decrease in TER score for the “Confidence” criterion range from 1.72 to 4.55 absolute points (2.68~8.13 relative points) compared to the baseline.

It can also be seen that 1) the “CED” criterion beats the baseline in all tasks, and it performs better than other prioritization methods except “Confidence”; 2) the “Geom n-gram” method performs worst in all experiments; 3) the “PPL” method performs slightly better than the baseline only in the Europarl “De–En” and “En–De” tasks.

Figures 3 and 4 show the TER scores of the En–De language pair per iteration for each of the criteria in terms of the DGT and Europarl data sets.\(^6\)

From the figures we can see that there is no obvious decrease (i.e., improvement) for the baseline in terms of TER score. However, the other four prioritization criteria have a trend of

\(^6\)The trends are similar for the other language pairs.
decreasing the TER score, i.e. starting from a higher score and arriving at a lower score for the last iterations. The decreasing trends show the effectiveness of these methods to prioritize the input segments. As in Dara et al. (2014), the improvements over the baseline are shown after the initial 8-9 iterations. In our scenarios, the “Confidence” results in a noticeable decrease of the overall TER score.

In Figure 3, the “CED” and “Confidence” methods have a fluctuation at Iteration 2 and Iteration 8, respectively, but the overall trend decreases in the TER score. In Figure 4, we can see that the TER score at Iteration 1 for the “Confidence” method is over 1 which indicates the MT translation is quite poor and needs too many edits to transform it into a good output sentence.
6 Analysis

The “Confidence” method performs best in our incremental retraining experiments, which motivates us to investigate the hidden reasons by examining: the sentence length distribution and OOVs as well as the correlation between them.

6.1 Sentence Length Distribution

As in Section 5.4, we take “En–De” as an illustration of the sentence length distribution shown in Figures 5 and 6. From both figures, we find that the sentence length distribution of the “Confidence” criterion strongly fluctuates per iteration, i.e. starting from very long sentence length and arriving at very short sentence length. Furthermore, the fluctuations of the length distribution of “Confidence” consistently correspond to the TER score trend in Figures 3 and 4, i.e., when the length is short, the TER score is low and vice versa.

![Sentence length distribution of En–De pair on DGT data](image)

Figure 5: Sentence length distribution of En–De pair on DGT data

The length distributions of the “Geom n-gram”, “PPL” and “CED” methods are more smooth than that of “Confidence”. The “Geom n-gram” always starts from shorter sentences and then the length increases that indicates this method prefers to select short sentences as most informative candidates at the beginning. The distributions of the “PPL” and “CED” methods are quite similar as they both correlate with the entropy.

From the sentence length distributions, we hypothesize that the prioritization score $\phi(s)$ of “Confidence” may correlate to the sentence length of the input segment. We then calculate correlations between the score $\phi(s)$ and the sentence length by the Pearson Correlation,\(^7\) and the results for the En-De language pair are shown in Table 6.

<table>
<thead>
<tr>
<th></th>
<th>Geom n-gram</th>
<th>PPL</th>
<th>CED</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>DGT</td>
<td>0.21</td>
<td>0.009</td>
<td>0.02</td>
<td>0.29</td>
</tr>
<tr>
<td>Europarl</td>
<td>0.14</td>
<td>0.06</td>
<td>0.10</td>
<td>0.48</td>
</tr>
</tbody>
</table>

Table 6: Correlation between the prioritization score and the sentence length of input segments

\(^7\)http://en.wikipedia.org/wiki/Pearsoncorrelationcoefficient

In Table 6, we can see that the “Confidence” method is more correlated to the sentence
length of the input segment than the other methods, which shows that the longer the sentence is, the more difficult it is to be translated.

The relationship between the score $\phi(s)$ and sentence length poses a question: should we normalize the score by the sentence length or not in the segment prioritization task? In order to answer this question and verify that the unnormalized “Confidence” method is more effective to the incremental retrained PE-SMT, we perform a further experiment using the normalized “Confidence”. The results for En–DE on DGT data between these two “Confidence” methods are shown in Figure 7. In Figure 7, we can see that the trends of these two “Confidence” criteria are similar, but the normalized “Confidence” curve is more smooth. The average TER score for the normalized “Confidence” is 56.09 which is much higher (i.e. worse) than the baseline. Based on these results, we can say that the unnormalized “Confidence” method is more effective to reduce the PE time. Some other language pairs in our experiments have similar results.

6.2 OOVs

In SMT, it is known that OOVs are a big problem and significantly influence translation quality. In the Moses decoding process, when an OOV occurs, the probability $p(e|f)$ will be significantly decreased, i.e. the confidence of the translation becomes lower. Thus, the MT output score $\phi(s)$ is not only correlated with the sentence length, but is more closely correlated with the number of OOVs in the sentence.

We then calculate the correlation between the score $\phi(s)$ of “Confidence” and the OOVs by the Pearson Correlation. For En–De scenario, $\rho = 0.9993$ for the DGT data and $\rho = 0.9997$ for the Europarl data. It can be seen that as expected the more OOVs a sentence contains, the lower the confidence score, and the greater the possibility that it is ranked at the top.

6.3 Pros and Cons

The “Confidence” criterion achieved the best performance in our segment prioritization experiments for the incrementally retrained PE-SMT. However, it has some potential disadvantages that should be considered from a practical point of view:

- at each iteration, all the incremental segments need to be translated beforehand, which might be a problem for sentence-level incremental PE-SMT.
Comparison of unnormalized and normalized Confidence method of En–De pair on DGT data

- the sentence length at the first several iterations is much longer than that of the last several iterations, which might significantly increase the amount of post-editing which in turn may adversely affect the perception of translators/post-editors as to the utility of this approach.

However, based on the analysis in the sections above, we know that the performance of the “Confidence” is strongly correlated with the sentence length and OOVs, so we can design a new practical segment prioritization algorithm that only takes into account the training data rather than the translations according to these two crucial factors.

7 Conclusions and Future Work

In this paper, we conducted an empirical study on four different segment prioritization algorithms, namely the Sequential, Geom n-gram, PPL, CED and Confidence methods for incrementally retrained PE-SMT. Experiments conducted on two data sets and several language pairs show that the “Confidence” method achieved the best results in all tasks that reduced the TER score of 1.72-4.55 absolute points. An investigation was carried out to examine the crucial factors that make the “Confidence” effective. Finally, some suggestions are proposed for the design of new algorithms going forward.

In future work, we intend to carry out further studies on incrementally retrained PE-SMT regarding 1) the context problem: the sorted input segments lose the sequential context that is helpful to the translators; 2) developing a new algorithm which fully considers the influence of sentence length and OOVs; 3) carrying out actual PE experiments using our different segment prioritization algorithms.

Acknowledgments

We thank the reviewers for their insightful comments. We also thank our ex-colleague Dr. Sandipan Dandapat for his preliminary work on this topic. This research is supported by Science Foundation Ireland through the ADAPT Centre (Grant 13/RC/2106) (www.adaptcentre.ie) at Dublin City University and Trinity College Dublin, and by Grant 610879 for the Falcon project funded by the European Commission.
References


Effects of Word Alignment Visualization on Post-Editing Quality & Speed†

Lane Schwartz  
lanes@illinois.edu  
Dept. of Linguistics, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Isabel Lacruz  
ilacruz@kent.edu  
Tatyana Bystrova  
tbystro2@kent.edu  
Dept. of Modern & Classical Language Studies, Kent State University, Kent, OH 44242, USA

Abstract
Phrase-based machine translation can be configured to produce alignment data that indicates which machine translated target language words correspond to which original source language words. In most prior work that examined the efficacy of post-editing machine translation, post-editors were presented with machine translations (and in most cases the original source language sentences) without also being presented with source-to-target alignment links. We select four news articles, and ask six Russian-English bilinguals and eleven Spanish-English bilinguals to post-edit English machine translation results, in some cases using alignments and in other cases without. We obtain human adequacy judgements of the post-edited sentences, and demonstrate that when machine translation quality is low, post-editing quality is consistently higher, by a statistically significant amount, when bilingual post-editors are presented with alignment data.

1 Introduction
Post-editing is the process whereby a human user corrects the output of a machine translation system. The use of basic post-editing tools by bilingual human translators has been shown to yield substantial increases in terms of productivity (Plitt and Masselot, 2010) as well as improvements in translation quality (Green et al., 2013) when compared to bilingual human translators working without assistance from machine translation and post-editing tools. More sophisticated interactive interfaces (Langlais et al., 2000; Barrachina et al., 2009; Koehn, 2009b; Denkowski and Lavie, 2012) may also provide benefit (Koehn, 2009a).

The question of how a post-editing interface should be configured and presented to users is a fundamentally interdisciplinary and empirical one. Issues of user interface design, human factors, translation studies, and machine translation quality are all likely relevant. Phrase-based machine translation can be configured to produce alignment data that indicates which machine translated target language words correspond to which original source language words. In most prior work that examined the efficacy of post-editing machine translation, post-editors were presented with machine translations (and in most cases the original source language sentences) without also being presented with source-to-target alignment links.

This work begins an attempt to answer two novel questions regarding post-editing interface design: To what extent, if at all, does the presentation of source-to-target word-level alignment links affect the quality or speed of post-editing? Is any such effect, if it exists, dependent on certain aspects of machine translation quality, or on the language pair?

†All code, scripts, data & analysis files for this paper are at https://github.com/dowobeha/MT_Summit_2015
(a) Russian-English adequacy evaluation guidelines

<table>
<thead>
<tr>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>The post-edited translation is superior to the reference translation</td>
</tr>
<tr>
<td>10</td>
<td>The meaning of the Russian sentence is fully conveyed in the English translation</td>
</tr>
<tr>
<td>8</td>
<td>Most of the meaning of the Russian sentence is conveyed in the English translation</td>
</tr>
<tr>
<td>6</td>
<td>The English translation misunderstands the Russian sentence in a major way, or has many small mistakes</td>
</tr>
<tr>
<td>4</td>
<td>Very little information from the Russian sentence is conveyed in the English translation</td>
</tr>
<tr>
<td>2</td>
<td>The English translation makes no sense at all</td>
</tr>
</tbody>
</table>

(b) Spanish-English adequacy evaluation guidelines

<table>
<thead>
<tr>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>The meaning of the Spanish sentence is fully conveyed in the English translation</td>
</tr>
<tr>
<td>8</td>
<td>Most of the meaning of the Spanish sentence is conveyed in the English translation</td>
</tr>
<tr>
<td>6</td>
<td>The English translation misunderstands the Spanish sentence in a major way, or has many small mistakes</td>
</tr>
<tr>
<td>4</td>
<td>Very little information from the Spanish sentence is conveyed in the English translation</td>
</tr>
<tr>
<td>2</td>
<td>The English translation makes no sense at all</td>
</tr>
</tbody>
</table>

Table 1: Adequacy evaluation guidelines for bilingual Russian-English human judges (Schwartz et al., 2014), and for bilingual Spanish-English human judges (Albrecht et al., 2009). Because no reference translation was available for Spanish-English, the 12 category is omitted.

To address these questions, we conduct a bilingual post-editing experiment (§2) where bilingual post-editors are presented with machine translation output of varying quality, with and without word-level alignment link visualization. In the first condition, we ask six Russian-English bilingual translation students to post-edit two Russian language news articles starting with relatively low quality English machine translation. In the second condition, we ask eleven Spanish-English bilingual translation students to post-edit two Spanish language news articles starting with relatively high quality English machine translation. We find (§3) that when machine translation quality is low, post-editing quality is consistently higher, by a statistically significant amount, when bilingual post-editors are presented with alignment data. We find no statistically significant effect when machine translation quality is high. We also found that for both Russian-English and Spanish-English the mean post-editing times were shorter for texts with alignment than for texts without alignment. These differences were not significant, but the difference for the Russian-English texts approached significance. Finally, in §4 we briefly survey the current state of post-editing research and situate this work within the context of related work in post-editing.

2 Methodology

We hypothesize that the presentation of word-level alignment links to human post-editors may affect the quality or speed of the resulting output, and that such effects may be dependent on the quality of the underlying machine translations. To test this hypothesis, we conduct a bilingual post-editing experiment where bilingual post-editors are presented with machine translation output of varying quality, with and without word-level alignment link visualization.

2.1 Bilingual Participants

2.1.1 Russian-English Bilingual Participants

There were six participants who served as post-editors in the Russian-English portion of this study, all of whom were paid for their time at the rate of $25 for each hour or part of an hour.
These participants were all English-Russian bilinguals. We designate these participants as PE1–PE6. Four of the six bilingual participants (PE2, PE3, PE4, & PE6) had Russian as their first language (L1) and were highly proficient in English as their second language (L2). The other two bilingual participants (PE1 & PE5) had English as their first language and were highly proficient in Russian as their second language. Three of the six bilingual participants were graduate students and three were undergraduate students; all were enrolled in a university Russian Translation program.

2.1.2 Spanish-English Bilingual Participants

There were eleven participants who served as post-editors in the Spanish-English portion of this study, all of whom were paid for their time at the rate of $25 for each hour or part of an hour. They were all Spanish-English bilinguals. We designate these participants as PE7–PE17. The first language (L1) of all eleven participants was English, and all eleven were highly proficient in Spanish as their second language (L2). These participants were students in a university Master of Spanish Translation program.

2.2 Source Language Data

2.2.1 Russian Data

For the Russian-English portion of this study, we selected as source texts a subset of the texts from the 2014 Workshop on Statistical Machine Translation (WMT-14) shared translation task (Bojar et al., 2014). Source texts were news articles covering world news events in late 2013. The first text was originally a Russian-language BBC news article covering Syrian chemical weapons. The second text was originally an English-language news article covering U.S. spying policy. We designate the former as Doc A and the latter as Doc B.

These two texts were each divided into segments (32 and 33 segments, respectively) that corresponded to sentences or stand-alone phrases (typically corresponding to news headlines, captions, or cutlines). Segments in Doc A varied in length from 3 to 35 words (mean length 17 words); segments in Doc B varied in length from 9 to 55 words (mean length 23 words).

Professional translations of Doc A into English and Doc B into Russian were commissioned as part of the WMT-14 shared translation task (Bojar et al., 2014). The Russian version of each text was translated automatically using Moses (Koehn et al., 2007) by Schwartz et al. (2014) as part of their WMT14 shared task submission. As a side effect of the phrase-based MT process, Moses can be configured to produce alignment links, indicating which target language words were produced from which source language words. To enable maximal comparability with the post-editing results of Schwartz et al. (2014), we make use of Russian-English machine translation results and alignments from that work here.

2.2.2 Spanish Data

Two Spanish source texts were selected. Both were extracts from a news article from a Spanish newspaper covering current world news events. The two texts were divided up into segments that corresponded to sentences or stand-alone phrases. The first text had 26 segments that varied in length from 4 to 24 words (mean length 15 words) and the second text had 25 segments that varied in length from 4 to 28 words (mean length 16 words). We designate the former as Doc C and the latter as Doc D. No reference translations exist for either Spanish text.

The Spanish source texts were translated automatically using Microsoft Bing Translator through its online developer API. Bing Translator, when accessed via the developer API, can be configured to return character-level alignment links from source characters to target characters, in addition to translated target language sentences. Our scripts derive word alignments from the character alignments returned by the Bing Translator API.
2.3 Translation Quality

We hypothesize that any effects of word alignment visualization on post-editing may be dependent on the quality of the underlying machine translations displayed to the post-editors. Because we care about the adequacy of post-edited translations, we consider actual human judgements to be preferable to automated metrics such as BLEU (Papineni et al., 2002), which at best serve as a flawed proxy for human judgements. Instead, following Albrecht et al. (2009) and Schwartz et al. (2014), we therefore obtained human judgements of translation adequacy for the Russian-English and Spanish-English machine translations used in this study.

The Russian language news articles used in this study have corresponding reference translations. It is therefore possible (although given current machine translation quality, highly unlikely) that machine translation quality for any given segment could conceivably surpass the quality of the corresponding reference translation (if for example, the reference translator makes a mistake). For assessing the quality of the Russian-English machine translations, then, we follow the 12-point adequacy scale of Schwartz et al. (2014). This adequacy scale is shown in Table 1a on page 2; this scale ranges from a low of 2 (the English translation makes no sense at all) to a high of 12 (the translation is superior to the reference).

The Spanish language news articles used in this study lack corresponding reference translations. Thus, unlike the case of our Russian data, no matter how high the quality of machine translations, no Spanish-English machine translation segment could possibly receive a score of 12. For Spanish-English, we therefore follow the 10-point adequacy scale of Albrecht et al. (2009). This adequacy scale is shown in Table 1b on page 2; this scale is very similar to the former, but has a high of 10 (the meaning of the source sentence is fully conveyed in the English translation) instead of 12.

2.4 Post-Editing Interface

For this study, we developed a novel post-editing interface, based on the open source software used and released by Schwartz et al. (2014). Our software is written using Scala (Odersky, 2014), and is released as open source (see the software supplement that accompanies this work). This code constitutes a ground-up rewrite of the Java-based post-editing interface of Schwartz et al. (2014), written using a strict model-view-controller software design pattern to be easy for other researchers to use and extend.

Our post-editing interface can be seen in Figure 1 above. Each text was presented to post-editors in one of two variant modalities — word-level alignment links could either be visualized or left absent. In both variants, each source language segment was presented along with the corresponding machine translated English segment; a text field (initially populated with the machine translated segment) where the post-editor could make changes was also presented. In the first variant, the word-level alignment links produced by the machine translation decoder (Moses for Russian-English, Bing Translator for Spanish-English) were graphically displayed, linking source words to their corresponding machine translated target words. In the second variant, the word-level alignment links were omitted from the visualization interface.
Figure 2: Percentage of segments judged to be in each adequacy category. For each language pair, we report percentages for raw (unedited) machine translation output, as well as output post-edited by a bilingual post-editor with access to alignments and without access to alignments. For Russian-English, we additionally report percentages for output post-edited by a monolingual post-editor (Schwartz et al., 2014) with access to alignments.
2.5 Procedure

Participants performed the test individually in an office setting and were instructed to minimally post-edit. Specifically, participants were asked to disregard issues of style and to focus on a) how well the translation conveyed the meaning of the source text, and b) the grammatical correctness of the translated segments. Participants sat in front of a computer that displayed the source texts divided up into segments (see Figure 1 on page 4). Directly below each source text segment, its machine translation was displayed. Below that was an active response area, where participants were asked to carry out the post-editing.

During initial data collection (the Russian-English condition), the only data collected was the final post-edited output and the overall time taken per text. Subsequently, we enhanced the post-editing software with additional logging functionality, enabling the software to record key-logging and mouse-logging data. For the subsequent Spanish-English condition, this enhanced software was utilized; for this condition millisecond-precision keyboard-event and mouse-event logs were recorded in addition to collecting final post-edited output and overall time taken per text.

We believe that scientific inquiry is at its strongest when experiments can be easily replicated, and when the raw and processed data from such experiments can be directly verified by reviewers, readers, and other experimenters. In that spirit, all data and code produced or used in this work are provided in the attached dataset and software supplements. This includes all logs, along with raw machine translation output, alignment data, post-edited output, adequacy judgements, post-editing software, and supplementary scripts.

2.5.1 Russian-English Participant Assignment

Each participant was instructed to edit one of the two texts using the interface where alignment links were shown, and to edit the other text using the interface where alignment links were omitted. Participants post-edited the two texts in one session lasting less than two hours, although there were no time limits set for the task. The experimenter was present in the room and manually recorded the times taken. Post-Editors 2, 4, and 6 were assigned to post-edit Doc A using the variant 1 interface that displayed alignments, and Doc B using the variant 2 interface that omitted alignments. Post-Editors 1, 3, and 5 were assigned Doc A using variant 2 and Doc B using variant 1.

2.5.2 Spanish-English Participant Assignment

Participants post-edited one text using the interface where alignment links were shown and the other text using the interface where alignment links were omitted. Participants post-edited the two texts in one session lasting less than one hour. Timings were recorded by a keylogger. Post-editors 7, 9, 11, 13, 15, and 17 edited Doc C using the interface that omitted the alignments and Doc D using the interface that displayed the alignments. Post-editors 8, 10, 12, 14, and 16 edited Doc C using the interface that displayed the alignments and Doc D using the interface that omitted the alignments.

3 Results

3.1 Rating Translation Adequacy

Following the methodology outlined in §2.3, all post-edited output as well as all machine translations were evaluated by bilingual human judges using the adequacy scales shown in Table 1.

3.1.1 Rating Adequacy of Russian-English

Following the adequacy guidelines from §2.3, an experienced English-Russian translator and grader rated all English output translations of the Russian-English post-edited segments. In addition, all English machine translations of the Russian documents were manually rated for
Figure 3: Mean adequacy score, categorized by the adequacy score of the unedited MT. The red horizontal line indicates the mean adequacy score (Russian-English: 6.1; Spanish-English: 7.1) of the unedited MT.
adequacy.

For each segment, the human rater was presented with a vertically-arranged list showing all variants of that segment. The first entry in each list was the segment in the source language (Russian). The source segment was followed by the reference translation in English. The subsequent eight entries were English translations of the source segment, presented in a randomized order. The English translations included the unedited machine translation output, as produced by Moses, a post-edited translation produced by a monolingual post-editor from Schwartz et al. (2014), and the six post-edited translations produced by the Russian-English bilingual post-editors in this study.

All Russian-English translations were rated using the translation adequacy scale in Table 1a on page 2, with possible ratings ranging from 2 (translation makes no sense) to 12 (translation is superior to the reference translation).

3.1.2 Rating Adequacy of Spanish-English
Following the adequacy guidelines from §2.3, an experienced Spanish-English translator and grader worked in cooperation with a second Spanish-English bilingual to rate all English output translations of the Spanish-English post-edited segments. In addition, all English machine translations of the Spanish documents were manually rated for adequacy. For each segment, the human raters were presented with a vertically-arranged list showing all variants of that segment. The first entry in each list was the segment in the source language (Spanish). The subsequent twelve entries were English translations of the source segment, presented in a randomized order. The English translations included the unedited machine translation output, as produced by Bing Translator, and the eleven post-edited translations produced by the Spanish-English bilingual post-editors in this study.

Unlike the Russian documents, no reference translation was available for the Spanish documents; for this reason (as described in §2.3), the top category (12) used in evaluating Russian-English segments was omitted from the Spanish-English evaluation. All Spanish-English translations were rated using the translation adequacy scale in Table 1b on page 2, with possible ratings ranging from 2 (translation makes no sense) to 10 (the meaning of the Spanish sentence is fully conveyed in the English translation). Unlike all other participants, participant PE17 consistently produced post-edited segments of lower adequacy than the corresponding raw MT output. This participant was therefore dropped from all analyses of the Spanish-English data.

3.2 Adequacy Results
Figure 2 on page 5 presents the percentage of segments judged to be in each adequacy category. Mean adequacy scores for each experimental condition are presented in Figure 3 on the previous page. By subtracting the adequacy score of each machine translated segment, we obtain the adequacy gain obtained by post-editing; these values are presented in Figure 4 on the following page. Finally, Figure 5 on page 11 presents mean adequacy score by post-editor. We now analyze these results by experimental condition.

3.2.1 Machine Translation Adequacy
In Figure 2 on page 5, we observe that the Russian-English machine translation segments tend to be of lower quality (as measured by adequacy), while the Spanish-English machine translation segments tend to be of higher quality. Two-thirds of Russian-English machine translation segments are judged to have major errors (ratings 2-6), while one-third are rated as mostly or completely correct (8-12). Contrast this with the Spanish-English machine translations segments; a minority (about two-fifths) are judged to have major errors (ratings 2-6), while the majority (about three-fifths) are rated as mostly or completely correct (8-10).
Figure 4: Mean gain in adequacy score over unedited MT, categorized by the adequacy score of the unedited MT.
3.2.2 Russian-English Adequacy
The mean adequacy score when bilingual participants were presented with alignments was 8.35. When alignments were omitted from the post-editing tool, the mean adequacy score was 7.85. A Wilcoxon signed-rank test (Wilcoxon, 1945) showed that when participants were presented with alignments the ratings of their translations were significantly higher than when participants post-edited without access to alignments (N = 6, Z = -2.207, p = 0.027).

3.2.3 Spanish-English Adequacy
The mean adequacy score when Spanish bilingual participants were presented with alignments was 8.22. When alignments were omitted from the post-editing tool, the mean adequacy score was 8.02. These means are not significantly different.

3.3 Timing Results
3.3.1 Russian-English Timing
The times taken by the Russian-English post-editors to post-edit each text were recorded manually to the nearest minute. Participants post-edited each text without a break. They took a break of approximately five minutes between texts. The mean times were 33 minutes for texts with alignment and 40 minutes for texts without alignment. This difference approached significance (p = .082).

3.3.2 Spanish-English Timing
The times taken by the Spanish-English post-editors to post-edit each text were recorded by the keylogger to nearest millisecond. Participants post-edited each text without a break. They took a break of approximately five minutes between texts. The mean times were 21 minutes and 5 seconds for texts with alignment and 22 minutes and 5 seconds for texts without alignment. An independent samples t-test showed that these times are not significantly different from each other ((t(18) = .295, p = .77). Participants were not given time limitations, so timing data must be interpreted with caution. Note, however, that the mean time with alignment is numerically shorter than the mean time without alignment. The shorter editing times for Spanish-English may in part be explained by the shorter length of these documents.

4 Analysis and Related Work
Our results suggest that when machine translation quality is poor (2–4), bilingual post-editors may produce higher quality translations when presented with bilingual alignment links between source words and machine-translated target words. Alternatively, when machine translation quality is high (8–12), no effect is seen by presenting alignment visualizations. We explain this by hypothesizing that word alignment visualization may enable post-editors to better recover from certain types of translation errors produced by MT systems; when MT quality is high enough that such errors are absent, word alignment visualization may no longer play a restorative role.

We examine the effect that alignment link visualization has on each bilingual post-editor in Figure 5 on the next page. In the Russian-English condition, where overall MT quality is poor, we observe that post-editing quality varies widely between post-editors (with PE2 and PE3 performing best). For all six bilingual post-editors, we observe higher mean adequacy scores when alignment links were presented than when they were omitted from the post-editing tool. We also note that when alignment links were absent, one bilingual post-editor (PE5) performed worse than the monolingual post-editor (PE0) from Schwartz et al. (2014). On the other hand, in the Spanish-English condition, where overall MT quality is good, we observe relatively little variation in quality between the ten post-editors. When compared to the unedited machine trans-
Figure 5: Mean adequacy score per post-editor. The red horizontal line indicates the mean adequacy score (Russian-English: 6.1; Spanish-English: 7.1) of the unedited MT.
lations, post-editing resulted in improved mean adequacy for all post-editors, both bilingual and monolingual.

Our results also suggest that post-editing time tends to be reduced for texts with alignment. These numerical reductions were consistent across all the Russian-English participants with the exception of PE3, but they were not consistent for the Spanish-English participants.

We hypothesize that texts with alignment are less cognitively demanding to process, and so less effortful to post-edit than texts without alignment. If this is the case, shorter post-editing times for texts with alignment are consistent with previous findings by Koponen et al. (2012), who found that per word post-editing times were shorter for segments that were less cognitively demanding because of the linguistic structure. Related work on cognitive effort in post-editing (Lacruz et al., 2014; Lacruz and Shreve, 2014) has also shown decreased densities of short pauses when less cognitively demanding segments are post-edited.

The keystroke logging data gathered for Spanish-English post-editors allowed the computation of Pause to Word Ratio (PWR). For each segment, PWR is the ratio of the number of pauses exceeding 300ms to the number of words in the MT segment; it is a measure of cognitive effort in post-editing (Lacruz and Shreve, 2014). Higher PWR corresponds to higher cognitive effort. Contrary to expectation, the mean PWR for Spanish-English post-editors was slightly higher for the segments with alignment (0.70) than for those without alignment (0.63). However, the numerical difference was not significant.

It is possible that the effect of alignment on PWR was masked by the fact that the adequacy of the Spanish-English MT segments was generally high. Since our prediction is that alignment should both increase post-editing adequacy and reduce post-editing effort, the Gain to Effort Ratio, GER = (PE Rating − MT Rating)/PWR is a promising metric to investigate. We hypothesize that GER is higher for segments with alignment than for segments without alignment.

Figure 6 above shows GER for Spanish-English. GER for segments with alignment was 2.53, and GER for segments without alignment was 1.47. Our prediction was confirmed: a paired samples t-test showed that GER was higher for segments with alignment (t(9) = 2.49, p =.034.). This result suggests that GER may be a robust metric for measuring the effects of alignment on post-editing. It would be interesting to conduct further studies involving language pairs different from Spanish-English where the adequacy of machine translations may be lower.

5 Conclusion

In this work, we observe that when machine translation quality is poor, bilingual post-editors may produce higher quality translations when presented with bilingual alignment links between source words and machine-translated target words. We explain this by hypothesizing that word alignment visualization may enable post-editors to better recover from certain types of translation errors produced by MT systems; when MT quality is high enough that such errors are absent, word alignment visualization may no longer play a restorative role. The timing results we observe, while not statistically significant, appear to be consistent with prior work that found per word post-editing times to be shorter for segments that were less cognitively demanding.
References


A Systematic Evaluation of MBOT in Statistical Machine Translation

Nina Seemann seemanna@ims.uni-stuttgart.de
Fabienne Braune braunefe@ims.uni-stuttgart.de
Andreas Maletti maletti@ims.uni-stuttgart.de
Institute for Natural Language Processing, University of Stuttgart,
Pfaffenwaldring 5b, 70569 Stuttgart, Germany

Abstract
Shallow local multi-bottom up tree transducers (MBOTs) have been successfully used as translation models in several settings because of their ability to model discontinuities. In this contribution, several additional settings are explored and evaluated. The first rule extractions for tree-to-tree MBOT with non-minimal rules and for string-to-string MBOT are developed. All existing MBOT systems are systematically evaluated and compared to corresponding baseline systems in three large-scale translation tasks: English-to-German, English-to-Chinese, and English-to-Arabic. Particular emphasis is placed on the use of discontinuous rules. The developed rule extractions and analysis tools will be made publicly available.

1 Introduction
The area of statistical machine translation (SMT) (Koehn, 2009) concerns itself mostly with the development of automatic methods of deriving probabilistic translation models from a parallel corpus. Such a corpus contains corresponding sentences in two languages. The rule extraction mechanism automatically extracts rules for the translation model from such a corpus and assigns them a probability. Several different translation approaches are currently discussed in SMT. Among the best-performing systems one often finds phrase-based (Koehn et al., 2003) and syntax-based translation systems. Phrase-based systems use no linguistic information and can be obtained directly from the corpus. The same observation is true for hierarchical phrase-based systems (Chiang, 2005) as those are syntactic in a formal sense only. Syntax-based systems require some form of syntactic annotation, which is often represented as a (parse) tree. Correspondingly, we obtain several models, namely tree-to-tree, string-to-tree, and tree-to-string systems, which use trees on the source or the target language side. A multitude of formalisms has been proposed as syntax-based translation models. The most prominent might be the synchronous tree substitution grammars (STSGs) of Eisner (2003) and the non-contiguous synchronous tree sequence substitution grammars (STSSGs) of Sun et al. (2009). Recently, Maletti (2011) proposed local multi bottom-up tree transducers (MBOT) as a translation model for syntax-based SMT. An MBOT is an extension of an STSG that allows sequences of tree fragments on the target side of its rules. In this manner, it can model discontinuities. It can also be understood as a restricted form of a STSSG, in which the rules consist of sequences of source and target tree fragments.

Recently, MBOTs have been implemented as a translation model inside the Moses framework (Koehn et al., 2007) by Braune et al. (2013). Initially, only the rule extraction for minimal
tree-to-tree rules of Maletti (2011) was available. In combination with synchronous context-free grammar (SCFG) rules, already this system led to significant improvements over a corresponding system using SCFG rules only. Later, Seemann et al. (2015) proposed a rule extraction procedure for string-to-tree MBOTs that is able to extract non-minimal rules. Since the number of such rules usually explodes, certain (parametric) restrictions are imposed on the extracted rules in the same spirit as in (Chiang, 2005). Also in this setting, significant improvements over a corresponding SCFG-based system are obtained. However, systems imposing even further syntactic restriction still outperform those systems in some cases. To understand the benefits of the MBOT model and its dependence on syntax, we develop additional rule extractions for the missing scenarios and provide a systematic evaluation of MBOT-based systems. In particular, we develop a rule extraction for non-minimal tree-to-tree rules. It is known that minimal rules are rather restrictive (Galley et al., 2004), so we expect sizable improvements from the obtained tree-to-tree system (when compared to the tree-to-tree system using only minimal rules). In addition, we explore whether the discontinuous rules of MBOTs remain useful without any syntax. Consequently, we also develop a string-to-string rule extraction for MBOT. All the mentioned MBOT models are systematically evaluated and compared to popular translation models.

We evaluate our models on three large-scale translation tasks: English-to-German, English-to-Arabic, and English-to-Chinese. We demonstrate that the expected improvements from the non-minimal rules is indeed realized. However, the tree-to-tree SCFG baseline system, which also utilizes non-minimal rules, still outperforms the MBOT model. In the string-to-tree setting, MBOT significantly outperform SCFG as demonstrated by Seemann et al. (2015). Finally, in the string-to-string setting, discontinuous rules seem to be hardly useful at all. The obtained evaluation scores for such string-to-string (hierarchical) systems are comparable. Only a detailed analysis of the number of the used rules that are (potentially) discontinuous indeed confirms that hardly any discontinuous rules are used when decoding the test set. Overall, these results seem to suggest that discontinuous rules are most successful in the string-to-tree setting, where the strict syntactic structure of the output tree makes discontinuities valuable, while the flat structure of the input (string) allows sufficient freedom. Chiang (2010) arrives at a similar conclusion for general syntax-based systems with the argument that the target-side syntax might enable more grammatical translations. String-to-tree MBOTs offer an even better performance than string-to-tree SCFGs in our translation tasks, so discontinuities seem to be very relevant.

2 Statistical Machine Translation with MBOTs

Syntax-based statistical machine translation models use the syntactic structure of the input and/or output sentences during training and decoding. The syntactic structure of the sentences is typically automatically obtained with the help of a (constituent) parser, so we use ‘parse’ to refer to the syntactic structure. Several different settings can be distinguished depending on the use of parses:

- **tree-to-tree**: In this setting, parses are used for both the source and target sentences during training and for the input sentences during decoding.
- **tree-to-string**: Parses are used only for the source language; i.e., for the source sentences during training and for the input sentences during decoding.
- **string-to-tree**: In this setup, parses are only used for the target language sentences.
- **string-to-string**: No parses are used and the system is not syntax-based.

Shallow local multi bottom-up tree transducers (MBOTs) have already been successfully applied as translation model in the tree-to-tree setting (Braune et al., 2013) as well as in the string-to-tree setting (Seemann et al., 2015). It is generally accepted that tree-to-string systems yield worse performance than string-to-tree systems (Chiang, 2010). We complete the picture by adding non-minimal rules to the tree-to-tree system, establishing a string-to-string variant (i.e., similar
to a hierarchical phrase-based model) that requires no parses at all, and providing an extensive
evaluation of all those variants.

Let us start by illustrating the different variants. To this end, we first show the general
rule shape of MBOTs, and then recall the existing rule extraction algorithms together with the
particularities of the obtained rules. Our presentation is necessarily illustrative only. For the
formal details we refer the reader to (Maletti, 2011; Braune et al., 2013; Seemann et al., 2015).

Roughly speaking, each MBOT has terminal symbols (e.g., lexical items) and nonterminal
symbols (e.g., part-of-speech tags and syntactic categories). In essence, an MBOT is simply a
finite set of rules. Each rule $\ell \rightarrow r$ consists of a left-hand side $\ell$ and a right-hand side $r$. The
left-hand side $\ell = t$ consists of an object $t$ (string or tree) for the source side, and the right-hand
side $r = (t_1, \ldots, t_m)$ similarly consists of a sequence $t_1, \ldots, t_m$ of objects for the target side.
The objects $t, t_1, \ldots, t_m$ are either strings or trees (with the restriction that $t_1, \ldots, t_m$ have the
same type) formed from the terminal and nonterminal symbols with the additional restriction
that each exposed occurrence of a nonterminal in an object in the right-hand side is linked
to an exposed occurrence of a nonterminal in the left-hand side. More precisely, in a string,
the lexical items are the terminal symbols and each nonterminal occurrence is exposed. In a
tree, the lexical items only occur as leaves and additionally nonterminals are allowed as leaves.
However, an occurrence of a nonterminal is exposed if and only if it is a leaf. The 4 different
settings naturally correspond to the choice of strings or trees for the source and target side
objects $t$ and $t_1, \ldots, t_m$, respectively.

2.1 Minimal tree-to-tree rule extraction

Let us start with the tree-to-tree setting, for which a rule extraction for minimal rules\footnote{A rule is minimal if it cannot be obtained by means of substitution from others.} was
proposed in (Maletti, 2011). In this case, the rule extraction requires a word-aligned, bi-parsed
(parsed on both sides) parallel corpus. A sample entry of such a corpus is shown left in Figure 1.
The rule extraction is applied to each sentence pair of the corpus and essentially performs the
following steps:

(1) Select a minimal number of alignment edges $E$ such that

- the maximal (non-leaf nonterminal) source node $v$ containing (as leaves) all sources
  of the selected edges and no sources of non-selected edges exists and

Figure 1: Word-aligned, bi-parsed sentence pair before (left) and after (right) excision of a rule.
the maximal (non-leaf nonterminal) target nodes \( w_1, \ldots, w_m \) containing all targets of
the selected edges and no targets of non-selected edges exists.

In other words, the selected set \( E \) of edges admits maximal subtrees (below nodes \( v \) and \( w_1, \ldots, w_m \)) that are consistent with the word alignment (Chiang, 2005).

(2) Excise the subtree \( t \) below \( v \) for the source side and the subtrees \( t_1, \ldots, t_m \) below \( w_1, \ldots, w_m \), respectively, for the target side. In this way, we obtain the tree-to-tree rule \( t \rightarrow (t_1, \ldots, t_m) \). After the excision, the nonterminals at \( v \) and \( w_1, \ldots, w_m \) remain as leaf nonterminals and are linked. The result of excising the rule containing ‘predicted’ from the left entry of Figure 1 is shown right in Figure 1.

(3) Repeat the process with the linked pair of trees obtained after the excision.

Figure 2 shows all minimal MBOT rules that can be extracted from the word-aligned, bi-parsed sentence pair of Figure 1. The obtained rules are made shallow (Braune et al., 2013) by removing the internal (i.e., non-root and non-leaf) nodes. Figure 3 shows this process on the only rule of Figure 2 that is not yet shallow.

2.2 Non-minimal string-to-tree rule extraction

The parametrized non-minimal rule extraction of Seemann et al. (2015) for string-to-tree MBOT rules is an extension of the rule extraction of Chiang (2005) for hierarchical rules. As expected from the string-to-tree setting, it extracts rules from a word-aligned parallel corpus with parses for the target language side. Figure 4 shows an entry in such a corpus. Recall that a string-to-tree rule has the shape \( w \rightarrow (t_1, \ldots, t_m) \) for a string \( w \) and trees \( t_1, \ldots, t_m \). In the source string \( w \), we only allow lexical items and the nonterminal \( X \). The rule extraction proceeds similarly as described in Section 2.1 with the exception that a phrase (or span) is selected in the source side (instead of a node of the tree) and that the minimality and maximality conditions are dropped (non-minimality). When excising from the source side, weleave the nonterminal \( X \) instead of the excised material. In essence, we obtain all string-to-tree rules that are consistently word-aligned in this manner. However, there are way too many such rules, and Seemann et al. (2015) establish the following conditions on \( w \) that need to be fulfilled for a rule to be extracted.

- It should have (i) a lexical item that is the source of a word alignment or (ii) an occurrence of \( X \) (i.e., selecting the empty set \( E \) of edges is excluded).
- It should correspond to a span of length at most 10 and contain at most 5 occurrences of

![Figure 3: Non-shallow MBOT rule (left) and its shallow counterpart (right).](image-url)
Official forecasts predicted just 3 %

 lexical items or X.

• It cannot start with X and consecutive ‘X’ are forbidden.

All extracted rules are made shallow as in Section 2.1. In Figure 5 we illustrate some string-to-tree rules that are extracted from the sentence pair left in Figure 4. The excision of the rule with left-hand side ‘predicted’ from the sentence pair is illustrated right in Figure 4.

3 Non-minimal tree-to-tree rule extraction

To provide a complete picture, we also want to consider a non-minimal tree-to-tree rule extraction. To this end, we modify the (non-minimal) string-to-tree rule extraction of Section 2.2 to extract tree-to-tree rules as follows. We first extract string-to-tree rules, so let \( r = w \rightarrow (t_1, \ldots, t_m) \) be such an extracted rule. Let \( w = w_1 \cdots w_n \) be the decomposition of the string \( w \) into tokens. Since we want to extract tree-to-tree rules, our sentence pairs are now bi-parsed (as in Figure 1), so a parse of the source sentence is available. Based on this parse, we determine whether the left-hand side \( w \) corresponds to a constituent in it. If it corresponds to a constituent labeled \( N \), then we construct the tree-to-tree rule \( N(w'_1, \ldots, w'_n) \rightarrow (t_1, \ldots, t_m) \), where \( w'_i \) is simply \( w_i \) for all lexical items \( w_i \) and the corresponding constituent for \( w_i = X \). Otherwise we ignore the string-to-tree rule \( r \) and proceed with the next one. For example, the string-to-tree rule

\[
\text{predicted just } 3 \% \rightarrow (\text{VAFIN(sind)}, \text{ADV(nur)}, \text{VVPP(ausgegangen)})
\]

is extracted in the string-to-tree setting for the sentence pair of Figure 4 (left), but ‘predicted just’ is not a constituent in the source sentence parse of Figure 4 (left). Consequently, this string-to-tree rule is ignored.

Since the left hand side \( w \) has to match a constituent, the number of extractable rules is drastically lower than in the string-to-tree setting. Hence, we can remove the last of the conditions described above. Note that rules like those in Figure 2 can be extracted using the
non-minimal tree-to-tree rule extraction. Some additional tree-to-tree rules that can be extracted from the word-aligned, bi-parsed sentence pair left in Figure 1 are displayed in Figure 6.

4 String-to-string rule extraction

Secondly, we want to completely abandon the syntactic annotation and derive a rule extraction for string-to-string MBOT rules. We again achieve this by a simple modification of the existing string-to-tree rule extraction of Seemann et al. (2015). Overall, string-to-string MBOT rules are similar to hierarchical rules (Chiang, 2005), but with a sequence of strings in the right-hand side. We now have no parses at all, so our training data is a simple word-aligned parallel corpus. An entry of such a corpus is displayed in Figure 7. To accommodate this situation, we perform the same changes mentioned in the step from the tree-to-tree rule extraction to the string-to-tree rule extraction also for the target side. Thus, we no longer need to identify a node in the target sentence parse, but rather identify a sequence of phrases (or spans) that are consistent with the word alignment. The additional restrictions on the left-hand side \( w \) in the string-to-tree rule extraction are also imposed onto the left-hand sides of the extracted string-to-string rules, but we do not impose them onto the right-hand side. These restrictions were imposed to reasonably limit the number of rules, but it shows that restricting the right-hand side is (generally) not necessary.

5 Experimental Evaluation

Our main contribution is the experimental evaluation of MBOTs in the various settings (tree-to-tree, string-to-tree, and string-to-string). We also compare to standard models (SCFG and
hierarchical models) in order to evaluate the effect of the discontinuities offered by the MBOT model. We try to be comprehensive, but naturally we can only report results for a limited number of experiments. We chose to perform experiments for the translation directions English-to-German, English-to-Arabic, and English-to-Chinese. The languages were selected such that constituency parsers and large parallel corpora are readily available. In addition, we selected target languages, in which the discontinuity offered by the MBOT model might be useful.

5.1 Resources

For better comparisons, we use exactly the same resources as Seemann et al. (2015) for the evaluation. We summarize the experimental setup in Table 1. We applied length-ratio filtering to all data sets. Furthermore, all training sets have been word aligned using GIZA++ (Och and Ney, 2003) using the grow-diag-final-and heuristic (Koehn et al., 2005).

The tasks required various forms of preprocessing of the data. The English (source) side of the training data was true-cased and parsed with the provided grammar of the Berkeley parser (Petrov et al., 2006). Next, we comment on the preprocessing tasks that are specific for each translation task.

- **English-to-German**: The German text was also true-cased and parsed with the provided grammar of BitPar (Schmid, 2004). German is morphologically rich and in order to avoid sparseness issues, we removed the functional and morphological annotation from the tags used in the parses.

- **English-to-Arabic**: The Arabic text was tokenized with MADA (Habash et al., 2009) and transliterated according to Buckwalter (2002). Since the Berkeley parser (Petrov et al., 2006) also provides a grammar for Arabic, we parsed the Arabic training data with it.

- **English-to-Chinese**: The Chinese sentences were word-segmented using the Stanford Word Segmenter (Chang et al., 2008). Again, the Berkeley parser (Petrov et al., 2006) with its provided grammar delivers the parse trees for the Chinese training data.

After the preprocessing steps, we obtained a word-aligned, bi-parsed parallel corpus, to which we applied the described rule extractions together with the baseline rule extractions provided by Moses (Koehn et al., 2007; Hoang et al., 2009). To give a quick overview, we report the number of extracted rules for all translation tasks and rule extractions in Table 2. We can immediately confirm that relaxing the conditions during rule extraction (e.g., from tree-to-tree to string-to-tree) greatly increases the number of extracted rules for each translation task. For example, the string-to-tree rule extraction for MBOTs (Seemann et al., 2015) extracts 12–17 times more rules than the minimal tree-to-tree rule extraction for MBOTs (Maletti, 2011). In addition, the availability of several target-side objects (and thus discontinuities) also leads to additional freedom during rule extraction, which is evidenced by the larger numbers of rules extracted for MBOTs compared to those extracted for the SCFG baseline. For our experiments we heavily

<table>
<thead>
<tr>
<th></th>
<th>English to German</th>
<th>English to Arabic</th>
<th>English to Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>training data</td>
<td>7th EuroParl (Koehn, 2005)</td>
<td>MultiUN (Eisele and Chen, 2010)</td>
<td></td>
</tr>
<tr>
<td>training data size</td>
<td>≈ 1.8M sentence pairs</td>
<td>≈ 5.7M sentence pairs</td>
<td>≈ 1.9M sentence pairs</td>
</tr>
<tr>
<td>language model</td>
<td>5-gram SRILM (Stolcke, 2002)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>add. LM data</td>
<td>WMT 2013</td>
<td>Arabic in MultiUN</td>
<td>Chinese in MultiUN</td>
</tr>
<tr>
<td>LM data size</td>
<td>≈ 5.7M sentences</td>
<td>≈ 9.7M sentences</td>
<td>≈ 9.5M sentences</td>
</tr>
<tr>
<td>tuning data</td>
<td>WMT 2013</td>
<td>cut from MultiUN</td>
<td>NIST 2002, 2003, 2005</td>
</tr>
<tr>
<td>tuning size</td>
<td>3,000 sentences</td>
<td>2,000 sentences</td>
<td>2,879 sentences</td>
</tr>
<tr>
<td>test data</td>
<td>WMT 2013 (Bojar et al., 2013)</td>
<td>cut from MultiUN</td>
<td>NIST 2008 (NIST, 2010)</td>
</tr>
<tr>
<td>test size</td>
<td>3,000 sentences</td>
<td>1,000 sentences</td>
<td>1,839 sentences</td>
</tr>
</tbody>
</table>

Table 1: Summary of the used resources.
Table 2: Number of extracted rules for the different rule extractions.

<table>
<thead>
<tr>
<th>System</th>
<th>Number of extracted rules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>English-To-German</td>
</tr>
<tr>
<td>tree-to-tree SCFG (baseline)</td>
<td>6,630,590</td>
</tr>
<tr>
<td>minimal tree-to-tree MBOT</td>
<td>12,478,160</td>
</tr>
<tr>
<td>tree-to-tree MBOT</td>
<td>40,736,687</td>
</tr>
<tr>
<td>string-to-tree SCFG (baseline)</td>
<td>14,092,729</td>
</tr>
<tr>
<td>string-to-tree MBOT</td>
<td>143,661,376</td>
</tr>
<tr>
<td>hierarchical SCFG (baseline)</td>
<td>406,433,344</td>
</tr>
<tr>
<td>string-to-string MBOT</td>
<td>1,084,007,782</td>
</tr>
</tbody>
</table>

5.2 Translation features

As usual, the task of the decoder is to find the best translation \( \hat{f} \) of the input object \( e \) (string or tree) licensed by the translation model and the language model.

\[
\hat{f} = \arg \max_f p(f | e) = \arg \max_f p(e | f) \cdot p(f)
\]

The probability \( p(f) \) is provided by a (string) 5-gram language model for the target language. Thus, if target syntax is used, then the yield (the sentence written on the frontier) of the tree \( f \) is used for language model scoring. The data used to train the language model and the used toolkit are reported in Table 1. The translation model provides the probability \( p(e | f) \) and uses either the MBOT model or the baseline SCFG model as implemented in Moses. More precisely, the translation model uses a log-linear model (Och, 2003) of weighted features \( h_k(\cdot)^{\lambda_k} \) over derivations \( D \) for the pair \((e, f)\).

\[
p(e | f) = \max_{D \text{ derivation for } (e, f)} p(D) = \max_{D \text{ derivation for } (e, f)} \left( \prod_i h_i(D)^{\lambda_i} \right),
\]

where \( h_i(\cdot) \) are features on derivations. The features of the derivation are usually derived as a product of the rule features of those rules that constitute the derivation. We used the following (mostly standard) rule features (Koehn, 2009):

- the forward and backward translation probabilities,
- the forward and backward lexical translation probabilities,
- the phrase and word penalty, and
- the gap penalty, which is specific for MBOTs.

The forward and backward translation probabilities are obtained as normalized relative frequencies. We applied Good-Turing smoothing (Good, 1953) to all rules that were extracted at most 10 times. Both lexical translation probabilities are obtained as usual, and the MBOT-specific gap penalty is defined as \( 100^{1-c} \), where \( c \) is the number of target objects used in all rules that contributed to \( D \). This feature is intended to allow the model to tune the amount of discontinuity to the specific target language. Indeed in all experiments the feature weights \( \lambda_i \) of the log-linear model were trained using minimum error rate training (Och, 2003).

The task of the decoder is the identification of the best-scoring target object \( f \) and derivation \( D \) in the above definition of \( p(e | f) \). In all our experiments a CYK+ chart parser is used as decoder. The decoder for the SCFG model is provided by the syntax component (Hoang et al., 2009) of the Moses framework, and the decoder for the MBOT model is provided by MbotMoses branch (Braune et al., 2013) of Moses.
<table>
<thead>
<tr>
<th>Setting</th>
<th>System</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>En-De</td>
</tr>
<tr>
<td>tree-to-tree</td>
<td>Moses (baseline)</td>
<td>14.50</td>
</tr>
<tr>
<td></td>
<td>minimal MBOT</td>
<td>14.09</td>
</tr>
<tr>
<td></td>
<td>non-minimal MBOT</td>
<td>14.41</td>
</tr>
<tr>
<td>string-to-tree</td>
<td>Moses (baseline)</td>
<td>14.96</td>
</tr>
<tr>
<td></td>
<td>MBOT</td>
<td><em>15.49</em></td>
</tr>
<tr>
<td></td>
<td>GHKM (Galley et al., 2004, 2006)</td>
<td>17.10</td>
</tr>
<tr>
<td>hierarchical</td>
<td>Moses (baseline)</td>
<td>17.00</td>
</tr>
<tr>
<td></td>
<td>MBOT</td>
<td>16.57</td>
</tr>
<tr>
<td></td>
<td>phrase-based Moses</td>
<td>16.80</td>
</tr>
</tbody>
</table>

Table 3: BLEU evaluation results for all 3 translation tasks. Starred results indicate statistically significant improvements over the baseline (at confidence $p < 1\%$).

### 5.3 Quantitative evaluation

In this section, we first compare all systems to each other using the score BLEU (Papineni et al., 2002). We also present the results obtained by systems that were high-ranked on public shared tasks (Bojar et al., 2014) such as phrase-based systems (Koehn et al., 2003) or string-to-tree systems obtained following Galley et al. (2004, 2006). All systems were tuned for BLEU on the tuning data, and we report the BLEU scores obtained by the tuned systems on the test sets. The MBOT-based systems were evaluated against their corresponding syntax component (Hoang et al., 2009) of the Moses toolkit, which implements tree-to-tree, string-to-tree, and string-to-string (hierarchical) rule extractions. All of them follow essentially the procedure outlined in Chiang (2005), which was also the basis for the rule extraction of Seemann et al. (2015) and our string-to-string rule extraction. Our implementation of the non-minimal tree-to-tree MBOT rule extraction is also an extension of the corresponding procedure of the syntax component of Moses. We also checked statistical significance for the MBOT results using the implementation of Gimpel (2011).

We performed large scale experiments on three major translation tasks, namely English-to-German (En-De), English-to-Arabic (En-Ar), and English-to-Chinese (En-Zh). The goal was to evaluate the following MBOT systems: (i) the minimal tree-to-tree system (Section 2.1), (ii) the non-minimal tree-to-tree system (Section 3), (iii) the non-minimal string-to-tree system (Section 2.2), and (iv) the string-to-string system (Section 4). The obtained results are reported in Table 3. Unfortunately, the rule table of the string-to-string MBOT for the English-to-Arabic translation task — although already filtered on the given input — was too large to load into main memory (available: 500GB RAM).

Let us now discuss the results for the various settings. Overall, we observe that the tree-to-tree systems perform worst. For the baseline system using SCFG rules (i.e., MBOT rules with a single tree on both the left- and right-hand side), this result is not surprising. Already, Ambati and Lavie (2008) have shown that tree-to-tree rules are too restrictive to achieve good lexical coverage. However, our results show that making rules more flexible by allowing several target trees hurts the performance instead of yielding improvements. This effect is particularly visible when only using minimal rules. On the English-to-Arabic and English-to-Chinese translation tasks, the minimal MBOT system loses 10.61 and 5.62 BLEU points, respectively, over the baseline. Interestingly, on the English-to-German translation task the loss is only 0.41 BLEU points. Adding non-minimal MBOT rules yields the expected large improvements, but is overall still not good enough to beat the tree-to-tree baseline. This result is interesting insofar as it
Table 4: Number of rules per type used when decoding test (Lex = lexical rules; Struct = structural rules; [dis]cont. = [dis]continuous).

do not confirm the results of Sun et al. (2009) on large scale experiments with target side discontinuities.3

The results for the string-to-tree setting are much better than those for the tree-to-tree systems (see Table 3). The BLEU score improvements are not very pronounced for English-to-German and English-to-Chinese, but on the English-to-Arabic translation task the string-to-tree systems (both baseline and MBOT) achieve huge improvements. Those systems come in at 4.74 and 5.61 BLEU points, respectively, ahead of the tree-to-tree baseline. As already demonstrated by Seemann et al. (2015), the MBOT system yields significant improvements over the baseline on all those language pairs. The GHKM systems achieve mixed results. They outperform the MBOT system on English-to-German, achieve the same performance as the MBOT system on English-to-Chinese, and lose even against the baseline on the English-to-Arabic translation task.

Finally, the string-to-string systems generally yield the best translation quality (as measured by BLEU). The experiments for the English-to-German and English-to-Chinese translation task show that our string-to-string MBOT system does not improve performance in these cases. Indeed, the analysis presented in Section 6 suggests that the string-to-string rules are flexible enough to achieve high coverage even without the need for multiple phrases in the right-hand side. This is slightly disappointing as Galley and Manning (2010) incorporated discontinuous phrases into a phrase-based system, and their evaluation on Chinese-to-English showed significant improvements over a standard phrase-based baseline as well as over a hierarchical baseline. However, the differences are generally not large, and even the phrase-based system achieves similar performance.

6 Analysis of Discontinuity

Another goal was to identify whether discontinuous rules are useful and to what extent these are useful. We try to estimate their impact on the translation quality by inspecting the statistics on the rules used in the derivations. Consequently, only rules that produce part of the final output in each of the translation tasks count. The current tools of MbotMoses (Braune et al., 2013) only allow the counting of rules used during decoding. At present, it is infeasible to track discontinuous objects through the derivation to decide whether they are actually assembled continuously or discontinuously. Thus, discontinuous rules only indicate a potential discontinuity.

---

3The experiments of Sun et al. (2009) report scores for the translation task Chinese-to-English for systems trained on 240,000 sentences only. Their model allows discontinuities on the source language side, which should be comparable to target-side discontinuities for the opposite translation direction English-to-Chinese.
Table 5: Number of rules per type used when decoding test (Lex = lexical rules; Struct = structural rules; [dis]cont. = [dis]contiguous).

<table>
<thead>
<tr>
<th>MBOT variant</th>
<th>Type</th>
<th>Lex</th>
<th>Struct</th>
<th>Total</th>
<th>Target tree fragments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>minimal t-to-t</td>
<td>cont.</td>
<td>18,389</td>
<td>2,855</td>
<td>21,244</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discont.</td>
<td>1,138</td>
<td>1,920</td>
<td>3,085</td>
<td>2,525</td>
</tr>
<tr>
<td>non-minimal t-to-t</td>
<td>cont.</td>
<td>9,826</td>
<td>1,581</td>
<td>11,407</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discont.</td>
<td>1,605</td>
<td>746</td>
<td>2,315</td>
<td>1,577</td>
</tr>
<tr>
<td>non-minimal s-to-t</td>
<td>cont.</td>
<td>1,839</td>
<td>651</td>
<td>2,490</td>
<td></td>
</tr>
<tr>
<td></td>
<td>discont.</td>
<td>3,670</td>
<td>1,324</td>
<td>4,994</td>
<td>3,008</td>
</tr>
</tbody>
</table>

Tables 4, 5, and 6 show the statistics on the rules used during decoding. Continuous rules (i.e., rules with a single object in the right-hand side) are abbreviated by *cont*, and (potentially) discontinuous rules are abbreviated by *discont*. To provide a deeper analysis, we also distinguish between lexical and structural rules. Lexical rules, abbreviated *Lex*, are rules that contain no exposed nonterminal symbols. Similarly structural rules, abbreviated *Struct*, are rules containing at least one such nonterminal symbol. Finally, we abbreviate the settings tree-to-tree, string-to-tree, and string-to-string by ‘t-to-t’, ‘s-to-t’, and ‘s-to-s’, respectively.

We first discuss the results for the tree-to-tree systems presented across Tables 4, 5, and 6. If we only use minimal MBOT rules, then 11% of the rules used during decoding are discontinuous in the English-to-German and English-to-Chinese translation tasks. The rate is slightly higher in the English-to-Arabic translation task (14.5%). For all the translation tasks, the majority of the discontinuous rules are structural. This fact is not very surprising since the leaves of the minimal tree-to-tree rules are either lexical items or exposed non-terminal occurrences. The minimality constraint encourages word-by-word translation, and once the lexical rules are excised, only structural rules remain. Based on the observed high BLEU score losses, it seems that minimal tree-to-tree rules are, at present, unable to correctly assemble discontinuous parts.

If we additionally use non-minimal tree-to-tree rules, then the rates of discontinuous rules change. For English-to-German the rate remains almost the same at 13%, whereas for the English-to-Arabic translation task it increases to 20%. Finally, for English-to-Chinese suddenly only 4% of the rules applied during decoding are discontinuous. The non-minimality encourages large rules, which are more likely to contain only lexical items. As expected, the number of discontinuous lexical rules is always larger than the number of discontinuous structural rules in this setting. This is particularly true for English-to-German and English-to-Arabic, where two third of the discontinuous rules are lexical, whereas the distribution is almost even for English-to-Chinese. We believe that these lexical discontinuous rules capture relevant idiomatic expressions or encode agreement or correspondences, which yield the large improvements in translation quality over minimal rules only.

For string-to-tree systems we only present the statistics since they have been discussed already by Seemann et al. (2015). It is evident that they generally use the largest amounts of discontinuous rules, which is rewarded with significant improvements over the baseline system without discontinuities.

Finally, for the string-to-string systems, the opposite situation presents itself. Here, the number of discontinuous rules is indeed marginal. On the English-to-German translation task only 1.1% of 33,962 rules are discontinuous. The English-to-Chinese system also only uses 2.3% discontinuous rules (out of 25,575 rules). We believe that the low use of discontinuous string-to-string rules can be explained by the absence of linguistic annotations. Without them,
Table 6: Number of rules per type used when decoding test (Lex = lexical rules; Struct = structural rules; [dis]cont. = [dis]contiguous).

the rules become very flexible, thus removing the need for discontinuous MBOT rules in this setting. Since the number of used discontinuous rules is so low, it can be assumed that essentially the same rules were used during decoding when comparing the MBOT system to the baseline. This would also explain their comparable BLEU scores.

7 Conclusion

We have extended the existing rule extraction techniques for shallow local multi bottom-up tree transducers to the two main missing settings. First, we designed a non-minimal tree-to-tree rule extraction for MBOT, which extends the corresponding rule extraction for minimal rules. Secondly, we developed a rule extraction for the string-to-string setting, which does not rely on syntactical information. Naturally, we also evaluated these new rule extractions together with several other systems in 3 large scale translation tasks (English-to-German, English-to-Arabic, and English-to-Chinese).

As expected, the non-minimal tree-to-tree system performs much better than the corresponding system using only minimal rules, but even the system with non-minimal rules does not beat the SCFG baseline (using non-minimal rules). It seems that discontinuity remains a challenge for tree-to-tree rules. Overall, tree-to-tree systems report the worst scores. For the string-to-tree systems already Seemann et al. (2015) report significant improvements in translation quality when using discontinuous rules. Finally, in the string-to-string (hierarchical) setting, discontinuous rules are hardly ever used, so when compared to the SCFG baseline essentially the same performance is obtained. Most likely, hierarchical rules are flexible enough to handle most common forms of discontinuity without the need to explicitly represent it in its rules. In summary, since MBOT offers certain consistent advantages across the different language pairs it may be useful to exploit a hybrid approach in the future.

To support further experimentation by the community, we publicly release our developed software and analysis tools (http://www.ims.uni-stuttgart.de/forschung/ressourcen/werkzeuge/mbotmoses.en.html).

Acknowledgement

The authors would like to thank the reviewers for their helpful comments and suggestions.

All authors were financially supported by the German Research Foundation (DFG) grant MA 4959 / 1-1, which we gratefully acknowledge.
References


Register-Based Machine Translation Evaluation with Text Classification Techniques

Mihaela Vela  
Saarland University, Saarbrücken, Germany  
m.vela@mx.uni-saarland.de

Ekaterina Lapshinova-Koltunski  
Saarland University, Saarbrücken, Germany  
e.lapshinova@mx.uni-saarland.de

Abstract
This paper presents a novel approach to machine translation evaluation by combining register features – characterised by particular distributions of lexico-grammatical features – with text classification techniques. The goal of this method is to compare machine translation output with comparable originals in the same language, as well as with human reference translations. The degree of similarity – in terms of register features – between machine translations and originals, and machine translations and reference translations is measured by applying two text classification methods trained on 1) originals and 2) reference translations, and tested on machine translations. The results from the experiments prove our assumption that machine translations share register features rather with human translations than with non-translated texts produced by humans. This confirms that registers are one of the most important factors that should be integrated into register-based machine translation evaluation.

1 Introduction: Motivation and Goals
The state-of-the-art in evaluating machine translation (MT) nowadays is to measure lexical and eventually syntactic and semantic overlap between a machine translation (called hypothesis translation) and a human-produced reference translation. In this paper, we present a new approach to evaluation, integrating the knowledge on register, i.e. language variation according to context, as defined by Halliday and Hasan (1989) and Quirk et al. (1985). The difference in terms of register between original and translated texts has been shown by several studies (Hansen-Schirra et al., 2012; Kruger and van Rooy, 2012; Neumann, 2013), proving that translations tend to share a set of lexical, syntactic and/or textual features. More recent investigations by Baroni and Bernardini (2006), Kurokawa et al. (2009) and Lembersky et al. (2012) applied text classification methods to automatically identify these differences.

The aim of the research presented here is twofold: 1) to show that machine translations and the corresponding reference translations are related to each other in terms of register-specific features and as a consequence of this 2) to show that hypothesis translations and human translations share more than the lexical surface. The novel idea introduced here is the notion of register-specific features which relate reference and hypothesis translations, and therefore have implications for MT (evaluation).

We measure the “closeness” between comparable non-translated originals and machine translations, as well as between human and machine translations by applying two different classification methods. The classification is performed on the basis of extracted register-specific features for two data sets. First, we use original non-translated texts as training data and ma-
chine translations as test data. In a second step, human translations are used as training data and machine translations as test data. Our assumption is that in terms of register specificity quantified in the corresponding features, MT output is closer to the corresponding reference translations than to the comparable non-translated originals. We base our hypothesis on the fact that b) translations tend to normalise towards target language conventions and that a) machine translations will adapt more to these conventions than to source texts (Diwersy et al., 2014).

The remainder of the paper is structured as follows. In Section 2, we present related work from the areas of machine translation evaluation and register theory. In Section 3 we present our research questions, describe the selected features and resources used for the experiments, as well as the applied methods. Section 4 demonstrates the results of our analysis, and in Section 5, we discuss the outcome and give an outlook on further analyses.

2 Related Work

2.1 Machine Translation Evaluation

State-of-the-art MT evaluation applies automatic language-independent metrics such as BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) in order to compare MT output (hypothesis translation) with one or more human translations (reference translations). Several studies (Callison-Burch et al., 2006; Vela et al., 2014a,b) have confirmed the fact that BLEU scores should be treated carefully, thus advancing the development of new metrics. New evaluation metrics such as METEOR (Denkowski and Lavie, 2014), Asiya (González et al., 2014) and VERTa (Comelles and Atserias, 2014), are incorporating lexical, syntactic and semantic information into their scores, whereas metrics like BEER (Stanojević and Sima'an, 2014), ReVal (Gupta et al., 2015) and COMET (Vela and Tan, 2015) use machine learning approaches for MT evaluation. The accuracy of the newly introduced evaluation methods is usually proven by human evaluation inputs, more specifically by measuring the correlation of the automatically provided scores with human judgements. Human evaluation is realised by ranking MT outputs (Bojar et al., 2013, 2014; Vela and van Genabith, 2015). In addition, post-editing, which is mainly used for measuring productivity (Guerberof, 2009; Zampieri and Vela, 2014), is also a valid human evaluation method.

2.2 Main Notions within Register Theory

Studies related to register theory (Quirk et al., 1985; Halliday and Hasan, 1989; Biber, 1995) are concerned with contextual variation of languages, and state that languages vary with respect to usage context within and across languages. For example, languages may vary according to the activity of the participants involved or the relationship between speaker and addressee(s). These parameters correspond to the variables of (1) field, (2) tenor and (3) mode defined in the framework of systemic functional linguistics (SFL), which describes language variation according to situational contexts; see, for instance, Halliday and Hasan (1989) and Halliday (2004). These variables are associated with the corresponding lexico-grammatical features, e.g. field of discourse is realised in term patterns or functional verb classes, e.g. activity (approach, supply, etc.), communication (answer, inform, suggest, etc.) and others; tenor is realised in modality expressed e.g. by modal verbs (can, may, must, etc.) or stance expressions (used by speakers to convey personal attitude to the given information, e.g. adverbs like actually, certainly, amazingly, importantly, etc.); and mode is realised in information structure and textual cohesion, e.g. coreference via personal (she, he, it) and demonstrative (this, that) pronouns. Thus, differences between registers can be identified through the analysis of occurrence of lexico-grammatical features in these registers; see Biber’s studies on linguistic variation, e.g. Biber (1988), Biber (1995) or Biber et al. (1999). The field of discourse also includes experiential domain realised in the lexis. This corresponds to the notion of domain used in the machine
translation community. However, it also includes colligation (morpho-syntactic preferences of words), in which grammatical categories are involved. Thus, domain is just one of the parameter features a register can have.

### 2.3 Register in Translation

Several studies in systemic functional linguistics are concerned with register settings in human translation (Steiner, 2004; Hansen-Schirra et al., 2012; De Sutter et al., 2012; Neumann, 2013; House, 2014) and their application into translation practice (Vela and Hansen-Schirra, 2006; Vela et al., 2007). To our knowledge, the machine translation (including its evaluation) community has not yet taken into consideration the notion of register, at least according to the definition in the present paper. Studies in the field of MT concerned with translation errors of new domains are covering only the lexical level (Irvine et al., 2013), as the authors operate solely with the notion of domain (field of discourse) and not register (which includes more parameters, as described in Section 2.2 above). Research on adding in-domain bilingual data to the training material of SMT systems (Eck et al., 2004; Wu et al., 2008) or on application of in-domain comparable corpora (Laranjeira et al., 2014; Irvine and Callison-Burch, 2014) consider the notion of domain. However, further register features are mostly ignored.

Domains reflect what a text is about, i.e. its topic. So, consideration of domain alone would classify news reporting on certain political topics together with political speeches discussing the same topics, although they belong to different registers. We expect that texts from the latter (political speeches) translated with a system trained on the former (news) would be lacking in persuasiveness, argumentation and other characteristics reflected in their lexico-grammatical features, e.g. imperative verbal constructions used to change the addressee’s opinion, or interrogatives as a rhetorical means, etc. The similarity in domains would cover only the lexical level, in most cases terminology, ignoring the lexico-grammatical patterns specific for the given register, as shown by Lapshinova-Koltunski and Pal (2014) in their discussion on domain vs. register. Although some NLP studies employing web resources are arguing for the importance of register conventions, as by Santini et al. (2010), register remains out of the focus of machine translation. One of the few works addressing the relevance of register features for machine translation is Petrenz (2014), in which the author uses text features to build cross-lingual register classifiers.

### 3 Methodology and Resources

#### 3.1 Research Questions

Following the assumption that translated language should normalise the linguistic features (like those described in Section 2.2 above) in order to adapt them to target language conventions, we use two different classification methods, KNN and SVM, to prove that in terms of register settings 1) machine translations correspond to human reference translations to a greater extent than 2) to comparable original non-translated texts in the same language. This requires a two-fold experiment design. In the first experiment, we use German original data for training and German machine translations for testing. Based on classification accuracy we can determine the “closeness” of machine translations to comparable non-translated texts in the same language. For the second experiment, we use a different data set applying the same classification methods. Human reference translations are used as training data and machine translations as test data. In this way we can observe the relation between machine translations and comparable non-translated texts in the same language, as well as between machine translations and reference translations.

We also aim at answering the following questions:
(i) Do German machine translations correspond to comparable German non-translated originals?

(ii) Are German machine translations closer to human reference translations than to comparable original German texts?

(iii) What are the main parameters influencing the classification outcome?

Our assumption is that machine translations will comply more with register standards of human-produced translated texts rather than with non-translated texts written by humans, as it was shown by Lapshinova-Koltunski and Vela (2015).

3.2 Feature Selection

For our analysis, we select a set of features derived from register studies described in Section 2.2 above. These features represent lexico-grammatical patterns of more abstract concepts, i.e. textual cohesion expressed via pronominal coreference or other cohesive devices, evaluative patterns (e.g. *it is interesting*/*important that*, etc.) and others. The selected features reflect linguistic characteristics of all texts under analysis, are content-independent (do not contain terminology or keywords), and are easy to interpret yielding insights on the differences between the analysed variables. For instance, we use groupings of specific types of phrases (e.g. nominal, verbal, etc.) instead of part-of-speech n-grams, as they are easier to interpret as n-grams. The set of selected features for our analysis is outlined in Table 1. The first column denotes the extracted and analysed patterns, the second represents the corresponding linguistic features, and the third denotes the three context parameters according to register theory as previously described in Section 2.2.

The number of nominal and verbal parts-of-speech, chunks and nominalisations (*ung*-nominalisations) reflects participants and processes in the field parameter. The distribution of abstract or general nouns and their comparison to other nouns gives information on the vocabulary (parameter of field). Modal verbs grouped according to different meanings (Biber et al., 1999), and evaluation patterns express modality and evaluation, i.e. the parameter of tenor. Content words and their proportion to the total number of words in a text represent lexical density, which is an indicator of the parameter of mode. Conjunctions, for which we analyse distributions of logico-semantic relations, belong to the parameter of mode as they serve as discourse-structuring elements. Reference expressed either in nominal phrases or in pronouns reflects textual cohesion (mode). Overall, we define 21 features representing subtypes of the categories given in Table 1.

<table>
<thead>
<tr>
<th>pattern</th>
<th>feature</th>
<th>parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>nominal and verbal chunks</td>
<td>participants and processes</td>
<td>field</td>
</tr>
<tr>
<td><em>ung</em>-nominalisations and general nouns</td>
<td>vocabulary and style</td>
<td></td>
</tr>
<tr>
<td>modals with the meanings of permission, obligation, volition</td>
<td>modality</td>
<td>tenor</td>
</tr>
<tr>
<td>evaluative patterns</td>
<td>evaluation</td>
<td></td>
</tr>
<tr>
<td>content vs. functional words</td>
<td>lexical density</td>
<td>mode</td>
</tr>
<tr>
<td>additive, adversative, causal, temporal, modal conjunctive relations</td>
<td>logico-semantic relations</td>
<td></td>
</tr>
<tr>
<td>3rd person personal and demonstrative pronouns</td>
<td>cohesion via reference</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Features under analysis
However, for the final interpretation, a reduced number of features is used, which results from the validation step described in Section 3.4.1 below.

3.3 Data

In the first experimental setting, the training data set consists of German non-translated texts (GO=German originals) extracted from CroCo (Hansen-Schirra et al., 2012), a corpus of both parallel and comparable texts in English and German. The dataset contains 108 texts which cover seven registers: political essays (ESSAY), fictional texts (FICTION), manuals (INSTR), popular-scientific articles (POPSCI), letters to shareholders (SHARE), prepared political speeches (SPEECH), and tourism leaflets (TOU). The decision to include this wide range of registers is justified by the need for heterogeneous data for our experiment. Therefore, the dataset contains both frequently machine-translated texts, e.g. SPEECH, ESSAY and INSTR, and those, which are commonly not translated with MT systems, such as FICTION or POPSCI. The total number of tokens in GO is 252711.

The corresponding test data set is smaller and includes 50 texts translated from English into German with a rule-based (RBMT) and a statistical (SMT) machine translation system. The rule-based machine translations were produced with the rule-based system SYSTRAN6 (Systran, 2001). The statistical machine translations were produced with a Moses-based system1, trained with EUROPARL (Koehn, 2005), a parallel corpus containing texts from the proceedings of the European parliament. The total number of tokens in RBMT and SMT comprise 127865 and 124462 respectively. Both variants contain translations of the same texts belonging to the same registers as in the originals (training data).

In the second setting, English to German human translations (HT) – extracted from the CroCo corpus – are used for training, and comprise 100 texts (262655 tokens). For testing we use the same data as in the previous setting – machine translated texts (RBMT and SMT). Both training and test data are translations of the same source texts, which corresponds a common setting in MT evaluation. Table 2 gives an overview of the number of texts, sentences and tokens in both experiment settings.

<table>
<thead>
<tr>
<th></th>
<th>Setting 1</th>
<th></th>
<th>Setting 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>Texts</td>
<td>GO</td>
<td>RBMT</td>
<td>SMT</td>
<td>HT</td>
</tr>
<tr>
<td></td>
<td>108</td>
<td>50</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Sentences</td>
<td>15736</td>
<td>6195</td>
<td>6131</td>
<td>13077</td>
</tr>
<tr>
<td>Tokens</td>
<td>252711</td>
<td>127865</td>
<td>127865</td>
<td>262655</td>
</tr>
</tbody>
</table>

Table 2: Statistics for the data sets used in experiments

To extract the occurrences of register features described in Section 3.2, we annotate both training and test sets with information on token, lemma, part-of-speech, syntactic chunks and sentence boundaries using Tree Tagger (Schmid, 1994). The availability of these annotation levels in both corpora allows us to analyse certain lexico-grammatical patterns (see Section 3.2) required for register-sensitive analysis of translation. The features are then defined as linguistic patterns and modelled as regular expressions for the Corpus Query Processor (Evert, 2005), available within the CWB tools (CWB, 2010).

3.4 Classification Approaches

For our classification task, we train two models by using two different classifiers for each experiment setting: $k$-nearest-neighbors (KNN), a non-parametric method, and support vector

---

1We did not perform tuning for the SMT system.
machines (SVM) with a linear kernel, a supervised method, both commonly used in text classification. The models are then tested on German translations, and in this way, we obtain classification performance scores for:

- KNN and SVM, having German originals as training data and machine translations (English-German) as test data;
- KNN and SVM, having human translations (English-German) and again machine translations (English-German) as test data².

The performance scores are then judged in terms of precision, recall and f-measure. These scores are class-specific and indicate the results of automatic assignment of register labels to certain machine-translated texts. In case of precision, we measure the class agreement of the data with the positive labels given by the classifier. For example, there are ten German fictional texts in our data. If the classifier assigns FICTION labels to ten texts only, and all of them really belong to FICTION, then we will achieve the precision of 100%. With recall, we measure if all translations of a certain register were assigned to the register class they should belong to. So, if we have 10 fictional texts, we would have the highest recall if all of them are assigned with the FICTION label. F-measure combines both precision and recall, and is understood as the harmonic mean of both.

3.4.1 K-Nearest Neighbors (KNN)

When using KNN, the input consists of the K closest training examples in the feature space³, and the output is a class membership. This method is instance-based, where each instance is compared with existing ones using a distance metric, and the distance-weighted average of the closest neighbours is used to assign a class to the new instance, see (Aha et al., 1991; Witten et al., 2011).

For our experiments we have to determine the final number for K and the most appropriate number of features used for classification. By measuring the distribution of errors during training (by performing 10-folds cross-validation) we determined the best K=11⁴ and the final number of features for our setting. For our classification analysis we work with the tuple (numberOfFeatures=17, K=11), performing classification on the German translation (test) data by using the knn library in Weka (Hall et al., 2009). The final list of features include:

→ total words – words per text
→ content words – content (lexical) words
→ NP chunks – nominal chunks
→ VP chunks – verbal chunks
→ chunks – chunks per text
→ nominal – nominal part-of-speech categories
→ verbal – verbal part-of-speech categories
→ adversative – adversative conjunctive relations
→ causal – causal conjunctive relations
→ temporal – temporal conjunctive relations
→ modal – modal conjunctive relations
→ ung-nominalisations – nominalisations formed with ung-suffix
→ pron – personal pronouns
→ dempron – demonstrative pronouns
→ pronp – nominal phrases filled with pronouns
→ gnouns – general nouns
→ modals denoting permission

²Note that human and machine translations are translation variants of the same English source text.
³In our experiments, the features are quantified by their frequency in the corresponding data set.
⁴The final value for K was chosen from an interval between 3 and 19
3.4.2 Support Vector Machines (SVM)

When using SVM models (Vapnik and Chervonenkis, 1974), the learning algorithm tries to find the optimal boundary between classes by maximising the distance to the nearest training data of each class. Given German labelled training data, the algorithm outputs an optimal hyperplane which categorises new instances, here German translations\(^5\). We use the same list of features for the classification with SVM as in the classification with KNN. We perform SVM classification with a 10-fold cross-validation.

The cross-validation in the training phase has shown that registers SPEECH and SHARE show low accuracy, especially in the first experiment setting (when German original data is used). For this reason, we we exclude these registers from further analyses.

3.5 Experimental Setup

In the first setting, both classifiers are supposed to store all cases from German originals (108 data points) with the corresponding register labels available. New cases from test data, which are machine translations in this case (50+50 data points), are then classified, i.e. assigned register labels. Classification is performed on the basis of distance function measure, for which Euclidean distance is used. The results of automatic assignment are indicated with scores (precision, recall and f-measure), which are used to measure the class agreement of the data with the positive labels given by the classifiers.

In the second setting, both classifiers store all labelled cases available in human translations (100 data points). The trained model is then applied on the same machine translations as in the first setting (50+50 data points). And again, we use precision, recall and f-measure to judge the class agreement of the data with the positive labels given by the classifiers.

4 Classification Results

As already mentioned above, the results of both classification algorithms are analysed in terms of precision, recall and f-measure. In case of precision, we measure the class agreement of the data with the positive labels given by the classifier. Our assumption is that precision values would indicate if the test data correspond to the training data in terms of the register settings. Hence, in the first experiment setting, the higher the precision, the better a machine translation corresponds to comparable originals, whereas in the second setting, the higher the precision, the better a machine translation corresponds to human translations.

4.1 Setting 1: Originals vs. Machine Translations

<table>
<thead>
<tr>
<th></th>
<th>ESSAY</th>
<th>FICTION</th>
<th>INSTR</th>
<th>POPSCI</th>
<th>TOU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KNN</td>
<td>SVM</td>
<td>KNN</td>
<td>SVM</td>
<td>KNN</td>
</tr>
<tr>
<td>RBMT</td>
<td>0.36</td>
<td>0.00</td>
<td>0.86</td>
<td>0.75</td>
<td>0.17</td>
</tr>
<tr>
<td>SMT</td>
<td>0.48</td>
<td>0.00</td>
<td>0.86</td>
<td>0.80</td>
<td>0.33</td>
</tr>
</tbody>
</table>

Table 3: F-measure scores for classification per machine translation variant and register in the first setting.

Table 3 provides the confusion matrix for the five registers and the two MT outputs split on the two classification methods. Concerning both the classification method and the MT system, we notice that FICTION is the register performing best, achieving an f-measure of 0.86% for the combination RBMT-SVM and SMT-SVM. Based on the f-measure we observe that

---

\(^5\)One of the reasons why SVM are often used is their robustness towards overfitting, as well as their ability to map to a high-dimensional space.
ESSAY performs well for the classification with KNN but fails with SVM. INSTR and TOU perform similar in terms of f-measure for both KNN and SVM, the lowest f-measure values being measured for INSTR in the combination RBMT-KNN. POPSCI fails for the combinations RBMT-SVM and SMT-SVM, but performs well for KNN (0.53% for RBMT and 0.46% for SMT). These results imply that overall, FICTION performs best for both translation types and both classifiers, which indicates that translated fictional texts obey best to original fictional texts in terms of register settings.

Obviously, the data used for developing/training MT systems as well as the classification method play an important role. Registers like ESSAY (political essays) and INSTR (manuals) are usually used for training SMT systems, whereas registers like FICTION (fictional texts) and TOU (tourism leaflets) are less likely used for training MT systems. The more surprising are here the results for FICTION. The big difference in the results for both classifiers can be explained in the distinction between these two classification techniques. KNN uses all training data (predefined neighbours) in classification, while for SVM the maximised distance (margin) to the nearest example of each class plays a crucial role (all non-support vectors being discarded), thus influencing the difference in the classification outcome.

The overview of the performance of both MT systems in Table 4 (split on register and classification method) reveals that SMT performs better than RBMT in certain combinations for the registers ESSAY, FICTION and INSTR. The classification failure of the RBMT and SMT produced POPSCI translations for SVM indicates that SVM is not the appropriate classification method for this kind of texts. The fact that we observe contradictory results with both classifiers for ESSAY and POPSCI prevents us to claim that certain registers are generally more difficult to be identified in translated data than others.

<table>
<thead>
<tr>
<th>register</th>
<th>KNN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESSAY</td>
<td>SMT</td>
<td>RBMT=SMT</td>
</tr>
<tr>
<td>FICTION</td>
<td>RBMT=SMT</td>
<td>SMT</td>
</tr>
<tr>
<td>INSTR</td>
<td>SMT</td>
<td>SMT</td>
</tr>
<tr>
<td>POPSCI</td>
<td>RBMT</td>
<td>RBMT=SMT</td>
</tr>
<tr>
<td>TOU</td>
<td>RBMT</td>
<td>RBMT</td>
</tr>
</tbody>
</table>

Table 4: Performance of KNN and SVM across registers based on f-measure in the first setting

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBMT</td>
<td>0.43</td>
<td>0.61</td>
<td>0.50</td>
</tr>
<tr>
<td>SMT</td>
<td>0.50</td>
<td>0.61</td>
<td>0.54</td>
</tr>
</tbody>
</table>

Table 5: Classification accuracy per machine translation variant in the first setting

In the last step, we analyse the performance of the MT systems disregarding registers, see Table 5. We notice that results do not differ much in the MT systems under analysis, which means that register agreement between non-translated and machine-translated texts is not dependent on the method involved in translation. This is proven by Pearson’s chi-square test, which confirms our observation, as for both the KNN and the SVM results, p-value is higher than 0.05 (0.85 and 0.58 respectively).

### 4.2 Setting 2: Reference Translations vs. Machine Translations

The same experiments and analysis steps as in Section 4.1 are performed for the second setting, where human reference translations are used as training data and hypothesis translations as test
data. However, we observe a strong improvement in the results presented in Table 6. The registers ESSAY and FICTION achieve the best performance, showing an f-measure of up to 100%. Over all, f-measure remains over 50% for all registers, which means that classifiers perform also well for TOU, POPSCI and INSTR. These results can, in fact, serve as indicators that human and machine translations have more similarities, sharing register features.

<table>
<thead>
<tr>
<th>register</th>
<th>ESSAY</th>
<th>FICTION</th>
<th>INSTR</th>
<th>POPSCI</th>
<th>TOU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KNN</td>
<td>SVM</td>
<td>KNN</td>
<td>SVM</td>
<td>KNN</td>
</tr>
<tr>
<td>RBMT</td>
<td>0.93</td>
<td>1.00</td>
<td>1.00</td>
<td>0.54</td>
<td>0.71</td>
</tr>
<tr>
<td>SMT</td>
<td>0.93</td>
<td>1.00</td>
<td>0.50</td>
<td>0.62</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 6: F-measure scores for classification per machine translation variant and register in the second setting

In this setting, ESSAY matches register features of human translations best, leading to the assumption that the texts used for developing and training MT systems play a key role. In contrast to the results in the first setting, the difference between the results of the two classification methods for ESSAY is minor. The lowest value is scored by INSTR with an f-measure of 0.54 for RBMT-KNN.

Table 7 demonstrates that, different from the results in the first experiment in Section 4.1, RBMT performs better than SMT when compared to human reference translations, with some exceptions. RBMT-INSTR and RBMT-TOU are better classified than the same registers translated with the SMT system, if the results from KNN are taken into account.

<table>
<thead>
<tr>
<th>register</th>
<th>KNN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESSAY</td>
<td>RBMT=SMT</td>
<td>RBMT=SMT</td>
</tr>
<tr>
<td>FICTION</td>
<td>RBMT</td>
<td>RBMT</td>
</tr>
<tr>
<td>INSTR</td>
<td>SMT</td>
<td>RBMT</td>
</tr>
<tr>
<td>POPSCI</td>
<td>RBMT</td>
<td>RBMT</td>
</tr>
<tr>
<td>TOU</td>
<td>SMT</td>
<td>RBMT</td>
</tr>
</tbody>
</table>

Table 7: Performance of the classification methods across registers based on the f-measure in the second setting

Similar tendencies are observed if we average the scores for registers.

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN</td>
<td>SVM</td>
<td>KNN</td>
</tr>
<tr>
<td>RBMT</td>
<td>0.81</td>
<td>0.75</td>
</tr>
<tr>
<td>SMT</td>
<td>0.82</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table 8: Classification accuracy per translation variant in the second setting

We observe in Table 8 that, in terms of MT system, RBMT performs almost always better than SMT, and in terms of classification method applied, SVM performs always better than KNN. However, significance test show that the difference between both machine translation systems is not significant, as the computed p-values exceed 0.05 for both KNN and SVM scores (0.78 and 0.97 respectively), confirming not only our assumption that reference and hypothesis translations are similar if register features are considered, but also that both machine-translated variants are similar as well.
5 Conclusion

The results of the presented experiments have proven our assumption that machine translations share their register features rather with human produced translations than with human produced non-translated texts, regardless the method involved in translation.

**Setting 1: Original vs. Machine Translation**  The results of the first experiment show that register settings of most German machine translations do not comply with the register settings of non-translated German, this being shown also by Lapshinova-Koltunski and Vela (2015). This is especially valid for ESSAY and POPSCI, which show low performance in most classification scenarios, not adapting the target language register conventions. The good performance for FICTION is an indicator that fictional texts adapt to the conventions of the target language. However, as known from Neumann (2013), German and English fictional texts share a lot of features, which might also influence the performance of the classifiers. We suppose that the influence of the source language register conventions might also have an impact on the outcome. To prove this, we would need to perform additional experiments, in which English source texts should be used as training data.

Furthermore, as the performed significance tests have shown that the difference between RBMT and SMT is not meaningful, our suggestions apply regardless the translation method involved. In case of SMT, the results are apparently influenced by the training data used, as classification performs better for registers which are commonly used for SMT training, i.e. like ESSAY and INSTR. Off-the-shelf RBMT systems, like the one used here, are developed to cover a more general aspect of language which also essentially complicate the adaptation of register features.

**Setting 2: Reference vs. Machine Translation**  The results of the second experiment presented in Section 4 have proven our assumption that the overlap between hypothesis and reference translations is higher than between hypothesis translations and comparable non-translated texts. On the one hand, this corresponds to our intuition in Lapshinova-Koltunski and Vela (2015), where we show that both human and machine translations do not correspond to comparable German originals, suggesting that both machine and manual translations should have more in common. In fact, this is also shown by Lembersky et al. (2012), who demonstrate that the BLEU score can be improved if they apply language models compiled from translated texts and not non-translated ones. They also show that language models trained on translated texts fit better to reference translations in terms of perplexity. In fact, this confirms our claim that machine translations comply more with translated rather than with non-translated texts produced by humans. This results in the improvement of the BLEU score, but not necessary leading to a better quality of machine translation.

Following the results from both experimental setting, we argue that register features should be integrated into MT evaluation process, as an additional layer to the already existing automatic metrics. As future work, we would like to test this hypothesis by combining and correlating the results presented here with state-of-the-art evaluation metrics.

**References**


---

6Here, we mean “shining through” of the source language as defined by Teich (2003).


Error-Tolerant Speech-to-Speech Translation

Rohit Kumar  
Sanjika Hewavitharana  
Nina Zinovieva  
Matthew E. Roy  
Edward Pattison-Gordon  
Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA, USA, 02138

Abstract
Recent efforts to improve two-way speech-to-speech translation (S2S) systems have focused on developing error detection and interactive error recovery capabilities. This article describes our current work on developing an eyes-free English-Iraqi Arabic S2S system that detects ASR errors and attempts to resolve them by eliciting user feedback. Here, we report improvements in performance across multiple system components (ASR, MT and error detection). We also present a controlled evaluation of the S2S system that quantifies the effect of error recovery on user effort and conversational goal achievement.

1. Introduction
Over the past decade, considerable progress has been made in developing usable, two-way speech-to-speech (S2S) translation systems that enable real time cross-lingual spoken communication [1][2]. Conventionally, S2S systems comprise a pipeline of three speech and language technology components: automatic speech recognition (ASR), machine translation (MT) and text-to-speech synthesis (TTS).

While each of these components technologies have continued to improve in performance, each is data-driven and its performance will degrade when faced with novel vocabulary. For example, large-vocabulary ASR systems are incapable of recognizing out-of-vocabulary (OOV) words, MT systems cannot translate unseen source words and TTS often mispronounces novel words (often high-value concepts like names and technical terminology). A combination of these deficiencies can render S2S systems unusable, especially on conversational topics not well represented in the training data.

Different approaches to addressing component failures in the context of S2S systems have been explored. There have also been attempts at joint optimization of ASR and MT, as well as MT and TTS [23][24][25]. Most recently, S2S system that actively detect and recover from failures have been built and evaluated [16][26]. In this work, we report our recent efforts to extend and enhance the usability of S2S translation systems through active error detection and recovery.

To address system robustness, we have operationalized recent advances in ASR and statistical MT (SMT) into our real-time S2S system. These advances have dramatically improved ASR and SMT performance, leading to a more robust S2S pipeline, and yet they are fast enough for real-time use. To address the problem of novel vocabulary, we have developed interfaces that enable non-technical users to rapidly enrich our S2S system with new words and phrases that are relevant to emerging use cases. Finally, we detect potential ASR errors before they are amplified by being sent through the SMT and TTS components. A simple in-
teraction strategy then allows users to correct the translation in the event of an error. Section 3 evaluates an end-to-end S2S system augmented with this capability.

2. BBN’s Speech-to-Speech Translation System

2.1. Architecture

Two-way S2S systems comprise a pair of symmetric unidirectional pipelines of ASR, SMT and TTS components. Each pipeline serves as a communication channel in one direction. Figure 1 depicts one of these pipelines for our S2S system. Spoken input is converted by the ASR into a word lattice and a 1-best, whole-sentence hypothesis. The SMT produces the target language translation of the 1-best ASR hypothesis. The ASR error detector (AED) analyzes the word lattice to identify and rank potentially erroneous spans in the ASR hypothesis. This analysis, along with the current discourse state, drives a rule-based action selection module, which decides whether the translation should be transmitted to the listener or a clarification prompt should be presented to the speaker.
Our English-Iraqi Arabic S2S system uses ASR, AED and SMT components developed at BBN and off-the-shelf English and Iraqi Arabic TTS. The system is deployed on an off-the-shelf mobile computing device shown in Figure 2. The system is self-contained and runs the entire pipeline fully on this device. To enable eyes-free use, the platform is augmented with a pair of proprietary audio I/O devices, also shown in Figure 2. Each device, referred to as a “SuperMic”, offers a push-to-talk button, a high-quality close-talking microphone and a speaker.

2.2. System Components

**Automatic Speech Recognition:** The baseline ASR system is built using data from the DARPA TransTac English-Iraqi Arabic parallel two-way spoken dialogue collection [3]. It is based on the BBN Byblos system, which uses a multi-pass decoding strategy where models of increasing complexity are used in successive passes in order to refine the recognition hypotheses [4]. Speech is modeled as the output of context-dependent phonetic hidden Markov models (HMMs), whose outputs are mixtures of multi-dimensional diagonal Gaussians. Byblos uses various forms of parameter tying, including state tied mixture (STM) triphone models and state clustered tied mixture (SCTM) quinphone models. The models were trained on a set of acoustic features, part of which were obtained by using neural networks as described ahead.

Acoustic features: We use neural networks (NNs) to generate stacked bottleneck (SBN) features [5][6]. The SBN structure contains two NNs. The input features of the first NN are 24 critical-band energies obtained with a Mel filter-bank, with online mean and variance normalization applied. 15 frames of these features are stacked and a Hamming window multiplies the time evolution of each parameter. Finally, DCT is applied, of which 0th to 15th coefficients are retained. The first NN consists of two hidden layers, each with 1,500 nodes, followed by a bottleneck layer. The bottleneck (BN) outputs from the first NN are stacked, down-sampled, and taken as an input vector for the second NN. This second NN is also a bottleneck layer that is roughly the same size as the first NN. Both NNs were trained to classify phoneme states (5 states per phoneme). These targets were generated by forced alignment with baseline, perceptual linear prediction (PLP) models and remained fixed during the training. The final feature stream was built by concatenation of 9 frames of the PLP features together with the bottleneck layer output from the second NN. Finally, region dependent transformation (RDT) [7] is performed to estimate a discriminative feature projection to reduce the dimensionality to 46.

The English acoustic model was trained on approximately 200 hours of transcribed English speech from TransTac data, and the Iraqi Arabic acoustic model was trained on about 600 hours of transcribed speech from TransTac data. We tested ASR performance on held-out development sets, consisting of 13,074 and 3,354 utterances for Iraqi Arabic and English, respectively. Word error rate (WER) on these sets is 16.7% and 7.9%, respectively. For both languages, the WER is better than our previously reported ASR performance in this application [15], attributable to recent implementation of SBN features within Byblos.

**Statistical Machine Translation:** Our SMT system is based on the state-of-the-art, string-to-string hierarchical decoder described in [8]. In addition to the log-linear combination of generative components such as forward and backward rule probabilities, lexical translation probabilities, n-gram language models, etc., this system further incorporates two multi-layer NN scoring components, viz. the neural network joint model (NNJM) and the neural network lexical translation model (NNLTM). The NNJM estimates the probability of a hypothesized target word $t$, conditioned on both the n-1 preceding target words and an m-word source context,
centered at the source word s that t is affiliated with. The NNLTM estimates the probability that a source word s translates to a target word t (or NULL if none), given only t’s source context.

These models are trained with one hidden layer, and therefore leverage the robust performance of multi-layer neural networks, but, following [8], we achieve look-up speeds on a par with n-gram language models by precomputing the trained hidden layer (avoiding feed-forward computations at run-time) and by training the model to produce approximately normalized output (avoiding costly softmax computations across the whole target vocabulary at run-time). To further speed up the decoder, we do not perform n-best hypothesis re-ranking, and we also tighten up various parameters in the decoder’s beam search, leading to average decoding speeds of approximately 41 words/second.

We trained both our English-to-Iraqi Arabic (E2I) and Iraqi Arabic-to-English (I2E) systems on the Iraqi-English parallel text portions of the DARPA TransTac corpus [3], which we word-aligned using GIZA++ [9] and extracted hierarchical rules from using the method of [10]. We trained 4-gram Kneser-Ney language models on the respective target side of the corpus for each system. The NNJM is also limited to a 3-word target history, and both neural network models have an 11-word source context window. The log-linear combination of all components (including neural network models) is tuned using k-best optimization with an expected BLEU objective function [11] on held-out development data.

Table 1 shows model performance using the BLEU [12] and translation edit rate (TER) [13] metrics on a held-out blind set of the TransTac corpus. The addition of the neural network models has improved SMT performance significantly in both directions (+3 BLEU for English-to-Iraqi +2 BLEU for Iraqi-to-English), over our best-performing prior system [17].

<table>
<thead>
<tr>
<th></th>
<th>BLEU↑</th>
<th>TER↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2I</td>
<td>19.1</td>
<td>60.1</td>
</tr>
<tr>
<td>I2E</td>
<td>33.6</td>
<td>50.0</td>
</tr>
</tbody>
</table>

Table 1. BLEU and TER scores on held-out TransTac data

**ASR Error Detection:** The ASR error detector (AED) component performs a word level analysis of the 1-best ASR hypothesis. We trained a conditional random field (CRF) model with the same data used to train our ASR and using automatically generated error annotation by comparing ASR outputs to reference transcriptions. Words in the ASR hypothesis are represented using the following features: (1) ASR confidence; (2) language model perplexity; (3) whether a 1-best word is present in the (pruned) confusion network; (4) density of the corresponding slot in the confusion network; and (5) an indication if the word is commonly misrecognized. The performance of the AED component was evaluated on a collection of utterances designed to be representative of errors encountered by S2S systems [15]. Based on the ASR WERs reported earlier, we choose operating points corresponding to 1% false alarm at word level for English and 2% false alarm for Iraqi Arabic. The corresponding detection rates for AED were 36.4% and 19.8% respectively. These significant improvements over our previous operationalized AED implementation [21] are attributable to both improvements in ASR but also use of new features and classifier formulation for AED [14].

Word-level AED analysis is aggregated to identify error spans (contiguous spans of detected word-level errors). To prevent very large spans from being identified, we apply a heuristic that temporarily increases the operating threshold of the AED to localize only the highest confidence sub-span within large error spans. The results of ASR error detection are used to select an appropriate system action.
Note that the AED approach employed in this work only uses resources available for training and evaluation of the ASR which makes this approach extensible to new languages. Furthermore, features used by AED are based on rich information produced by the ASR. The need for tighter integration of the ASR error detection capability with the ASR makes a case for implementation of AED as a commonly available module within modern ASRs.

**Interactive Error Recovery:** Error recovery is presented as an optional module which users can choose to enable. When error recovery is enabled, action selection filters out very small error spans (<0.25 seconds long) identified by the AED analysis and ranks the spans by the maximum error confidence of their constituent tokens. Furthermore, action selection is influenced by the discourse state. For example, if three consecutive attempts fail to resolve an error, action selection by-passes error recovery. Table 2 shows three excerpts of an English speaker using the S2S system to communicate with an Iraqi Arabic speaker. Figure 3 summarizes the dialog model of error recovery.

<table>
<thead>
<tr>
<th>Excerpt I: No error is detected</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speaker</strong></td>
</tr>
<tr>
<td><strong>ASR</strong></td>
</tr>
<tr>
<td><strong>Translation</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Excerpt II: Error is detected (True Detection)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speaker</strong></td>
</tr>
<tr>
<td><strong>ASR</strong></td>
</tr>
<tr>
<td><strong>Clarification</strong></td>
</tr>
<tr>
<td><strong>Speaker</strong></td>
</tr>
<tr>
<td><strong>ASR</strong></td>
</tr>
<tr>
<td><strong>Translation</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Excerpt II: Error is detected (False Alarm)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Speaker</strong></td>
</tr>
<tr>
<td><strong>ASR</strong></td>
</tr>
<tr>
<td><strong>Clarification</strong></td>
</tr>
<tr>
<td><strong>Speaker</strong></td>
</tr>
<tr>
<td><strong>ASR</strong></td>
</tr>
<tr>
<td><strong>Translation</strong></td>
</tr>
</tbody>
</table>

Table 2. Excerpts of interaction with the English-Iraqi Arabic S2S System

Excerpt I in Table 2 shows the S2S system behavior when no error is detected in the user input (HT). In this case, a translation is presented to the listener (ST). Excerpts II and III shows the case where the AED finds an error (a true detection in II and a false alarm in III). In each case, the speaker is asked to confirm (SC) if the audio corresponding to the top ranked error span corresponds to a name (or a concept that could be transferred as-is to the other lan-
guage). The user can respond (HR) by saying “yes” to confirm the accurate error detection and localization as shown in excerpt II, or skip the system’s attempt to recover from a potential error by issuing the “go ahead” command. If error detection is confirmed, the audio segment corresponding to the error span is spliced into the translation through the source-target alignment produced by SMT. In addition to commands shown in excerpt II and III, the user can also choose to rephrase the input which is analyzed as a new input.

![Dialog model of error recovery in S2S system](image)

Figure 3. Dialog model of error recovery in S2S system

System behaviors in terms of action selection and prompt generation are symmetric in both directions (i.e. E2I and I2E). We note that, given the user-mediated nature of this error recovery strategy, successful cross-lingual concept transfer depends not only on component performance, but also on the appropriateness of users’ responses to clarifications.

3. Evaluation

3.1. Experiment Design & Data Collection

DARPA Broad Operational Language Technologies (BOLT) is a three-phase research focused on advancing the state of the art in translation technologies. Under the third phase of this program, an evaluation of error tolerant S2S systems was conducted by NIST over 5 days in January 2015.

Nine speaker pairs, each comprised of one native English speaker and one native Iraqi Arabic speaker, were tasked with communicating with one another using only a S2S system. The speakers were placed in two rooms separated by a see-through glass wall that served as a sound barrier. Each speaker pair interacted with each system over two 120-minute-long sessions. The S2S system was configured to perform ASR error detection in only one of the two sessions. The ordering of the two sessions (with and without error detection) was balanced across speaker pairs.

Each speaker pair was provided 16 conversational scenarios to be accomplished in each session. The scenarios assigned one of the two speakers a conversational driver role and the other a respondent role. The driver of the scenario was provided a conversational objective, such as ascertaining the extent of damage due to a hypothetical natural disaster. Speakers
were asked to achieve the conversational objective for each scenario in less than 8 minutes, and then to move on to the next scenario until either all scenarios were completed or the 120 minutes had expired (whichever came first). The scenarios cover a range of conversational domains relevant to military and humanitarian operations as well as everyday conversational topics like sports, family and pets.

Before using the S2S system, speakers received 30 minutes of training which included a 4-minute-long video demonstrating SuperMic features and the error recovery functionality, as well as several minutes of free-form interaction for additional practice.

Each conversational trial collected was scored by 6 independent judges along three metrics. These metrics, listed below, used a 7 point scale ranging from -3 (strongly disagree) to +3 (strongly agree).

- **Goal**: The initiator of the dialog achieved his/her goal.
- **Quality**: The overall quality of the translations was adequate.
- **Clarification**: The clarification(s) helped during the dialog.

Note that the clarification scale is applicable only when the S2S system was configured to perform error recovery. In this paper, we will report average computed over the multiple judgements.

### 3.2. Results

<table>
<thead>
<tr>
<th>Clarification</th>
<th>ON</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Trials ↑</td>
<td>113</td>
<td>125</td>
</tr>
<tr>
<td>Avg. Trial Duration ↓</td>
<td>396.2</td>
<td>378.2</td>
</tr>
<tr>
<td>Turns per Trial (En) ↓</td>
<td>9.2</td>
<td>8.0</td>
</tr>
<tr>
<td>Turns per Trial (IA) ↓</td>
<td>9.7</td>
<td>8.5</td>
</tr>
<tr>
<td>#Clarifications (En)</td>
<td>191</td>
<td>-</td>
</tr>
<tr>
<td>#Clarifications (IA)</td>
<td>229</td>
<td>-</td>
</tr>
</tbody>
</table>

### ASR Performance

<table>
<thead>
<tr>
<th></th>
<th>(En)</th>
<th>(IA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WER ↓</td>
<td>5.7</td>
<td>6.8</td>
</tr>
<tr>
<td>OOV Rate (En) ↓</td>
<td>0.9</td>
<td>1.8</td>
</tr>
</tbody>
</table>

### ASR Error Detector Performance

<table>
<thead>
<tr>
<th></th>
<th>(En)</th>
<th>(IA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall ↑</td>
<td>30.4</td>
<td>-</td>
</tr>
<tr>
<td>Recall (IA) ↑</td>
<td>24.0</td>
<td>-</td>
</tr>
</tbody>
</table>

### SMT Performance

<table>
<thead>
<tr>
<th></th>
<th>(E2I)</th>
<th>(I2E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TER ↓</td>
<td>58.9</td>
<td>58.5</td>
</tr>
<tr>
<td>UNK rate (En) ↓</td>
<td>0.57</td>
<td>0.44</td>
</tr>
<tr>
<td>UNK rate (IA) ↓</td>
<td>0.42</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Table 3. Statistics of trials collected at BOLT evaluation of BBN’s S2S system & Component performance metrics

We logged 238 conversational trials over 18 sessions of use of our S2S system. S2S system configured for ASR error detection and recovery was used in 9 of these sessions during which
113 trials were logged. Table 3 provides summary statistics about these trials and reports the
performance of key system components (ASR, AED, SMT) in these trials both with and without error recovery (Clarification ON or OFF).

ASR performance, reported as WER and OOV rate, is consistent across the two types of sessions. The WER is significantly lower than on the development sets, especially for Iraqi Arabic (IA). This is likely because the development set was collected in a human-mediated cross-lingual communication setting unlike the computer-mediated setting in which the S2S system is used. AED performance is reported as recall at the operating points mentioned in section 2.2.3. Despite a more conservative choice of operating point, AED is able to accurately detect a larger fraction of erroneous words for English. SMT performance is measured both as TER on 1-best ASR hypothesis and as the rate of untranslatable input words (UNKs). TER is comparable to development sets. UNK rate for English is lower than Iraqi Arabic.

While error recovery appears to not harm ASR and MT performance, we note that it achieves lower conversational throughput. Speakers have 10% fewer conversations when error recovery is enabled, and the average duration of a conversational trial increased by 5%. User effort, quantified as the number of turns spoken per trial, increases by 15% with error recovery dialogs. Also, we found that 9% of the English error clarification sub-dialogs required more than one turn (7% for Iraqi Arabic). This is an improvement over previously reported measures of error recovery cost [16], where it was found that erroneous inputs consumed 1.4 clarification turns.

We found significant variation in all of the metrics reported in Table 3 across different speaker pairs and across different domains covered in the scenarios. The standard deviation for user effort across different speaker pairs (s.d.=2.89) was higher than across different domains (s.d.=1.26). Table 4 characterizes the effect of different system components on user effort by reporting Pearson correlations of various metrics with user effort. The correlations are computed over the variations among the speakers.

Without error recovery, higher WER corresponds to fewer turns taken by the English speaker. This is likely to be indicative of fewer concepts discussed by the speaker. More English OOVs lead to increased effort when error recovery is enabled. However, use of OOVs by the Iraqi speaker does not increase the number of turns. We found that the correlation between OOV rate and AED Recall for English is positive (r=0.18), but its negative for Iraqi Arabic (r=-0.23). This is due to the difference in AED performance between the two languages. For both the speakers, more untranslatable tokens lead to an increase in effort when error recovery is enabled.

Finally, we examine the human judgements of the conversational trails. Averages for the three scales are shown in Table 5. The use of error recovery does not improve goal achievement or quality of translation. However, a closer examination of the scenarios that
were trialed with both error recovery enabled and disabled (N=112) shows that for difficult scenarios, error recovery in fact improves conversational goal achievement. Difficult scenarios are defined as subset of scenarios that had below average goal achievement score in the baseline system (i.e. without error recovery).

<table>
<thead>
<tr>
<th>Clarification</th>
<th>ON</th>
<th>OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal ↑</td>
<td>2.27</td>
<td>2.40</td>
</tr>
<tr>
<td>Quality ↑</td>
<td>1.79</td>
<td>1.89</td>
</tr>
<tr>
<td>Clarification ↑</td>
<td>1.70</td>
<td>-</td>
</tr>
<tr>
<td>#Difficult Trials ↑</td>
<td>38</td>
<td>74</td>
</tr>
<tr>
<td>Goal (Difficult Trials) ↑</td>
<td>2.10</td>
<td>2.00</td>
</tr>
<tr>
<td>Goal (Easy Trials) ↑</td>
<td>2.47</td>
<td>2.73</td>
</tr>
</tbody>
</table>

Table 5. Judgement scores for BBN’s S2S systems

Intuitively, this can be interpreted as indication that error recovery is only helpful when the users are having a conversation that the S2S system has difficulty with. For easy scenarios, error recovery does not help. One of the immediate design implications of this for S2S systems is to empower the users with the ability to turn error recovery on only when needed. However, user enabled error recovery assumes that the users are able to judge when the S2S system is having difficulty with a conversation.

4. Conclusion

Application of interactive error recovery have been investigated for multiple spoken language technologies including spoken dialog systems [18][19] and S2S systems [16][20]. Our prior work attempted to address various types of errors encountered due to imperfections of S2S system components. In contrast to that, the work presented in this paper builds on recent improvements in component performance and focuses only on errors introduced by misrecognition of input speech. This leads to simplification of interaction design as well as reduction in the cost of error recovery, quantified here in terms of user effort and time. Extending the need to minimize the cost of error recovery, we have principally chosen conservative false-alarm rate for AED based on language specific WERs. While in the work presented here we focus on an eyes-free use case, another configuration of BBN’s S2S system employs a touch screen interface along with visual cues to resolve errors without requiring spoken commands.

References


[19] D. Bohus, and A. I. Rudnicky, "Sorry, i didn’t catch that! - an investigation of non-understanding errors and recovery strategies", Proc. of SIGDIAL, pp. 128-143, 2005


Mixed-Domain vs. Multi-Domain
Statistical Machine Translation

Matthias Huck
mhuck@inf.ed.ac.uk
Alexandra Birch
a.birch@ed.ac.uk
Barry Haddow
bhaddow@inf.ed.ac.uk
School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh EH8 9AB, UK

Abstract
Domain adaptation boosts translation quality on in-domain data, but translation quality for
domain adapted systems on out-of-domain data tends to suffer. Users of web-based translation
services expect high quality translation across a wide range of diverse domains, and what makes
the task even more difficult is that no domain label is provided with the translation request.
In this paper we present an approach to domain adaptation which results in large-scale, general
purpose machine translation systems. First, we tune our translation models to multiple individ-
ual domains. Then, by means of source-side domain classification, we are able to predict the
domain of individual input sentences and thereby select the appropriate domain-specific model
parameters. We call this approach multi-domain translation.
We develop state-of-the-art, domain-adapted translation engines for three broadly-defined do-
mains: TED talks, Europarl, and News. Our results suggest that multi-domain translation
performs better than a mixed-domain approach, which deploys a system that has been tuned on
a development set composed of samples from many domains.

1 Introduction
Domain adaptation is a common approach to significantly improve machine translation quality
on input documents from a given domain. Domain adaptation techniques for statistical ma-
chine translation (SMT) have been extensively studied and are well established (Federico and
Bertoldi, 2012). In practice, machine translation systems are often engineered to perform well
on the domain of one specific application. Most research on domain adaptation assumes that
any prospective input data originates from a single domain, and the characteristics of this do-
main are known beforehand, e.g. by means of existing samples from the same domain which
can be employed for training and tuning. The adaptation task is then defined as utilizing a small
amount of in-domain training resources effectively in order to learn system parameters that are
more appropriate for translating in-domain input. The in-domain training resources constitute
a minor fraction of the overall training data only, the majority of which has a domain mismatch
with the designated application.

The downside of systems that have been highly tweaked towards the characteristics of a
single domain is a diminished translation quality on out-of-domain data (Haddow and Koehn,
2012). Online translation systems, on the other hand, are usually designed for open-domain
scenarios where the domain of the input text is not predefined. Being able to take advantage
of the benefits of domain adaptation while not having to compromise quality on out-of-domain
data would be desirable for online systems.
A viable utilization of domain adaptation approaches in open-domain online translation systems comes in two components:

- A number of different parameter sets, each tuned to optimize translation quality on texts from a specific domain.
- A text classifier that predicts the domain of foreign-language input data prior to decoding.

As the input data is not labeled with a domain (and, in an open-domain setting, may even originate from a new domain), the text classifier first has to assign the most likely class from a set of multiple known domains. The decoder is then reconfigured with a domain-specific parameter set (e.g., a weight vector), which should be the most appropriate one for achieving high translation quality on the current input. We refer to this approach as **multi-domain SMT**.

In this work we investigate different methods for domain classification:

- Classification based on the scores of language models (LMs) which have been interpolated with interpolation weights tuned on in-domain development sets.
- Maximum entropy text classifiers trained on medium-sized training corpora.
- Maximum entropy text classifiers trained on the same smaller domain-specific development sets which are employed for tuning the machine translation systems.

An obvious alternative method for building an open-domain online translation system is tuning on a corpus containing samples of texts from all known domains, which collectively are considered to be representative for the application. We refer to this approach as **mixed-domain SMT**. A difficulty here is the choice of the corpus samples in a way that brings about good performance across all domains. However, a high-quality generic system with a single parameter set that does not depend on a domain label is appealing.

In the empirical part of this paper, we compare multi-domain and mixed-domain SMT on the English→German, English→Italian, English→Portuguese, and English→Greek language pairs using training corpora of diverse origin, totalling tens of millions of parallel sentences.

## 2 Related Work

A significant amount of research on domain adaptation for SMT has been conducted in recent years. Some methods which are commonly used are:

- Tuning of the decoder model weights (Och and Ney, 2002) on an in-domain development set (Pecina et al., 2012).
- Model combination (of language models, translation models, or reordering models) via interpolation or other schemes, e.g. phrase table fill-up (Foster and Kuhn, 2007; Koehn and Schroeder, 2007; Nakov, 2008; Bisazza et al., 2011; Niehues and Waibel, 2012; Chen et al., 2013).
- Data selection (Moore and Lewis, 2010; Axelrod et al., 2011).
- Instance weighting (Matsoukas et al., 2009; Foster et al., 2010; Shah et al., 2012; Mansour and Ney, 2012).
- Further exploitation of in-domain monolingual data (Ueffing et al., 2007; Bertoldi and Federico, 2009; Schwenk and Senellart, 2009; Lambert et al., 2011).
- Domain-specific features, e.g. binary features indicating the provenance of phrase pairs as implemented in the open-source Moses toolkit (Durrani et al., 2013b) or “domain augmentation” (Clark et al., 2012).

However, few authors have tackled the question of how to benefit from domain adaptation in scenarios where a domain label of the input is not present. An important aspect of our approach to multi-domain MT is the need for domain classification.
Xu et al. (2007) perform domain classification for a Chinese→English task. The domains are newswire and newsgroup. The classifiers operate on whole documents rather than on individual sentences. The authors propose two techniques for domain classification. Their first technique is based on interpolated LMs: a general-domain LM is interpolated with LMs which were trained on in-domain development sets, resulting in a number of domain-specific interpolated LMs. The interpolation weight is heuristically chosen. The classifier computes LM perplexities over input documents and assigns the domain with the lowest perplexity. Their second technique is based on a metric which measures similarity wrt. vocabulary.

Banerjee et al. (2010) conduct machine translation experiments with classification of two technical documentation domains, availability and security in the area of computing. Empirical results on Chinese→English and English→Chinese tasks are presented. The authors build a Support Vector Machine (SVM) classifier using Term Frequency Inverse Sentence Frequency features over bigrams of stemmed content words. Classification is carried out on the level of individual sentences. The SVM is trained on the SMT training corpora (~226k sentences in total). Several setups with different domain-adapted and domain-agnostic systems are evaluated. The authors show that a pipeline with the SVM classifier is effective in multi-domain translation.

Wang et al. (2012) distinguish generic and patent domain data in experiments on 20 language pairs. For domain classification, the authors rely on averaged perceptron classifiers with various phrase-based features. The machine translation development sets serve as training data for the classifiers. An interesting aspect of their translation experiments is that they utilize a multi-domain optimization in order to jointly tune weights for all domains in a single run of lattice MERT (Macherey et al., 2008).

In a related strand of research, source-side text classifiers have recently been employed in order to detect Arabic dialects and select SMT systems accordingly (Salloum et al., 2014; Mansour et al., 2014).

3 Text Domains

Our application scenario is an online translation service with the requirement to provide high-quality translation not only of texts from a single domain, but of a wider range of text types. We therefore study a use case where the translation system is supposed to perform well on the following domains: TED talks, Europarl, and News. These three domains are fairly coarse-grained. Different documents from one of the domains are mostly not consistent regarding the covered topics. While all three domains comprise heterogeneous topics, the domains are set apart from each other by means of text style.

TED talks are transcripts of spoken language from short public presentations. The presentations often cover scientific subjects which are expressed in layman’s terms and in an informal manner. TED talks are not spontaneous speech. They are however designed to be entertaining. Europarl texts are transcripts of speeches on political matters from parliamentary proceedings. News texts are written news articles.

TED talks, Europarl, and News could be described as “genres”. We denote them as domains throughout this paper because the term “domain” is well established in related machine translation research literature and often used in a broad sense.

TED talks, Europarl, and News have been highly relevant domains in recent machine translation research. The International Workshop on Spoken Language Translation (IWSLT) hosts a yearly open evaluation campaign which focuses on the translation of TED talks since 2011 (Federico et al., 2011). The European Parliament Proceedings Parallel Corpus (Koehn, 2005) has been an influential resource for machine translation research ever since its first release over

1http://www.iwslt.org
a decade ago. It is freely available and includes parallel text for 21 European languages. Test sets and training data that enables research on machine translation of texts from the News domain have regularly been released for the shared translation task of the Workshop on Statistical Machine Translation (WMT). The WMT “newstest” corpora have become important test sets to measure progress in machine translation between English and several European languages.

Previous research has indicated that divergences of domains such as TED talks, Europarl, and News empirically matter for machine translation. For instance, Ruiz and Federico (2014) systematically analyze characteristics of TED talk transcripts and News Commentary texts and point out the differences in detail for English-German.

4 Adaptation Techniques

In order to adapt systems to a domain, we leverage previously proposed techniques. First, we tune the model weights on in-domain development sets. Secondly, we linearly interpolate language models: rather than training a single large LM on all the target-side data, we train separate models on each corpus and interpolate them based on weights that minimize perplexity over the development set, resulting in a new, domain-adapted large LM that can be used by the decoder. Finally, we add binary features indicating the provenance of phrase pairs: if a phrase pair has been seen in a particular training corpus, a binary indicator associated with the respective training corpus fires on application of that phrase pair during decoding. This increases the amount of features by a number equal to the number of parallel training corpora.

5 Domain Classification

A domain classifier is required for multi-domain SMT on unlabeled input data. The domain of the source-side text we receive for translation is unknown and we need to predict it in order to select appropriate decoder parameters.

We investigate classification based on source-side LMs as well as different variations of a maximum entropy classifier.

Unlike Xu et al. (2007), who classify documents, we predict the domain label on the level of single sentences. Sentence-level classification has the advantage that document boundaries do not need to be present, and we are able to decode an unstructured incoming stream of sentences.

5.1 Source LM Classifier

Classification based on source-side LMs predicts the domain label from LM scores. We train separate LMs on the source side of each parallel training corpus. Then we create adapted LMs for each domain by linearly interpolating those source language LMs, where the interpolation weights are tuned to minimize perplexity on the source side of the respective in-domain development set. The classifier computes LM scores with each of the domain-adapted source LMs and selects the domain label according to maximum score. Note that for the scores to be on a comparable level, all domain-adapted source LMs should be interpolations of the same set of individual LMs.

Besides classifying sentences rather than documents, our method differs from the one proposed in (Xu et al., 2007) with respect to another aspect: Xu et al. (2007) interpolate LMs trained on the respective in-domain development sets with a single huge generic LM. Disadvantages of their method are (1.) the tiny size of the development corpora in terms of LM training, and (2.) the necessity of setting the interpolation weights heuristically. We overcome these drawbacks by resorting to a more straightforward framework of reserving the in-domain develop-
development sets for tuning source LM interpolation weights. Furthermore, we argue that source-side scores of the interpolated LMs employed in our classifier may to some extent resemble target-side scores of the interpolated LMs which are applied in the respective domain-adapted SMT systems.

5.2 Maximum Entropy Classifiers

Maximum entropy text classifiers can utilize a larger number of features in order to predict the label (Berger et al., 1996). We incorporate features from single words, pairs of adjacent words, the first word of the sentence, and the last word of the sentence. The model is trained with L-BFGS (Byrd et al., 1995) and regularized using a Gaussian prior.

We build maximum entropy (ME) classifiers under two different training conditions: using the MT development sets (which are rather small) as training data, and using selected other corpora as training data (which might not always exactly match what is defined as in-domain to the MT systems, as the development sets essentially constitute the domains).

In a further flavor of our ME classifiers, in addition to the previously described features, we include source LM indicator features in the ME model. To create these features, we score the sentence with the same domain-adapted source LMs as employed by the source LM classifier. For each of the domain-adapted LMs, an associated source LM indicator feature fires if the respective LM yields the maximum LM score.

Overall, we end up with four variations:

- ME\textsubscript{train} Classifier trained on medium-sized training corpora with the basic set of features.
- ME\textsubscript{train+lm} Classifier trained on medium-sized training corpora with the basic set of features plus source LM indicator features.
- ME\textsubscript{dev} Classifier trained on the MT development sets with the basic set of features.
- ME\textsubscript{dev+lm} Classifier trained on the MT development sets with the basic set of features plus source LM indicator features.

6 Experimental Setup

We use Moses (Koehn et al., 2007) for machine translation, MGIZA++ (Gao and Vogel, 2008) to train word alignments, KenLM (Heafield, 2011) for LM training and scoring, SRILM (Stolcke, 2002) for LM interpolation, and the Stanford Classifier\textsuperscript{3} for maximum entropy text classification. We present experimental results on English\textarrow{German, English\textarrow{Italian, English\textarrow{Portuguese, and English\textarrow{Greek translation tasks.

6.1 Training Data

Our SMT systems are trained with the following bilingual corpora:

- TED from WIT3 (Cettolo et al., 2012)
- Europarl (Koehn, 2005)
- JRC-Acquis 3.0 (Steinberger et al., 2006)
- DGT’s Translation Memory (Steinberger et al., 2012) as distributed in OPUS (Tiedemann, 2012)
- OPUS European Central Bank (ECB)
- OPUS European Medicines Agency (EMEA)
- OPUS EU Bookshop
- OPUS OpenSubtitles\textsuperscript{4}

\textsuperscript{3}http://nlp.stanford.edu/software/classifier.shtml
\textsuperscript{4}http://www.opensubtitles.org
Statistics of a concatenation of all bilingual training corpora are presented in Table 1.

For language modeling on the target side, we furthermore add monolingual corpora from recent (April 2015) Wikipedia database dumps\(^5\) and—for German—the News Crawl corpora provided for the WMT 2015 shared translation task. Plain text was obtained from the Wikipedia XML dumps with the Wikipedia Extractor\(^6\) tool. Statistics of the additional monolingual training corpora are presented in Table 2.

### 6.2 Machine Translation Systems

Word alignments are created by aligning the data in both directions and symmetrizing the two trained alignments (Och and Ney, 2003; Koehn et al., 2003). We extract phrases up to a maximum length of five. The MT systems comprise these features:

- Phrase translation log-probabilities, smoothed with Good-Turing smoothing (Foster et al., 2006), and lexical translation log-probabilities in both directions.
- Phrase penalty and word penalty.
- Distance-based distortion cost.
- A hierarchical lexicalized reordering model (Galley and Manning, 2008).
- A 5-gram operation sequence model (Durrani et al., 2013a).
- Seven binary features indicating absolute occurrence count classes of phrase pairs.
- Sparse phrase length features.
- Sparse lexical features for the top 200 words.
- A 5-gram LM with modified Kneser-Ney smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998). We discard singleton \(n\)-grams of order three and higher.

Feature weights are optimized to maximize BLEU (Papineni et al., 2002) with batch MIRA (Cherry and Foster, 2012) on 1000-best lists. We prune the phrase table to a maximum of 100

---

\(^5\) [http://dumps.wikimedia.org](http://dumps.wikimedia.org)

\(^6\) [https://github.com/bwbaugh/wikipedia-extractor](https://github.com/bwbaugh/wikipedia-extractor)
best translation options per distinct source side and apply a minimum score threshold of 0.0001 on the source-to-target phrase translation probability. We use cube pruning in decoding. Pop limit and stack limit are set to 1000 for tuning and to 5000 for testing. We disallow reordering over punctuation. Furthermore, Minimum Bayes Risk decoding is employed for testing. Translation quality is measured in truecase with BLEU.

6.2.1 Development and Test Sets
Wherever possible, we tune and test on common sets as distributed on http://matrix.statmt.org/test_sets/list/ and https://wit3.fbk.eu/. The domain-adapted Europarl systems are tuned on the test2006 set. The domain-adapted TED systems are tuned on a concatenation of TED-dev2010 and TEDX-dev2012 (En→De), on a concatenation of TED-dev2010 and TEDX-dev2014 (En→It), and on TED-dev2010 (En→Pt). An English→Greek translation task was so far never organized as part of any of the IWSLT evaluation campaigns and for that reason no common TED sets exist for that language pair. However, the 2012-02 release of the WIT3 corpus contains a parallel corpus of 84,831 English-Greek sentence pairs. We reserved every fifth sentence of this data for tuning and testing. Of the tuning and testing part of the corpus, we assign every fourth sentence to the test set and the rest to the tuning set.7 The domain-adapted News systems are tuned on a concatenation of the newstest2008-2012 sets (En→De) and on newstest2009 (En→It). We use newsycomb2009 as an English→Italian News domain test set for lack of other English-Italian News test sets. Note that newsycomb2009 is a small set of only 502 sentences. No News test data was available to us for the English→Portuguese and English→Greek language pairs, so we experiment with only two domains (TED and Europarl) on these tasks.

The Portuguese TED development and test sets are Brazilian Portuguese whereas the Europarl sets are European Portuguese. The two Portuguese dialects have a number of differences in written language. Marujo et al. (2011) give a brief overview.

6.2.2 Domain-Adapted SMT
For our domain adaptation experiments, we first tune the systems with the features described above on the respective in-domain development set (TED-tuned, Europarl-tuned, News-tuned). We next replace the large baseline LM with a domain-specific interpolated LM (+LM interp.). We then add binary features indicating the provenance of phrase pairs (+LM interp. + indicator feat.).

6.2.3 Mixed-Domain SMT
We build mixed-domain SMT systems by tuning on a development corpus containing samples of texts from all domains. We include a balanced amount of development data from the different domains in the mixed-domain development set in order to avoid a bias towards any specific domain.

The mixed-domain systems (Mixed-domain-tuned) are tuned on a concatenation of TED-dev2010 and TEDX-dev2012 and Europarl test2006 and newstest2009 (En→De), on a concatenation of TED-dev2010 and TEDX-dev2014 and Europarl test2006 and newstest2009 (En→It), on a concatenation of TED-dev2010 and every second sentence from Europarl test2006 (En→Pt), and on a concatenation of every sixth sentence from our Greek TED development set and the full Europarl test2006 (En→El).

The LMs for the mixed-domain systems are trained on the full target language monolingual training data, not interpolated from individual LMs. Binary features indicating the provenance of phrase pairs are not used.

7We end up with English-Greek TED corpus sizes of 67,865 sentences for training, 12,725 sentences for tuning, and 4,241 for testing.
### 6.2.4 Multi-Domain SMT

Multi-domain systems classify the input sentence with a domain classifier. They parameterize the decoder according to the predicted domain label. We use the parameters of the Domain-tuned + LM interp. + indicator feat. MT setups. We evaluate five multi-domain system per language pair, one for each of our domain classifiers.

### 6.2.5 Oracle-Domain SMT

In an oracle domain setup, we assume that the correct domain label of each input sentence is given. We can parameterize the decoder according to the gold-standard domain label.

### 6.3 Domain Classifiers

The ME_{train} classifiers are trained on the source language side of the TED portion of the training data and fractions of both the Europarl portion of the training data and the English News Crawl 2014 corpus as provided for the WMT 2015 shared translation task. Again, we include a balanced amount of data from the different domains (e.g. 10% of the Europarl data and 1% of the News Crawl 2014 data for English→German) in order to not give preference to any of the domains. The ME_{dev} classifiers are trained on the MT development sets as used for mixed-domain MT tuning.

While building common English domain classifiers would be possible, we decided to train separate ones for each task and utilize the data resources from the respective language pair.

### 7 Experimental Results

#### Domain classifiers

The accuracies of the domain classifiers are presented in Table 3. We report accuracies (micro-averaged F1) on a concatenation of all test sets for each of the language pairs with the source LM classifier and four variations of the ME classifier (cf. Section 5). Naturally, accuracies are higher for the tasks where only two domains have to be distinguished (En→Pt, En→El) than on the tasks with three domain classes (En→De, En→It). Accuracies are generally of a high level, even for the simple source LM classifier. We are going to evaluate in MT experiments whether any of the differences in classification accuracy carry over to translation quality of multi-domain systems.

#### Translation quality

Tables 4-7 contain BLEU scores obtained with all MT systems on the test sets from the various domains for the four language pairs. The TED test sets are the common IWSLT tests sets, the Europarl test sets have been downloaded from matrix.statmt.org, and the News test sets are the standard ones from the WMT shared tasks. We test all systems on the sets from all domains, in particular, domain-adapted systems are tested on out-of-domain sets as well. We also report BLEU scores on concatenations of all test sets (all) in order to measure overall performance in open-domain scenarios.
Table 4: English→German experimental results (truecase BLEU scores).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TED-tuned</td>
<td>24.3</td>
<td>26.7</td>
<td>22.9</td>
<td>25.3</td>
<td>21.9</td>
<td>21.7</td>
<td>20.2</td>
<td>20.8</td>
<td>22.9</td>
<td>22.4</td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>24.2</td>
<td>26.8</td>
<td>22.7</td>
<td>24.6</td>
<td>21.2</td>
<td>20.9</td>
<td>19.5</td>
<td>20.1</td>
<td>22.2</td>
<td>21.9</td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>24.4</td>
<td>26.7</td>
<td>23.1</td>
<td>25.0</td>
<td>20.9</td>
<td>20.7</td>
<td>19.5</td>
<td>20.2</td>
<td>22.3</td>
<td>21.9</td>
</tr>
<tr>
<td>Europarl-tuned</td>
<td>24.4</td>
<td>26.3</td>
<td>22.8</td>
<td>25.2</td>
<td>22.4</td>
<td>22.1</td>
<td>20.4</td>
<td>21.2</td>
<td>23.0</td>
<td>22.6</td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>23.1</td>
<td>24.8</td>
<td>22.0</td>
<td>24.0</td>
<td>22.5</td>
<td>22.2</td>
<td>19.4</td>
<td>19.7</td>
<td>21.8</td>
<td>21.8</td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>22.8</td>
<td>24.7</td>
<td>21.7</td>
<td>24.0</td>
<td>22.6</td>
<td>22.1</td>
<td>19.3</td>
<td>19.5</td>
<td>21.6</td>
<td>21.7</td>
</tr>
<tr>
<td>News-tuned</td>
<td>23.7</td>
<td>26.2</td>
<td>22.2</td>
<td>24.5</td>
<td>21.5</td>
<td>21.4</td>
<td>20.6</td>
<td>21.1</td>
<td>22.9</td>
<td>22.2</td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>23.3</td>
<td>25.8</td>
<td>22.1</td>
<td>24.2</td>
<td>21.3</td>
<td>21.2</td>
<td>20.4</td>
<td>20.9</td>
<td>22.6</td>
<td>21.9</td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>23.4</td>
<td>25.7</td>
<td>22.2</td>
<td>24.4</td>
<td>21.4</td>
<td>21.0</td>
<td>20.5</td>
<td>20.9</td>
<td>22.6</td>
<td>21.9</td>
</tr>
<tr>
<td>Mixed-domain-tuned</td>
<td>24.6</td>
<td>26.9</td>
<td>23.1</td>
<td>25.3</td>
<td>22.2</td>
<td>22.0</td>
<td>20.6</td>
<td>21.2</td>
<td>23.2</td>
<td>22.7</td>
</tr>
<tr>
<td>Multi-domain, LM classifier</td>
<td>24.4</td>
<td>26.8</td>
<td>23.1</td>
<td>24.9</td>
<td>22.5</td>
<td>22.1</td>
<td>20.1</td>
<td>20.6</td>
<td>22.5</td>
<td>22.4</td>
</tr>
<tr>
<td>Multi-domain, ME En classifier</td>
<td>24.3</td>
<td>26.7</td>
<td>22.8</td>
<td>24.7</td>
<td>22.5</td>
<td>22.1</td>
<td>20.3</td>
<td>20.7</td>
<td>22.5</td>
<td>22.4</td>
</tr>
<tr>
<td>Multi-domain, ME En+lm classifier</td>
<td>24.3</td>
<td>26.7</td>
<td>23.0</td>
<td>24.9</td>
<td>22.5</td>
<td>22.2</td>
<td>20.4</td>
<td>20.8</td>
<td>22.5</td>
<td>22.5</td>
</tr>
<tr>
<td>Multi-domain, ME Ev classifier</td>
<td>24.2</td>
<td>26.8</td>
<td>22.8</td>
<td>24.7</td>
<td>22.5</td>
<td>22.1</td>
<td>20.3</td>
<td>20.8</td>
<td>22.5</td>
<td>22.5</td>
</tr>
<tr>
<td>Multi-domain, ME Ev+lm classifier</td>
<td>24.3</td>
<td>26.9</td>
<td>22.9</td>
<td>24.9</td>
<td>22.5</td>
<td>22.1</td>
<td>20.3</td>
<td>20.7</td>
<td>22.5</td>
<td>22.5</td>
</tr>
<tr>
<td>Oracle-domain</td>
<td>24.4</td>
<td>26.7</td>
<td>23.1</td>
<td>25.0</td>
<td>22.6</td>
<td>22.1</td>
<td>20.5</td>
<td>20.9</td>
<td>22.6</td>
<td>22.5</td>
</tr>
</tbody>
</table>

Table 5: English→Italian experimental results (truecase BLEU scores).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TED-tuned</td>
<td>26.8</td>
<td>26.6</td>
<td>27.3</td>
<td>32.9</td>
<td>24.6</td>
<td>25.3</td>
<td>27.5</td>
<td>26.9</td>
<td></td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>27.0</td>
<td>27.3</td>
<td>27.7</td>
<td>33.5</td>
<td>24.3</td>
<td>24.7</td>
<td>27.3</td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>27.3</td>
<td>27.7</td>
<td>28.0</td>
<td>33.4</td>
<td>24.1</td>
<td>24.6</td>
<td>27.5</td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td>Europarl-tuned</td>
<td>24.6</td>
<td>24.3</td>
<td>25.5</td>
<td>30.5</td>
<td>24.9</td>
<td>25.6</td>
<td>26.5</td>
<td>25.8</td>
<td></td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>23.9</td>
<td>24.4</td>
<td>25.6</td>
<td>30.0</td>
<td>24.9</td>
<td>25.4</td>
<td>25.2</td>
<td>25.6</td>
<td></td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>24.2</td>
<td>24.3</td>
<td>25.8</td>
<td>29.9</td>
<td>25.0</td>
<td>25.5</td>
<td>25.2</td>
<td>25.7</td>
<td></td>
</tr>
<tr>
<td>News-tuned</td>
<td>26.8</td>
<td>26.5</td>
<td>27.2</td>
<td>32.5</td>
<td>24.6</td>
<td>25.3</td>
<td>27.7</td>
<td>26.9</td>
<td></td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>26.8</td>
<td>27.4</td>
<td>28.0</td>
<td>33.3</td>
<td>24.7</td>
<td>25.1</td>
<td>27.8</td>
<td>27.2</td>
<td></td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>26.9</td>
<td>27.5</td>
<td>27.9</td>
<td>33.1</td>
<td>24.6</td>
<td>25.1</td>
<td>27.8</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>Mixed-domain-tuned</td>
<td>26.9</td>
<td>26.7</td>
<td>27.0</td>
<td>32.5</td>
<td>25.0</td>
<td>25.6</td>
<td>27.9</td>
<td>27.0</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, LM classifier</td>
<td>27.3</td>
<td>27.7</td>
<td>27.9</td>
<td>33.4</td>
<td>24.9</td>
<td>25.5</td>
<td>27.3</td>
<td>27.2</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME En classifier</td>
<td>27.3</td>
<td>27.6</td>
<td>27.8</td>
<td>33.3</td>
<td>25.0</td>
<td>25.5</td>
<td>26.1</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME En+lm classifier</td>
<td>27.3</td>
<td>27.6</td>
<td>27.9</td>
<td>33.3</td>
<td>25.0</td>
<td>25.5</td>
<td>25.8</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME Ev classifier</td>
<td>27.3</td>
<td>27.7</td>
<td>27.9</td>
<td>33.1</td>
<td>25.0</td>
<td>25.4</td>
<td>27.4</td>
<td>27.1</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME Ev+lm classifier</td>
<td>27.3</td>
<td>27.8</td>
<td>27.9</td>
<td>33.3</td>
<td>24.9</td>
<td>25.5</td>
<td>27.6</td>
<td>27.2</td>
<td></td>
</tr>
<tr>
<td>Oracle-domain</td>
<td>27.3</td>
<td>27.7</td>
<td>28.0</td>
<td>33.4</td>
<td>25.0</td>
<td>25.5</td>
<td>27.8</td>
<td>27.2</td>
<td></td>
</tr>
</tbody>
</table>

Domain adaptation. As expected, the domain-adapted systems perform better on in-domain data than when evaluated in a cross-domain experiment. The in-domain systems outperform systems tuned on out-of-domain development sets. More aggressive domain adaptation via LM interpolation and binary provenance indicator features gives mixed results. For English→German, we basically do not observe gains over in-domain tuning with any of the two adaptation methods. They just further reduce quality on out-of-domain data. For English→Italian, we observe gains with both methods on in-domain test sets on top of the TED-tuned system, but not on top of the Europarl-tuned and News-tuned systems. For English→Portuguese and English→Greek, we see larger gains mostly due to LM interpolation and on TED-domain adaptation.
Table 6: English→Portuguese experimental results (truecase BLEU scores).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TED-tuned</td>
<td>32.1</td>
<td>34.0</td>
<td>34.8</td>
<td>33.9</td>
<td>32.8</td>
<td>27.6</td>
<td>27.7</td>
<td>30.6</td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>33.7</td>
<td>35.2</td>
<td>36.1</td>
<td>34.9</td>
<td>34.6</td>
<td>25.5</td>
<td>25.2</td>
<td>30.2</td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>33.7</td>
<td>35.6</td>
<td>36.2</td>
<td>35.4</td>
<td>35.2</td>
<td>24.9</td>
<td>24.5</td>
<td>30.0</td>
</tr>
<tr>
<td>Europarl-tuned</td>
<td>29.9</td>
<td>32.0</td>
<td>31.0</td>
<td>31.6</td>
<td>31.2</td>
<td>30.1</td>
<td>30.1</td>
<td>30.7</td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>26.7</td>
<td>28.4</td>
<td>27.5</td>
<td>28.6</td>
<td>28.2</td>
<td>30.3</td>
<td>30.1</td>
<td>30.1</td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>26.5</td>
<td>27.7</td>
<td>27.1</td>
<td>28.0</td>
<td>27.6</td>
<td>30.4</td>
<td>30.1</td>
<td>28.8</td>
</tr>
<tr>
<td>Mixed-domain-tuned</td>
<td>32.3</td>
<td>34.1</td>
<td>33.4</td>
<td>33.2</td>
<td>33.5</td>
<td>29.4</td>
<td>29.5</td>
<td>31.6</td>
</tr>
<tr>
<td>Multi-domain, LM classifier</td>
<td>33.7</td>
<td>35.4</td>
<td>35.9</td>
<td>35.1</td>
<td>35.0</td>
<td>30.4</td>
<td>30.1</td>
<td>32.6</td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{train} classifier</td>
<td>33.6</td>
<td>35.4</td>
<td>35.9</td>
<td>35.0</td>
<td>34.8</td>
<td>30.4</td>
<td>30.1</td>
<td>32.6</td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{train+lm} classifier</td>
<td>33.7</td>
<td>35.4</td>
<td>35.9</td>
<td>35.1</td>
<td>34.9</td>
<td>30.4</td>
<td>30.1</td>
<td>32.6</td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{dev} classifier</td>
<td>33.3</td>
<td>34.9</td>
<td>35.4</td>
<td>34.6</td>
<td>34.3</td>
<td>30.2</td>
<td>29.9</td>
<td>32.3</td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{dev+lm} classifier</td>
<td>33.7</td>
<td>35.4</td>
<td>35.9</td>
<td>35.1</td>
<td>35.0</td>
<td>30.4</td>
<td>30.1</td>
<td>32.6</td>
</tr>
<tr>
<td>Oracle-domain</td>
<td>33.7</td>
<td>35.6</td>
<td>36.2</td>
<td>35.4</td>
<td>35.2</td>
<td>30.4</td>
<td>30.1</td>
<td>32.8</td>
</tr>
</tbody>
</table>

Table 7: English→Greek experimental results (truecase BLEU scores).

<table>
<thead>
<tr>
<th>System (En→El)</th>
<th>TED 2007</th>
<th>TED 2008</th>
<th>Europarl 2007</th>
<th>Europarl 2008</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>TED-tuned</td>
<td>28.2</td>
<td>25.3</td>
<td>24.6</td>
<td>26.3</td>
<td></td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>29.0</td>
<td>24.7</td>
<td>24.3</td>
<td>26.4</td>
<td></td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>28.9</td>
<td>24.6</td>
<td>24.2</td>
<td>26.3</td>
<td></td>
</tr>
<tr>
<td>Europarl-tuned</td>
<td>27.0</td>
<td>25.6</td>
<td>25.0</td>
<td>26.0</td>
<td></td>
</tr>
<tr>
<td>+ LM interp.</td>
<td>25.7</td>
<td>25.6</td>
<td>25.3</td>
<td>25.7</td>
<td></td>
</tr>
<tr>
<td>+ LM interp. + indicator feat.</td>
<td>25.7</td>
<td>25.6</td>
<td>25.1</td>
<td>25.6</td>
<td></td>
</tr>
<tr>
<td>Mixed-domain-tuned</td>
<td>27.5</td>
<td>25.6</td>
<td>25.0</td>
<td>26.2</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, LM classifier</td>
<td>28.9</td>
<td>25.6</td>
<td>25.1</td>
<td>26.8</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{train} classifier</td>
<td>28.9</td>
<td>25.6</td>
<td>25.1</td>
<td>26.8</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{train+lm} classifier</td>
<td>28.9</td>
<td>25.6</td>
<td>25.1</td>
<td>26.8</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{dev} classifier</td>
<td>28.8</td>
<td>25.6</td>
<td>25.1</td>
<td>26.7</td>
<td></td>
</tr>
<tr>
<td>Multi-domain, ME\textsubscript{dev+lm} classifier</td>
<td>28.9</td>
<td>25.6</td>
<td>25.1</td>
<td>26.8</td>
<td></td>
</tr>
<tr>
<td>Oracle-domain</td>
<td>28.9</td>
<td>25.6</td>
<td>25.1</td>
<td>26.8</td>
<td></td>
</tr>
</tbody>
</table>

On *TED*, our domain-adapted English→Italian and English→Portuguese systems outperform the best submissions from recent IWSLT evaluation campaigns by several BLEU points. On *News*, our domain-adapted English→German systems are on par with the best phrase-based system submissions at the WMT shared translation task.

**Mixed-domain vs. multi-domain SMT.** Looking at the performance on the concatenation of all test sets, mixed-domain SMT yields a higher BLEU score than any of the domain-adapted systems on two out of four language pairs (En→De: +0.3; En→It: -0.2; En→Pt: +0.9; En→El: -0.2). Apart from English→Portuguese, the differences are small.

Multi-domain SMT clearly outperforms mixed-domain SMT for English→Portuguese (up to +1.0 on all) and English→Greek (up to +0.6 on all). The choice of the domain classifier barely matters wrt. translation quality. Due to its compact model, the ME\textsubscript{dev} classifier would for instance be a reasonable choice despite not providing the highest classification accuracy.

---

8MT English→Italian: +4.3 points BLEU on tst2013 (Cettolo et al., 2013). MT English→Portuguese: +2.8 points BLEU on tst2014 (Cettolo et al., 2014).

9http://matrix.statmt.org
Figure 1: BLEU scores on a concatenation of all test sets.

Compared to oracle-domain SMT, which is equivalent to choosing the respective in-domain translation from the Domain-tuned + LM interp. + indicator feat. system, the best multi-domain results are on the same level of quality across the board, with a maximum drop of 0.2 points BLEU (En→Pt).

We visualized the results in a couple of plots (Figures 1-4). We use Domain-tuned as a shortcut for Domain-tuned + LM interp. + indicator feat. in all plots, i.e. the in-domain system results in the plots include LM interpolation and the provenance indicator. Multi-domain in the plots is the variant based on the ME\textsubscript{dev+lm} classifier.

BLEU histograms on the concatenation of all test sets are shown in Figure 1. The figure illustrates the results we just discussed, on the concatenation of all test sets. In Figures 2–4 we plotted average BLEU differences wrt. the in-domain system on the TED, Europarl, and News test sets. In terms of averaged BLEU scores over the in-domain test sets, in-domain systems are up to 7.8 points BLEU better than out-of-domain systems. Multi-domain SMT is up to 1.7 points BLEU better than mixed-domain SMT but can also perform minimally worse in some cases, for instance English→German TED and News, where mixed-domain SMT performs better than in-domain SMT. However, multi-domain SMT is typically on par with in-domain SMT.

8 Conclusion

While mixed-domain tuning worked for half of the language pairs, our results indicate that multi-domain SMT is the more reliable choice. Multi-domain SMT is always on par with in-domain SMT translation quality and under some circumstances mixed-domain SMT can perform much worse.

Multi-domain SMT can be easily implemented with a domain classifier and by allowing for run-time reconfiguration of the decoder with domain-specific weight vectors. Satisfactory domain classification accuracy can be achieved with a simple and compact maximum entropy text classifier trained on the small MT development sets and applied at the sentence level.

For scenarios where the model is expected to translate a wide variety of input text, the approach presented in this paper balances ease of implementation with high performance.

Acknowledgements

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements nº 644333 (TraMOOC) and 644402 (HimL).
German Italian
Avg. BLEU difference on TED wrt. in-domain system

Portuguese Greek
Avg. BLEU difference on TED wrt. in-domain system

Figure 2: Average BLEU differences on TED test sets.

German Italian
Avg. BLEU difference on Europarl wrt. in-domain system

Portuguese Greek
Avg. BLEU difference on Europarl wrt. in-domain system

Figure 3: Average BLEU differences on Europarl test sets.

Figure 4: Average BLEU differences on News test sets.
References


Niehues, J. and Waibel, A. (2012). Detailed Analysis of Different Strategies for Phrase Table Adaptation in SMT. In Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA), San Diego, CA, USA.


Korean-to-Chinese Word Translation using Chinese Character Knowledge

Yuanmei Lu  
lu@nlp.ist.i.kyoto-u.ac.jp  
Toshiaki Nakazawa  
nakazawa@pa.jst.jp  
Japan Science and Technology Agency, Tokyo, 102-8666, Japan

Sadao Kurohashi  
kuro@nlp.ist.i.kyoto-u.ac.jp  
Graduate School of Informatics, Kyoto University, Kyoto, 606-8501, Japan

Abstract

In this paper, we present a way of translating Korean words to Chinese using Chinese character information. A mapping table of Korean and Chinese characters is constructed and used to obtain possible combinations as translation candidates. The candidates are ranked by the combination score which accounts for the possibility of the character combination and context similarity score, which indicates contextual information among words. Parallel resources like Wikipedia aligned data or Wiktionary data are used for preliminary translation and also used during ranking candidates.

1 Introduction

The quality of the statistical machine translation heavily correlates with the amount of parallel corpora used, and improving the lexical coverage of the parallel corpora plays an important role in reducing the number of out-of-vocabulary (OOV) words. However, the vocabularies of languages keep growing over time, especially in the case of technical terms. It is impossible to cover all the newly appearing words by augmenting the parallel corpora and therefore we need to prepare bilingual dictionaries for the new words, or translate them separately, for example, using the transliteration technique.

There are some parallel dictionaries available for limited language pairs and limited domains. In addition, we can extract parallel resources from Wikipedia. It offers hyper-linked pages of the same topic in different languages, and the title pairs of the linked pages can be used as a parallel dictionary. However, the coverage is not sufficient for both cases especially for technical terms. In some cases, titles aligned by hyper-linked pages can not be seen as exact translations of each other since many of them may just be related to similar concepts. Another limitation in this domain is that although there may exist enough resources between English and other languages, there are very few resources between the non-English languages, such as Korean, Chinese and Japanese.

Korean, Chinese and Japanese use Chinese characters. In Japan, they use Kanji, which originated from China, while Korean uses Sino-Korean vocabularies, in which characters (Hangul) can be converted to corresponding Chinese characters (Hanzi). Just as Japanese writing includes both Kana and Kanji, Korean contains two kinds of characters–Hangul and Hanja. Hangul are Korean characters that are most commonly used and seen, and many Sino-Korean words have their Hanja writings–Hanja. For the sake of clarity refer to Table 1 which contains a few examples of Hangul and Hanja.
Table 1: Difference of characters written in Hangul, Hanja, Kanji and Hanzi

<table>
<thead>
<tr>
<th></th>
<th>Hangul (Korean)</th>
<th>Hanja (Korean)</th>
<th>Kanji (Japanese)</th>
<th>Hanzi (Chinese)</th>
</tr>
</thead>
<tbody>
<tr>
<td>애정 노동</td>
<td>愛情 労動</td>
<td>愛情 労働</td>
<td>愛情 労働</td>
<td>愛情 労動</td>
</tr>
</tbody>
</table>

In reality, Hanja writing is seldom used in modern Korean language. It is used in several technical writings to emphasize the definition of certain words. It is not hard to see that in many cases, the same Sino-Korean word may written in different Hanja that represent various meanings according to the context. Thus, the Hanja play important roles in word sense disambiguation (WSD), especially when it comes to terminology words.

Even though the forms are different, most of the vocabularies in these three languages (Korean, Chinese and Japanese) have one-to-one correspondence with respect to the characters. Using this characteristic, we can perform word translation and use the result to construct a terminological or scientific dictionary, which can further be used in machine translation systems for the above mentioned languages. However, there is neither a publicly available Hangul-Hanzi-Kanji mapping table, nor is there an official parallel dictionary between Sino-Korean and Chinese/Japanese (there exists an online dictionary for common words, but not for terminological words) on the web.

In this paper, we propose a method of translating Korean words (mainly terminological words) into Chinese and Japanese using the Chinese character knowledge. The input is Korean sentences, and our objective is the translation of terminological words in these sentences. We treat nouns as potential terminological words. A morphological analyzer is employed for extracting nouns, as the pre-processing of our model. We also construct a Hangul-to-Hanzi mapping table and use it to generate translation candidates, since Hangul and Hanzi have a one-to-one correspondence. Then we rank the candidates using the character combination and contextual similarity scores. We also apply a machine learning method for interpolating the two aforementioned scores of candidates.

2 Sino-Korean Words

2.1 Chinese Words in Korean and Japanese

In Asian languages like Korean and Japanese, majority of words are borrowed or adopted from other languages, especially when considering terminological words (Matsuda et al. (2008)). Apart from borrowing/adopting words phonetically like transliteration, a majority of words in these languages are originally borrowed from ancient Chinese. An example is the adoption of Kanji words (漢字語) of Japanese, or Sino-Korean words of Korean.

According to Studies on the Vocabulary of Modern Newspapers III published by National Institute for Japanese Language and Linguistics, Kanji (Chinese characters used in Japanese) accounts for over 70% of the readable content in newspapers (National Institute for Japanese Language and Linguistics (1972)). The Kanji words such as “使用 (use)” are preferably used than the native Japanese words such as “使う (to use)” in Japanese formal writings.

The situation is similar in Korean wherein a significant portion of the words are composed of Sino-Korean. The Standardized Korean Language Dictionary by National Institute of the Korean Language (NIKL) published in 2004 had 57% of its content as Sino-Korean words (鄭虎聲 (2000)); the Survey of Korean Vocabulary frequency, which was conducted in 1956, has shown that about 70% of the frequently used words are Sino-Korean (文敎部 (1956)). Nowadays, the percentage of Sino-Korean words of spoken language being generally used is grad-
ually decreasing but these kinds of words are still frequently used in formal writings, such as
ewspapers and dissertations. Most of the Sino-Korean words are not written in Hanja directly,
but in Hangul. However, we can convert them into Hanja and further to Hanzi because there
is a correspondence among them. Actually, some papers are published with combinations of
Hangul and Hanzi in order to specify definitions of vocabularies or emphasize them.

2.2 Related Work
There are some previous work on character conversion, both within language or between two
languages (Chen and Lee (2000), Huang et al. (2004)). During character conversion, a bilin-
gual dictionary is needed for candidate selection. For a low resource language like Korean,
a bilingual dictionary containing enough data, including polysemous words, is often hard to
obtain. Moreover, unlike sentence translation, character translation often ignores the context
information of the input source sentence.

Chinese character knowledge is widely used in cross-language information retrieval
(Hasan and Matsumoto (2000)), or translation of names of people (Wang et al. (2007, 2008,
2009)). During translation, they select named entities by removing the postpositions or the
endings, by applying the maximum matching algorithm. For Sino-Korean words that are writ-
ten using same Hangul word but expressing different meanings according to various context
environments (ambiguous words), they adopt some mutual information score to evaluate the
co-relation between the query term and the candidates. In languages such as Japanese and Ko-
rean, there is more ambiguity about where word boundaries should be, wherein some particles
may also be a part of a noun, or a verb, their method of extracting target words are often not
efficient.

Moreover, unlike information retrieval, the machine translation method also has to ensure
the meaning of the sentence in order to be fluent. In other words, we should also consider
context features of the sentences.

Since Korean characters are phonograms, we can find corresponding Hangul characters for
given Hanzi characters. Actually, almost all of the Hanzi can be converted to one (or in some
rare cases, many) Korean characters. Huang et al. constructed a Chinese-Korean Character
Transfer Table (CKCT Table) to reflect the correspondence between Hanzi and Hangul (Huang
and Choi (2000)). It is reported that the table contains 436 Hangul with corresponding 6763
Hanzi. Practically, there are 4888 common Hanja used in Korean (KATS (Korean Agency for
Technology and Standards) (1997), Hanyang Systems (1992)). Moreover, the number of daily-
used Hanzi in Korea numbers around 1800¹, while 3500 Hanzi are required to learn for practical
Chinese character level test². Obviously, most of the Hanzi in their table cannot be considered
as practical ones. After all, the table is non-public to ordinary users.

3 Proposed Method
Figure 1 gives an overview of our Korean-to-Chinese terminological word translation method-
ology using Chinese character knowledge. The determined translation result is marked in a
darker color. Since our objective is to perform terminological word translation, and in scientific
documents, terminological words are often Sino-Korean words, mostly nouns, we focus on the
translation of Korean nouns to Chinese.

Given a Korean sentence, we first need to extract the nouns for which we use morphologi-
cal analyzers to extract Korean nouns (Section 3.1). In the example in Figure 1, after morpho-
logical analysis, Korean words, 채광제, 臺, 情 are extracted from the input sentence. We

¹Wikipedia:상용한자
²Korea Foreign Language Evaluation Institute
then look up the Chinese translations of the Korean words in a Korean-Chinese parallel dictionary (Section 3.2). There are three different cases for a given word: unique translation (only 1 candidate per word), multiple translations (more than 1 candidate per word) and untranslatable. The words cannot be translated by the parallel dictionary are further processed wherein possible Chinese character combinations are generated as the translation candidates, using the Hangul-Hanzi mapping table constructed in Section 3.3. The candidates are ranked by two scores: combination score and context similarity score (Section 3.4). The combination score accounts for the possibility of the Chinese character sequences and is calculated using the Chinese web corpus, whereas the context similarity score considers the context feature of the input sentences and that of the sentences in the Chinese web corpus. Finally, we interpolate these two scores to handle the score of each translation candidate. The candidate with the highest score is considered as the final translation result. During the whole translation procedure, some data selection methods are involved.

3.1 Extracting Nouns

In Korean sentences, words are separated by spaces among them wherein each word may contain one or more morphological elements. On the other hand, a functional word may have different morphologies under different conditions Li et al. (2013). For example, in Korean sentences:
In the first sentence, the word “학교에” is composed of “학교 (noun)” and “에 (particle)”, while in the two sentences, “가” may act as either a verb (as in 가다) or a particle (as in 비가). To get terminologies from a sentence, some methods of extraction, which involves processes like tokenization and POS tagging should be implemented. However, there is no freely available tool to accomplish this task of extracting Korean terminological words. Instead, since most of the terminological words are nouns, we can use a Korean morphological analyzer to extract nouns from the given Korean sentences and treat them as terminological words.

To get terminologies from a sentence, some methods of extraction, which involves processes like tokenization and POS tagging should be implemented. However, there is no freely available tool to accomplish this task of extracting Korean terminological words. Instead, since most of the terminological words are nouns, we can use a Korean morphological analyzer to extract nouns from the given Korean sentences and treat them as terminological words.

For the given Korean sentences that contain Sino-Korean vocabularies, we extract nouns from these sentences with the help of a morphological analyzer, during which also performs word tokenization. After comparing several POS tag sets (KKMA (2011)) that mainly used for Korean language, we treat NNP and NNG as POS type for terminological words.

3.2 Translation by Dictionary Matching

Some of the Korean words are translated into Chinese with a parallel dictionary as an initial step. We use a dictionary of Wikipedia and Wiktionary aligned data to achieve this.

As a multilingual online encyclopedia, Wikipedia titles can be used as parallel data for many languages. In our model, we use the aligned Wikipedia title pairs of Chinese and Korean and Wiktionary data as a parallel dictionary. In addition, for aligned Wikipedia titles, we apply the following processes to improve the quality and coverage of the parallel dictionary.

- Make full use of redirect pages of each page, and validate the correctness using the first sentence (definition sentence) of each page to augment the parallel dictionary. Definition sentences of some pages containing more than one key words are analyzed to determine whether these key words are synonyms, by checking whether the definition sentence contains the word “또 는” and “혹은”, which means “or” between the key words. These synonyms have the same Chinese translation result.

- Convert Chinese characters of traditional Chinese into simplified Chinese for normalization. This step is done because translation result in dictionary matching step will be used as context feature of each of the character combination candidates (simplified Chinese characters).

On the other hand, we collect Sino-Korean words from Korean Wiktionary. Some of the Wiktionary pages have marked Chinese information which serves to indicate that the Chinese content is a translation of the title (語源: 漢字).

During the preliminary translation, some words may have more than one translation result (Chinese alignment). Some Wikipedia titles may contain brackets within them, the brackets often contain information for disambiguation. For example, consider a Korean word 전자. The word is contained in two titles 전자 and 전자 (전어학), having aligned Chinese “電子” (electricity) and “轉字” (transliteration), respectively. We combine titles of shared Korean words (without brackets) as translation candidates. The example above, 전자 has two aligned Chinese, 电子 and 转字 translations. Korean words that have more than one Chinese translations.
are considered ambiguous. In the Korean sentences we work with, we have Korean words belonging to both the above mentioned types. In the case of Korean words with unique Chinese translations, we do not need any further processing. However, in the case of multiple candidates, we compare the context vectors of the candidates calculated with the formerly translated results and determine the most appropriate candidate as the translation result. The Korean nouns which cannot be translated with the parallel dictionary are subjected to further processing.

3.3 Generating Translation Candidates

The purpose of this step is to generate possible translation candidates by combining Hanzi characters converted from the Hangul characters, using the Hangul-Hanzi mapping table. For instance, using the mapping table in Table 2, we can generate the translation candidates for the Korean word 한자 (Chinese character, 汉字) and the possible character combinations could be:

한자 (Chinese character, 汉字): 闲, 韩, ...汉字, ...汉字....

Whether these combinations have the proper meaning or not is still unknown. Most of them may not have practical meanings. So we need to select the most appropriate combination.

3.4 Rank the Translation Candidates

In this step, we utilize the segmented Chinese web corpus as a filter. First of all, we check the existence with the corpus and unify the POS type of input Korean words and their Hanzi combinations. According to Xia (2000), in Chinese, there are three POS tags for nouns: NR (Proper Noun), NT (Temporal Noun) and NN (Other Noun). Thus, according to these definitions, we extract all NN and NR type nouns from the segmented Chinese web corpus that contain POS information (see Shen et al. (2013)) as a Chinese noun corpus. After that, we generate all possible combinations and check whether they are included in this corpus. This step plays a significant role for reducing the cost of our model. In order to determine the most appropriate translations among the selected combinations, we utilize combination score and context similarity score.

3.4.1 Combination Score

Combination score $S_{comb}$ measures the strength of the link between the characters. Here, we utilize a language model to get the score. The language model returns the possibility (log score) of each combination according to the Chinese web corpus (see as equation (1)). We acquire the score of the original character sequence of the candidate and also the score for the reverse sequence. Equation (2) indicates the way the language model computes the score for the reverse sequence. Suppose that the combination consists of n characters, $c_{1..n}$

$$S(c_{1..n}) = \log \left( \prod_{i=1}^{n-1} P(c_i|c_{1..i-1}) \right)$$  \hspace{1cm} (1)

$$S(c_{n..1}) = \log \left( \prod_{i=1}^{n-1} P(c_i|c_{i+1..n}) \right)$$  \hspace{1cm} (2)

For example, the scores for “汉字语” and “语字汉” may be calculated as,

$$S(汉字语) = \log(P(汉) \times P(字 | 汉) \times P(语 | 汉字)),$$
\[ S(\text{语字汉}) = \log(P(\text{语}) \times P(\text{字} \mid \text{语}) \times P(\text{汉} \mid \text{语字})) \]

We combined these two scores by simply adding them. As the length of the word increases, the score would decrease drastically. We divide the sum with \textit{length-1} for normalization. The following formula defines the combination score. For words with many characters, the score will be much smaller after each multiplication and thus we divide the score with the length of the word. \( n \) here indicates the number of characters the word contains.

\[
S_{\text{combi}}(c_1 \cdots n) = \frac{S(c_1 \cdots n) + S(c_{n-1})}{n - 1}, \quad (n \geq 2)
\]

When \( n=1 \) (words with single character), we utilize unigram score as their combination scores. For example, the combination score for “汉字” may be calculated as

\[
S_{\text{combi}}(\text{汉字}) = \log(P(\text{汉}) \times P(\text{字} \mid \text{汉}) \times P(\text{汉} \mid \text{字}))
\]

Since the potential number of combinations may run into tens of thousands we conduct a selection again before moving on to the next step. We sort combinations according to their scores, and keep only the combinations whose scores are not lower than the highest score -2. For example, suppose \( S_{\text{combi}}(\text{汉字}) = -5.7126 \) is the highest one among the combinations, we remove combinations whose combination score is lower than -7.7126. If we still have many candidates, we keep top 10 candidates among them.

### 3.4.2 Context Similarity Score

Now that we have obtained candidates using the most possible combinations. To acquire the most proper ones for the given sentences, we consider context features. For each combination, context vector is constructed using the Chinese corpus. We use sentences which contain each combination as the context window, so the element of the vector is the co-occurrence Chinese characters (for character-based context vector) and frequency of them. We ignore 125 stop words (characters) such as 的 and 了. Some combinations that have higher frequency often contain bigger and wider range of count values in their context vectors. For words with this situation, co-occurrence characters with lower occurrence frequency are less effective as context information. Moreover, considering all of the co-occurrence characters is impractical since it will only increases complexity of time and space. We set the threshold as 100, that is to say, we ignore combinations that occur less than 100 times. Of course, there may also be some combinations such that all of their co-occurrence characters appear less than 100 times. For these candidates, we keep all these low-frequency characters as context information.

We also construct another context vector with the information of the input Korean sentence. We use the formerly dictionary translated Korean words (Section 3.2).

The context similarity score \( S_{\text{context}} \) is defined as the \textit{cosine similarity} of the two context vectors—one created using the Chinese corpus that contains Chinese sentences and the other using the translation results using the dictionary.

### 3.4.3 Interpolation

The combination score is useful to examine whether the combination is appropriate or not, and the context similarity score is helpful for selecting the the most appropriate one according to the context features where two or more combinations have practical meanings. Therefore, we used the former dictionary translated Korean words (Section 3.2).

The context similarity score \( S_{\text{context}} \) is defined as the \textit{cosine similarity} of the two context vectors—one created using the Chinese corpus that contains Chinese sentences and the other using the translation results using the dictionary.

---

\(^3\)https://code.google.com/p/verymatch/downloads/detail?name=stopwords.txt
interpolate two scores obtained in the former two sub chapters and calculate the score of each translation candidate $S(cand)$ as follows:

$$S(cand) = \alpha S_{combi} + (1 - \alpha) S_{context}$$

The specified value of $\alpha$ ($0 \leq \alpha \leq 1$) is determined using a method described in next chapter. The character combination with the highest score is regarded as the final translation result. If there are more than 2 words that cannot be translated by Wikipedia or Wiktionary, we process the procedure in sequence (from beginning to end of the sentence) and use the translation result as context feature to the remaining unknown words.

4 Experiment

4.1 Settings

Chu et al. had produced a Chinese character mapping table for Japanese (Kanji), Traditional Chinese (TC) and Simplified Chinese (SC) Chu et al. (2012). Thus, for constructing a table that contains mapping relationship between Korean Hangul and Chinese Hanzi, we need to construct rather Kanji-Hangul or Hanzi-Hangul tables and merge them. We collected Hangul-Hanja mapping information from the web.

- We acquired 1365 Hanja characters with their aligned Hangul from a freely accessible webpage\(^4\). These characters are contained by words whose frequency is higher than 5965 times in some Sino-Korean corpus (there is no specific information about the mentioned Sino-Korean corpus).
- There are also some materials that contain Hanja characters that are used in practice. We collected 3500 Chinese characters that are required for Hanja level tests PELT (2005), as mentioned in Section 2.2, to get most generally used Hanja characters with their aligned Hangul.
- As we mentioned previously, more data is needed to guarantee the coverage of the generally used Hanja characters. In order to collect as much data as possible about Hanja and their Hangul alignment, we crawled additional content from some Wikia pages\(^5\), which contain more than 20000 Hangul-Hanja correspondence. Of course most of them are not used in general.

We merged these data to get a Hangul-to-Hanja alignment, and combined this table with the one created by Chu et al. (2012). This combination is possible because most Hanja are traditional Chinese characters, and have one-to-one correspondences with Hangul. Hanja-Hangul tables mentioned above may contain a large amount of unused characters, which can affect time efficiency and the correctness of candidates. We checked the compatibility of the table with some Hanja dictionaries obtained from the web\(^6,\)\(^7\). There are also Hanja characters that cannot

\(4\)http://korean.nomaki.jp/site_j/kanji16.html
\(5\)사용자:Masoris/hani_converter.js - Wikia
\(6\)http://hanja.naver.com(데어버한자사전)
\(7\)http://small.dic.daum.net/index.do?dic=hanja(Daum한자사전)
Table 4: The α-Precision relation for each test set

<table>
<thead>
<tr>
<th>testset</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>α</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.44</td>
<td>0.44</td>
<td>0.20</td>
<td>0.44</td>
<td>0.44</td>
</tr>
<tr>
<td>Precision(%)</td>
<td>93.75</td>
<td>100.00</td>
<td>72.22</td>
<td>90.00</td>
<td>92.31</td>
</tr>
<tr>
<td></td>
<td>(15 / 16)</td>
<td>(5 / 5)</td>
<td>(13 / 18)</td>
<td>(9 / 10)</td>
<td>(12 / 13)</td>
</tr>
</tbody>
</table>

be converted to Hanzi and so we ignore them. This way, we can create a Hangul-Hanja-Kanji- Hanzi mapping table.

4.1.1 Experimental data
For our experiments, we obtained 100 Korean sentences from several technical documents (on natural science, such as biology, chemistry, physics, earth science) from the web, which contain 3281 words in total (1014 nouns, as translation object).

Totally, we have 5368 Hanzi characters map to 481 Hangul characters in our Hangul- Hanzi mapping table. The size of the aligned title dictionary is around 7.1MB (Wikipedia 6.6MB, Wiktionary 495KB) and that of the Chinese web corpus that contains ordinary Chinese sentences for calculating the context score and combination score of each translation candidate, is 47GB, among them are 184MB of NN, NR words. For querying the web corpus, we used the KenLM Heafield (2011) language model with smoothing technique included, ignoring start and end symbols and utilizes a character based process.

We used Google Translate as one of the baselines and compared the result with reference. We checked the precision with exact-match mechanism and obtained a precision of 38.17%.

4.2 Result
We manually set the reference translation results for the test data, and calculated precision for evaluation. To obtain the best value for α during interpolation, we tried 5-fold cross validation. We divided the test set into 5 parts and recursively selected four of them to get the α that gives the best score and used it on the remaining part to test the performance of the translation (see Table 4). We calculated precision of each test set for evaluation, and the best-performing α is the one with which we can get the highest precision. Figure 2 demonstrates the variation of precision with the change of α, for a randomly selected test set, testset 4.

We obtained an α for each test set, took the mean value and used it on the test sets again. The translation results are as shown in Table 5. (a) shows when we do not consider ambiguity of words in the dictionary, while (b) shows the results obtained when we determine ambiguous words with currently determined translation results (Chinese). The last columns of the two tables shows the result obtained with mean value of α, (α=0.18 in (a) and α=0.39 in (b)). Using the mean value of α, the precision dropped. (c) demonstrates the results when considering the scores separately as well as together. The column “+Combi” and “+Context” shows the translation result by only considering combination scores and context scores, separately, and “+Combi +Context” shows result by considering both combination scores and context features.

In another experiment, we utilized a machine learning algorithm that utilizes these features in order to get a better translation result.

We employed SVM rank, Support Vector Machine for Ranking. Taking a Support Vector approach, the resulting training problem is tractable even for large numbers of queries and large numbers of features Joachims (2002, 2006).

---

8 words are separated by space in the sentence
9 https://docs.google.com/spreadsheets/d/1hFlIJa7S5iSzIRp8NykJ8WRW68D3yRGlD4zeh2D4G4k/edit#gid=0
We used combination score of normal sequence and reversed one separately as features, with the third one as context similarity score. For the two combination scores, in order to pick candidates to process, we implemented a selection mechanism. We first added two scores, then sorted the sum and got the candidates among top 10 scores. We queried combinations scores of these candidates respectively and utilized as value of these two features.

On the other hand, the value of the third feature—context score, was calculated with formerly determined translation results in the given sentence. We again divided the sentences into 5 parts, and performed a 5 fold cross validation. As a result, the precision is 93.55%. This method outperforms the former one, is fast and has a low resource requirement.

4.3 Discussions

Among the 1014 words that is analyzed as nouns by the morphological analyzer, 59 are original Korean words, which are neither adopted words nor terminological words, 46 are transliterated words. Analysis of mistranslations is presented in Table 6.

When analyzing the experimental results, we found that among the 13 words that could not be translated, 11 of them are contain phonetic parts within them, such as 매카니즘 (mechanism), 시스템 (system). These words are untranslatable since they could neither be looked up in the dictionary (Wikipedia aligned data and Wiktionary data) nor did they have Hanzi mapping for the phonetic characters.

There is a compound Korean word “시양음악사학 (西洋音乐史学)”, consisting of three words, “시양 (西洋)”, “음악 (音乐)”,”사학 (史学)”. During the generation of Hanzi combination, the generated combination could not be found in the web corpus. We did use a corpus that contains compound words (nouns), and found that it contains “西洋音乐” and “音乐史学”, but no “西洋音乐史学” in it. For compound words that are space separated and do not have translation candidates in the dictionary, we checked the existence of their components in the dictionary and obtained candidates by combining the translations of the individual components. However, for compound words that contain more than 4 characters and do not have a space among them, the number of character combination increases drastically as n (length of the word) increases, which leads to an increase in space and time complexity. Thus, for words in this case, pre-processing during the morphological analysis phase gives better results. For example, for long words (containing more than 4 characters and do not have space among them), we can implement a split-and-analyze step before moving on to the translation step. We attempt to do further separation by adding spaces among characters and run the morphological analyzer again, until two divided parts are both analyzed as nouns. In this method, we successfully segmented “시양음악사학” into “시양” and “음악사학”, and confirmed that they can be translated by the
|                     | Dictionary | +Combination | +Context 
|---------------------|------------|-------------|-------------
| Correct             | 646        | 43          |             
| InCorrect           | 234        | 19          |             
| NoTransition        | 75         | 13          |             
| Precision(%)        | 73.41      | 69.35       |             
|                     | (646/880)  | (43/62)     |             

(a) Experimental results for not having consider ambiguous words in dictionary

|                     | Dictionary | +Combination | +Context 
|---------------------|------------|-------------|-------------
| Correct             | 549        | 190         | 50          
| InCorrect           | 35         | 106         | 12          
| NoTransition        | 371        | 75          | 13          
| Precision(%)        | 94.01      | 64.19       | 80.65       
|                     | (549/584)  | (190/296)   | (50/62)     

(b) Experimental results when considering ambiguous words in dictionary

|                     | + Combi | +Context | +Combi + Combi 
|---------------------|---------|----------|-----------------|
| Correct             | 44      | 47       | 50              
| InCorrect           | 18      | 15       | 12              
| NoTransition        | 13      | 13       | 13              
| Precision(%)        | 70.97   | 75.80    | 80.65           
|                     | (44/62) | (47/62)  | (50/62)         

(c) Experimental results when considering ambiguous words in dictionary (for each score)

Table 5: Experimental results

Wiki-dictionary (Wikipedia and Wiktionary).

In our experiments, instead of considering all combination candidates (obtained using the web corpus), we used several selection rules like: sort the candidates according to combination scores in the web corpus and select candidates to whose score differs by two points from the candidate with the highest combination score. From the candidates, we selected up to 10 candidates that have higher scores. We also utilize the segmented Chinese web corpus as a filter. We check the existence with the corpus and unify the POS type of input Korean words and their Hanzi combinations. However, this kind of filtering sometimes excludes “useful” candidates. For example, 유사점 is another word that could not be translated. All of the character combinations of it, including correct translation 类似点 is not contained in the web corpus, thus the word did not have any translation candidate. There are also other words, such as 영문자 and 상검가, which have candidates in the web corpus but correct translation is filtered during the procedure. It is mainly caused by the segmentation error of the web corpus.

Table 7 shows some good and bad example of translation results when using combination and context scores. The words being discussed are under-lined and the correct translations are marked with “*”. The table contains combination scores, context scores and interpolated candidate scores. Determined candidates by each score are marked in bold.
InCorrect trans
24 failed to find correct translation result in wik data
82 words in Wik dictionary
correct candidate has lower score
12 words consider combi & context sim

No trans result
2 have no combination as nouns in web corpus (NN,NR)
11 are transliterated words

Table 6: Error analysis of mistranslations

Good example:

<table>
<thead>
<tr>
<th>Korean</th>
<th>Candi</th>
<th>Combi</th>
<th>Context</th>
<th>Context+Combi</th>
</tr>
</thead>
<tbody>
<tr>
<td>현상</td>
<td>現象*</td>
<td>-10.5459</td>
<td>-2.2506</td>
<td>-5.4858</td>
</tr>
<tr>
<td>봉상</td>
<td>-10.3712</td>
<td>-3.4574</td>
<td>-6.1538</td>
<td></td>
</tr>
</tbody>
</table>

Bad example:

<table>
<thead>
<tr>
<th>Korean</th>
<th>Candi</th>
<th>Combi</th>
<th>Context</th>
<th>Context+Combi</th>
</tr>
</thead>
<tbody>
<tr>
<td>감도</td>
<td>感度*</td>
<td>-12.1544</td>
<td>-2.0072</td>
<td>-5.9646</td>
</tr>
<tr>
<td>感到</td>
<td>-10.9311</td>
<td>-2.0130</td>
<td>-5.4910</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Examples of good and bad translations.
(* indicates correct translation result)

5 Conclusions

In this paper, we present an automatic Korean-to-Chinese terminological translation mechanism that uses Chinese character knowledge. The ultimate goal of this work is to use the translation result for constructing useful resources in machine translation between Korean and Chinese. A morphology analyzer was used to extract nouns as Sino-Korean words. We used aligned Wikipedia title data and Hangul-Hanja information in Wiktionary to obtain reference translations. Some polysemous words may have more than one Chinese translation in the dictionary. To select the most proper one, we compared their context features within the sentence. In order to rank candidates, we both considered context information and probabilities of occurrence in the web corpus. We carried out character-based translation, for which segmented Chinese web corpus is used to create a character-based and word-based context vector for each translation candidate. Some incorrect reference translation results (insufficiency in alignment data or failed to be selected correctly with context similarity) caused some incorrect translation results. Moreover, some candidates which are correct translations were excluded from consideration since have too lower combination score among the candidate set.

In the future, we intend to determine candidates that are less ambiguous and easy to determine during candidate selection, to guarantee more contextual information. Moreover, we want to evaluate the proposed approaches on larger test data.
References


Topic Adaptation for
Machine Translation of E-commerce Content

Prashant Mathur prashant@fbk.eu
Marcello Federico* federico@fbk.eu
FBK, Trento, Italy
Selçuk Köprü skopru@ebay.com
Sharam Khadivi skhadivi@ebay.com
Hassan Sawaf hsawaf@ebay.com
eBay Inc., San Jose (CA), USA

Abstract
We describe effort to improve machine translation of item titles found in a large e-commerce inventory through topic modeling and adaptation. Item titles are short texts which typically contain brand names that do not have to be translated, and item attributes whose translation often depends on the context. Both issues call for robust methods to integrate context information in the machine translation process in order to reduce translation ambiguity. We survey both existing topic adaptation approaches and propose novel methods that augment the standard phrase-table models with sparse features and dense features measuring the topic match between each phrase-pair and the input text. We report extensive experiments on the translation of item titles from English into Brazilian Portuguese, and show the impact of topic adaptation both with and without domain adaptation.

1 Introduction
Domain adaptation and topic adaptation can be seen as complementary methods to cope with the variability between training and testing data in machine translation. Under this perspective, domain modeling typically assumes training data partitioned according to human defined labels, while topic modeling builds on fuzzy clustering of the data and automatically learned labels. In statistical machine translation, knowledge about the domain or topics of the input can be leveraged to bias the system towards training data matching the same labels of the input. Potential advantages of topic adaptation over domain adaptation is that fuzzy clustering can better cope with data sparseness than hard clustering, and that automatic labels do not require any manual intervention. On the other side, however, domain adaptation becomes difficult to beat when training data can be effectively and naturally partitioned into in-domain and out-of-domain data.

In this paper we discuss the application of domain and topic adaptation to an e-commerce online MT system (Guha and Heger, 2014), whose target is the translation of user queries and all item titles, descriptions, and specifics shown in the search result pages. In particular, our investigation focuses on the translation of item titles, which consists of concise user-generated texts describing each item put on sale. Item titles differ in several ways from text genres typically considered in machine translation research. Titles are usually short texts, of maximum

*Most of the work was carried out during a visiting period of the first two authors at eBay Inc.
100 characters, with a simple syntactic structure, and containing brand names, feature values, as well as specific jargon. Their translation poses several challenges (Sanchez and Badeka, 2014), such as the correct rendering of proper names, which can often be confused with common names, and the correct translation of product features, which often depends on the context. From a statistical learning perspective, MT of item titles is also hard because of the large variety of content present in eBay’s inventory, which are very unevenly populated but also significantly overlapping in terms of linguistic content.

The idea that we follow in this work is to employ topic modeling to better translate English item titles to a foreign language. Since we have a relatively small amount of bilingual in-domain data compared to bilingual out-of-domain data and English in-domain monolingual data, we aim to apply topic adaptation on the bilingual data and to train a topic model on the monolingual data (see Section 2). Then, we enrich in-domain and out-of-domain parallel data with topic information and embed this in the translation model of the MT system. At testing time, we infer the topics of the input and use them to dynamically adapt the MT system. In this work we survey and compare different topic adaptation methods from the recent literature and measure their impact on translation performance with and without domain adaptation. We report experimental result with a Moses-based phrase-based system on the English - Brazilian Portuguese language pair. This particular pair is one of the active language pairs at eBay. While for domain adaptation we deploy a state-of-the-art method, for topic adaptation we investigate the use of new sparse features which we compare against other features proposed in the literature.

The rest of the paper is arranged as follows. We first introduce topic modeling applied to our e-commerce content in Section 2, then we survey previous work on topic adaptation for statistical MT in Section 3, afterwards we present our take on topic adaptation in Section 4, and finally we report on our experimental set-up and results in Sections 5 and 6. We end the paper with some conclusions and future work.

2 Topic Modeling

2.1 Content

In eBay, machine translation plays an important role to facilitate cross-border trade between sellers and buyers with different languages (Guha and Heger, 2014). eBay is a marketplace where sellers can advertise their items on the site and buyers can search for the items and then electronically bid for them. To enable a trade between buyers and sellers with different languages, at least four types of texts need to be translated: queries, item titles, descriptions, and item specifics. This work focuses on the translation of item titles, which are concise and usually very informative descriptions of the items put on sale. For instance, the item title:

\textit{new men's white jekyll & hyde jeans winston designer regular fit shirt size s-xxl}

specifies, in order, the condition, target gender, color, brand, designer, fit, item type, and size of the product. Common challenges in the translation of eBay’s user generated content in general, and of titles (Sanchez and Badeka, 2014) and queries (Picinini, 2014) in particular, are the proper rendering of proper names and the translation of words which can have multiple senses, depending on the context in which they appear. For example, the word “age” might have different meanings if context is “baby”, “wine” or “collectibles”. Similarly, the word “j-hook” might have different meanings if context is “motors” or “garden”.

The core idea of this work, hence is to apply topic modeling to efficiently represent the context of single words or expressions in order to improve the accuracy of their translation by a phrase-based statistical MT system. In the following, we describe the monolingual data and the topic model that we used for this purpose.
2.2 Sampling

The amount of monolingual item titles available to eBay is huge and very unevenly distributed across different levels of categories of eBay’s inventory. Actually, items in eBay’s inventory are classified according to a hierarchical taxonomy. The hierarchy itself contains 34 top-level categories (L1) with varying degrees of depth in each category (L2=400 and L3=4000 categories). For example, the top level category “Books” has eleven second-level categories (L2) and each of those categories have anywhere from four to thirty categories, with many of these having subcategories as each topic becomes more and more specific. All traded items are placed in the leaves of this hierarchy.

For the sake of experimentation, we first perform sampling in order to collect a more balanced and manageable collection of titles. Hence, starting from a collection of billions of item titles, after stratified sampling from each L2 category we end up with a collection of 4.3M item titles. Using a uniform sampling method, we further sub-sampled data (from each L1 category) to around 708K item titles, which is actually used to train the topic model.

2.3 Models

Topic models can be trained from an arbitrary document or sentence collections with different methods, such as probabilistic latent semantic analysis (Hofmann, 1999), hidden topic Markov models (Gruber et al., 2007), and latent Dirichlet allocation (LDA) (Blei et al., 2003). For all methods, there are existing software tools that allow to train topic models after specifying the desired number of topics and a few training options. All tools permit then to use a training model to infer a topic distribution for a given sentence. In our case, we deployed a LDA model trained with the Standford TMT\(^1\) tool. In particular, we worked with a training configuration using 30 topics (similar to size of L1 categories) and 1000 iterations. We also experimented with different number of topics but empirically found 30 to be the optimum number. Moreover, we excluded all item titles with less than 5 words, all words occurring less than 3 times, and all words made of less than 3 characters. The pruning steps were performed to exclude from the model item titles providing too little context, words which are too infrequent, and very frequent and short words that to not bear any topic information.

In the following table, we report the 15 most relevant words of the first 10 topics trained with the LDA model. As can be seen, some of the topics are easily recognizable, e.g. T01 (toys), T04 (electronics), T05 (photo), and T09 (fashion), and T10 (hunting and fishing). The other topics look instead combinations of multiple categories, e.g. T06 seems a combination of pet supplies and fashion.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>T01</td>
<td>doll high barbie monster little dolls girl lot fashion dress pony and american hair with</td>
</tr>
<tr>
<td>T02</td>
<td>free shipping fish aquarium beads tank diy ship round glass pcs wholesale plastic water craft</td>
</tr>
<tr>
<td>T03</td>
<td>screen car baby lcd seat glass replacement touch protector cover cloth diaper safety holder glasses</td>
</tr>
<tr>
<td>T04</td>
<td>digital control module with remote power sensor switch board lcd meter system arduino air kit</td>
</tr>
<tr>
<td>T05</td>
<td>camera lens canon nikon mount digital video with sony gopro dslr camcorder adapter black hero</td>
</tr>
<tr>
<td>T06</td>
<td>dress pet dog clothes coat long winter size sweater women puppy warm apparel jacket fashion</td>
</tr>
<tr>
<td>T07</td>
<td>nail set art brush hair color makeup gel eye kit cream polish powder tool skins</td>
</tr>
<tr>
<td>T08</td>
<td>figure anime movie poster action disney hot mask toys sex toy series japan prop batman</td>
</tr>
<tr>
<td>T09</td>
<td>size black jacket mens shoes leather boots large blue womens nwt medium ski white men’s</td>
</tr>
<tr>
<td>T10</td>
<td>knife steel tool stainless set blade folding with pocket gun black tools handle hunting fishing</td>
</tr>
</tbody>
</table>

Table 1: LDA topic model of item titles: top 15 relevant words of the first 10 topics.

Finally, by using the same tool, the topic model is applied to annotate with topics all the

\(^1\)http://nlp.stanford.edu/software/tmt/tmt-0.4/
sentences in the source side of all the available parallel data. As a result, each sentence of the parallel training data is associated with a topic vector or distribution.

3 Related Work

Topic models have been investigated in the statistical MT literature to enhance both linear or hierarchical phrase-based models with additional context-topical information derived from the training and testing data.

Previous works have approached this issue under different perspectives, also providing different levels of integration of topic information. For instance, in (Gong et al., 2011) and (Ruiz and Federico, 2011) authors devised and exploited cross-lingual topic models to generate topic relevant target words for an entire document or sentence. This differs from other approaches and our one, which instead exploit monolingual topic models in the source language to directly enhance the selection process of single phrase translations pairs during decoding.

In (Eidelman et al., 2012), topic dependent lexical probabilities are directly integrated in a hierarchical phrase-based system. The authors start with topic distributions inferred at the document level on the source side of the parallel data. After extraction of the translation rules, topic-conditional translation probabilities are inferred by computing the expectation over the topic vectors observed in all the sentences where each translation rule was extracted from. At decoding time, these probabilities are weighted by the topic prior inferred on the test document. This results in a set of sparse features, one per topic, which are tuned on a development set. In this work, we implement and evaluate a similar set of sparse features for a phrase-based decoder.

In (Su et al., 2012) topic models are used to off-line adapt a phrase table trained on out-of-domain parallel data by using in-domain monolingual data. Topic distributions are inferred through hidden topic Markov models trained on monolingual sentences. Two distinct topic models are trained, one in-domain and one out-of-domain, which are mapped via a mixture model. Finally, similar to (Eidelman et al., 2012) topic-conditioned phrase translation probabilities trained on out of domain data are weighted with the in-domain topic prior probabilities. Contrary in our case, topic information is integrated just in the training phase to bias the translation model toward the in-domain data. Our purpose, instead, is to dynamically adapt the translation model at testing time.

In (Xiao et al., 2012) monolingual topic models are trained on the source side with LDA and topic vectors are associated to hiero rules. Differently from (Eidelman et al., 2012), during decoding topic posterior distribution on phrase-pairs are matched against the topic distribution of the test sentence. Matching is performed with the Hellinger divergence. In addition, a feature measuring topic sensitivity of each phrase-pair is included based on the entropy function. In our work, we integrate and compare several dense features that compute the match between the topic distributions of the input and of each phrase-pair candidate. Among them we also include the two features proposed by this work.

In (Hewavitharana et al., 2013) topic adaptation is performed in context of machine translation of task-driven conversations. Topics vectors are inferred via LDA on the source side of in-domain parallel training data. At test time, the topic vector of the conversation is incrementally updated at each turn. During decoding, with a Moses-like phrase-based system, each candidate phrase-pair activates a feature function measuring the highest similarity between the current topic vector and all topic vectors associated to the occurrences of the phrase-pair in the training corpus. Similarity is computed by taking the complement of the Jensen-Shannon divergence. In our work, we also deploy this divergence measure to measure the match between the topic vector of the input and the topic vector of each candidate phrase-pair.

Finally, in (Hasler et al., 2014a) the authors combine and compare domain adaptation and
topic adaptation in phrase-based statistical MT for the translation of texts from three different domains. Concerning topic adaptation, the standard Moses phrase-based feature functions associated to the phrase-table are augmented three sets of dense feature functions: (i) two translation probabilities, (ii) one language model score and (iii) three topic distribution similarity feature functions. The first set of features introduce source-to-target phrase probabilities that account for topic information, the second set scores unigrams of the target phrase according to their topic relevance, and the latter measures the similarity between the input topic distribution and the topic distribution associated, respectively, to the whole phrase-pair, to the target phrase, and to the most representative target of the phrase. Topics distributions are inferred similarly to (Eidelman et al., 2012) at the level of whole speeches. Particular care is taken about sampling data of different domains, to avoid domain biases, as well as sampling the same amount of context data for each phrase-pair, to not avoid context bias. The authors main conclusion is that topic adaptation helps especially if there is a high divergence between training and testing domains. Moreover, topic vectors can be helpful also to predict the domain of test data when topic domain vectors are used as proxy of data. As is the case with Hasler et al. (2014a), we also compare topic adaptation with domain adaptation, but with respect to topic adaptation our main emphasis is on the combination of sparse and dense features.

4 Topic Adaptation

4.1 Approach

We apply the topic model discussed in Section 2 to infer the topic distribution on the source side of all bilingual training data of our statistical MT system (details will follow in Section 5), on the development set and on the evaluation set. Notice that we inferred the topic distribution on out-of-domain training data at the sentence level rather than at paragraph or document level. This choice is to keep annotation consistent with the training and testing conditions of the topic model, which are performed on item titles.

Similar to previous works, we trickle-down topic information from the sentence level to the phrase-pair level. By borrowing the notation from (Hasler et al., 2014b), we estimate the probability \( p(t|s,k) \) of a target phrase \( t \) given a source phrase \( s \) and the topical information \( k \) \((k \in 1 \ldots K)\) where \( K \) is the total number of topics), through the formula:

\[
p(t|s,k) = \frac{\sum_d p(k|d) \cdot c(t,s;d)}{\sum_t \sum_d p(k|d) \cdot c(t,s;d)}
\]

Where \( c(t,s;d) \) is the count of target phrase \( t \) and source phrase \( s \) being extracted from sentence (our proxy for document) \( d \) in the training data, and \( p(k|d) \) is the probability of topic \( k \) in sentence \( d \) in the training data.

In addition to the probabilities computed with (1), we also infer topic vector (distribution) \( \phi_{s,t} \) for each phrase-pair \( (s,t) \) extracted from the training data. Each component of the vector is the probability \( p(k|s,t) \), which is inferred by averaging over all topic vectors corresponding to the sentences \( (d) \) in which the phrase pair was extracted from.

For each phrase pair we only keep the relevant topics and set the probability of the other topics to zero. In particular, we compute the perplexity\(^2\) \((PP)\) of the topic distribution and keep only the most probable \([PP]\) topics. In general, the perplexity tells how many equally likely topics can be represented with the number of bits of an optimal encoding of a topic distribution. We have empirically observed that the ceiling of the PP typically identifies the point in the ranked list of topics after which there is a significant drop in probability. Moreover, \(PP\) gives us also a measure of the topic specificity of a phrase pair. In particular, the lower

\(^2\)Definition and properties of the perplexity function can be found for instance in (Federico and De Mori, 1998).
the perplexity of the topic distribution is, the more topic-specific is the translation represented by the phrase-pair phrase. On the contrary, phrase-pairs with high perplexity values reflect translations observed in sentences from many different topics, and thus identify translations not weakly depending on their context. Hence, we expect that the translation model will mostly benefit from topic information only for phrase-pairs with low perplexity. For this reason, we add topic information \( p(t \mid s, k) \) and \( p(k \mid s, t) \) to every entry \( s, t \) of the phrase-table only when the perplexity of \( \phi_{s,t} \) is below a given threshold \( PP_{\text{max}} \) and up to \( PP \) topics.

The information entered in the phrase-table is exploited to activate feature functions at test time, by combining them with topic information inferred on the input sentence at test time. Similarly to (Hasler et al., 2014b) we denote the input topic information with the vector \( \phi_c \), where \( c \) stands for context representation. Notice, that we apply the same topic pruning strategy explained before also for the input topic vector.

In the following, we present the list of feature functions that we have explored in this work.

4.2 Features

1. **Joint Probability**: The topic probability \( p(t \mid s, k) \) in itself is not enough to disambiguate between the context. As an example, assume that our English-Portuguese phrase table contains the following two phrase pairs with corresponding probabilities of target given source and topic:

(a) \( \text{age} ||| \text{idade} ||| \text{topic14 0.6 topic04 0.2} \)
(b) \( \text{age} ||| \text{era} ||| \text{topic25 0.5 topic14 0.3} \)

The two entries show two different translations of the English word *age*, whose probabilities vary according to the context they have been observed with. In particular, the translation *idade* has been observed with *topic14* and *topic04*, while the translation *era* has been observed with *topic25* and *topic14*. However, this information can be properly exploited only once the topic information of the input sentence is known.

Let us assume the following two input sentences and (pruned) context vectors:

(a) *early english bronze age period blade c. 1600 bc* <topic25 0.9>
(b) *supre hempz age defying moisturizer- 1 bottle* <topic14 0.7>

Our first feature function activates for each translation option a sparse feature for each topic that occurs in both the context and the phrase-table with value:

\[
 f^k_{1}(t, s, c) = -\log p(k \mid c) \cdot p(t \mid k, s)
\]

Going back to the first input sentence of our example, translation of *age* with *idade* would not activate any sparse feature, while translation of *age* with *era* would activate sparse features *topic25* with score \(-\log(0.45)\). For the second input sentence, both translations of *age* with *idade* and *era* would activate sparse features *topic14* with scores \(-\log(0.42)\) and \(-\log(0.21)\), respectively. The plain interpretation of this example is that our sparse feature function only rewards the translation *era* for the first sentence, while for the second input sentence it rewards both translations but gives a higher score to the translation *idade*.

Notice, that our sparse feature is derived from the feature that Eidelman et al. (2012) proposed for hierarchical phrase-based decoding. This feature has a clear probabilistic interpretation: the product \( p(k \mid c) \cdot p(t \mid s, k) \) computes the joint translation-topic probability \( p(k, t \mid s) \) by multiplying the translation probability conditioned to topic with the prior topic probability coming from the context.
2. **Geometric Mean**: The same principle of 1. applies with our second basic feature, but this time with the phrase table annotated with posterior probabilities $p(k|s, t)$. Again, for each phrase-pair and for topics present both in the phrase table and in the context, our sparse feature takes the product of the topic probability on the input and the topic probability of the phrase pair.

$$f_k^2(t, s, c) = -\log p(k|c) \cdot p(k|t, s)$$

The intuitive interpretation behind this feature is to measure the level of each matched topic with the geometric mean between the probabilities in the context and phrase-table. Notice that the above expression is equivalent to the log of the geometric mean but a constant factor 0.5 which we assume to be absorbed by the feature weight.

While the previous features are sparse, in the sense that they are computed over single topics thus resulting in $K$ distinct features, the following feature are dense, i.e. it is a single feature is computed for all topics. In particular, all features try to measure the similarity between the topic distributions on the input and on each phrase pair.

3. **Crude-Count**: In this feature, we simply count the number of topics active in both input and the phrase pair and normalize this count over the total number of topics in both the context and the phrase-pair. Formally:

$$f_3(t, s, c) = -\log \frac{|\{k: p(k|c) \cdot p(k|t, s) > 0\}|}{|\{k: p(k|c) > 0\}| + |\{k: p(k|t, s) > 0\}|}$$

Notice that this feature as well as the remaining features are activated for phrase-pair only if there is at least one common topic between context and the phrase-pair.

4. **Cosine Similarity**: This feature computes the cosine similarity between the input topic vector ($\phi_c$) and the phrase-pair topic vector ($\phi_{t,s}$).

$$f_4(t, s, c) = -\log \cos(\phi_c, \phi_{t,s})$$

This feature was proposed in (Hasler et al., 2014b).

5. **JS Divergence**: This feature computes Jensen-Shannon divergence between input topic vector ($\phi_c$) and phrase-pair topic vector ($\phi_{t,s}$).

$$f_5(t, s, c) = -\log JS(\phi_c, \phi_{t,s})$$

This feature was proposed in Hewavitharana et al. (2013).

6. **Hellinger Divergence**: Finally, this feature computes the similarity with Hellinger’s divergence.

$$f_6(t, s, c) = -\log HD(\phi_c, \phi_{t,s})$$

$$HD(\phi_c, \phi_{t,s}) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{\phi_c,i} - \sqrt{\phi_{t,s},i})^2}$$

This feature was proposed in (Xiao et al., 2012).

7. **Sensitivity**: We measure how sensitive the phrase pairs are to the topics. Here, the idea is to penalize the phrase pairs with the topic vectors of high entropy. High entropy of a topic vector denotes that the phrase pair is susceptible to multiple topics, which means it occurs in multiple context and hence topic vectors are not useful for any disambiguation.

$$f_7(t, s, c) = -\log H(\phi_{t,s})$$

This feature which can be computed offline, as it does not depend on the context, was also proposed by Xiao et al. (2012).
The first two sparse features and the other dense features are linearly combined with standard features used in phrase based decoding. Hence, with the notation \( f_1 + f_3 + f_4 + f_5 + f_6 + f_7 \) we indicate that the standard translation score computed by the decoder is augmented with:

\[
\sum_{k=1}^{K} \lambda_{1,k} f_k^1(s,t) + \sum_{i=3}^{7} \lambda_{i} f_i(s,t) \tag{3}
\]

where the \( K + 5 \) \( \lambda \)-weights are tuned together with the other weights of the translation model.

### 5 Experiments

#### 5.1 Task and Data

We evaluated our topic adaptation approach on the translation of item titles from English to Portuguese (Brazilian). Parallel data used to train, tune and evaluate MT systems comes from various publicly available collections, proprietary repositories and in house translated item titles. In particular, in-house translated items, descriptions, and specifics are here considered as in-domain data while all the rest is regarded as out of domain data. For development and testing purposes we use manually translated item titles for which two reference translations are available. Statistics on the amount of parallel data for each category are given in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Train (Out-Domain)</th>
<th>Train (In-Domain)</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td>5.28M</td>
<td>336K</td>
<td>1631</td>
<td>1000</td>
</tr>
<tr>
<td>Tokens EN</td>
<td>69M</td>
<td>2M</td>
<td>27K</td>
<td>10K</td>
</tr>
<tr>
<td>Tokens PT</td>
<td>70M</td>
<td>2M</td>
<td>31K</td>
<td>11.6K</td>
</tr>
</tbody>
</table>

Table 2: Statistics of English-Portuguese parallel data.

#### 5.2 MT Systems

This section describes the topic adapted MT systems and the two baseline MT systems developed for comparison purposes. All MT systems are built using the Moses toolkit (Koehn et al., 2007) and the linear weights for all systems are optimized using the k-best batch MIRA implementation provided in the Moses toolkit (Cherry and Foster, 2012). Performance of all systems are reported in terms of BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) scores. Statistical significance tests were conducted using approximate randomization tests (Clark et al., 2011).

**Baseline System** In-domain and out-domain parallel data were taken in 10:1 ratio for training the word alignments. Translation models along with operation sequence models (Durrani et al., 2011) were trained using the standard pipeline of Moses. Due to the nature of the item titles, we did not use any lexicalized reordering model in the MT system. The distortion limit was set to 6. On the target side, we built a trigram LM, using KenLM (Heafield, 2011) trained with modified Kneser-Ney smoothing (Chen and Goodman, 1996).

**Domain Adapted System** The domain adapted system was built on top of the baseline system. An additional translation model was built using the in-domain data and then the fill-up adaptation method (Bisazza et al., 2011) was applied to combine the in-domain and out-domain phrase tables. Fill-up simply adds a provenance feature in the phrase table with a score of 1 if the phrase pair is present in in-domain phrase table and 0 if it is from out-domain phrase table.
**Topic Adapted Systems** To evaluate the performance of the topic adapted systems using the features functions presented in Section 4.2 we followed a component analysis approach. Each basic sparse feature was added to the domain adapted system separately, shortly (1) DA+f1, (2) DA+f2. We built two separate systems because when we tag the topics in the phrase table, we can either set them to a distribution of topics over phrase pair \( P(k|s,t) \) or a target-phrase translation probability given the source phrase and the topic \( P(t|s,k) \). Then we added the other dense features one by one, resulting in 10 distinct systems i.e. 1) DA+f1+f3 2) DA+f1+f4 3) DA+f1+f5 4) DA+f1+f6 5) DA+f1+f7 6) DA+f2+f3 7) DA+f2+f4 8) DA+f2+f5 9) DA+f2+f6 10) DA+f2+f7. Finally, we also combined all dense features together on top of basic sparse features to build 1) DA+f1+f3+f4+f5+f6+f7 2) DA+f2+f3+f4+f5+f6+f7. To evaluate the impact of topic adaptation independently from domain adaptation we performed the same analysis also with the baseline (BA) system.

6 Results and Discussion

6.1 Topic Model Analysis

Since, our method of topic adaptation depends on the quality of topic labels inferred on training data, development and evaluation sets; first, we analyze the distribution of topic labels. Figure 1 shows the average probability mass for each topic in three sets. The plot shows that topics represented in the training are not always very well covered in the development and evaluation sets. Topics 14, 15 and 26 are very frequent in the training data, as a result when we project the topic distribution to phrases in the phrase table these topics occur in more phrase pairs than any other topic. Thus, as a consequence, these topics should have less discrimination power than other topic features. To validate this hypothesis we also plotted the weights of the topic features f1 after tuning them with k-batch MIRA. In fact, we can observe that the algorithm weights them less compared to other topic features as shown by green lines in the figure.

An interesting note is that for some topic labels the tuning algorithm assigns negative weights. A possible reason is that there is a topic-translation mismatch between training and development data. This can be explained by the fact that the training data contains both in-domain and out-domain data while the development set contains only in-domain data. Hence, the possibility is that topics mostly occurring in phrase-pairs extracted from out-domain data are being penalized to the advantage of topics mostly occurring in in-domain phrase-pairs.

![Figure 1: Topic distribution on training, development, and test data set. Green bar shows the weight of each topic feature when tuned with MIRA.](image)
6.2 MT experiments

Table 3 presents the results for the baseline, domain adapted and all topic adapted systems. We split the results in three blocks: the first block of results shows the impact of adding sparse topic features \((f_1 \text{ and } f_2)\) on top of the baseline (BA) system. The second block shows results with the Joint Probability feature and the third shows results with the GeoMean feature. In the first block, both features improve performance consistently (at least 0.3 gain in BLEU points) over the baseline system and we even observe statistical significant improvements in BLEU score with the Joint Probability feature \((f_1)\). Since, our goal is to improve over domain adaptation we tested the rest of the features only on top of the domain adapted (DA) system.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (BA)</td>
<td>36.99</td>
<td>49.15</td>
</tr>
<tr>
<td>BA + (f_1)</td>
<td>37.52</td>
<td>49.09</td>
</tr>
<tr>
<td>BA + (f_2)</td>
<td>37.28</td>
<td>48.93</td>
</tr>
<tr>
<td>Domain Adapted (DA)</td>
<td>39.42</td>
<td>48.12</td>
</tr>
<tr>
<td>DA + (f_1)</td>
<td>39.56</td>
<td>48.11</td>
</tr>
<tr>
<td>DA + (f_1 + f_3)</td>
<td>39.66</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_1 + f_4)</td>
<td>39.70†</td>
<td>47.75†</td>
</tr>
<tr>
<td>DA + (f_1 + f_5)</td>
<td>39.56</td>
<td>48.84</td>
</tr>
<tr>
<td>DA + (f_1 + f_6)</td>
<td>39.66</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_1 + f_7)</td>
<td>39.60</td>
<td>48.84</td>
</tr>
<tr>
<td>DA + (f_1 + f_3 + f_4 + f_5)</td>
<td>39.67</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_1 + f_3 + f_4 + f_5 + f_6 + f_7)</td>
<td>39.57</td>
<td>47.72†</td>
</tr>
<tr>
<td>Domain Adapted (DA)</td>
<td>39.42</td>
<td>48.12</td>
</tr>
<tr>
<td>DA + (f_2)</td>
<td>39.60</td>
<td>47.86</td>
</tr>
<tr>
<td>DA + (f_2 + f_3)</td>
<td>39.61</td>
<td>47.79</td>
</tr>
<tr>
<td>DA + (f_2 + f_4)</td>
<td>39.60</td>
<td>47.76†</td>
</tr>
<tr>
<td>DA + (f_2 + f_5)</td>
<td>39.62</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_2 + f_6)</td>
<td>39.60</td>
<td>47.78</td>
</tr>
<tr>
<td>DA + (f_2 + f_7)</td>
<td>39.66</td>
<td>47.68†</td>
</tr>
<tr>
<td>DA + (f_2 + f_3 + f_4 + f_5)</td>
<td>39.71†</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_2 + f_3 + f_4 + f_5 + f_6 + f_7)</td>
<td>39.53</td>
<td>47.68†</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain Adapted (DA)</td>
<td>39.42</td>
<td>48.12</td>
</tr>
<tr>
<td>DA + (f_1 + f_3)</td>
<td>39.60</td>
<td>47.86</td>
</tr>
<tr>
<td>DA + (f_1 + f_4)</td>
<td>39.61</td>
<td>47.79</td>
</tr>
<tr>
<td>DA + (f_1 + f_5)</td>
<td>39.60</td>
<td>47.76†</td>
</tr>
<tr>
<td>DA + (f_1 + f_6)</td>
<td>39.62</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_1 + f_7)</td>
<td>39.60</td>
<td>47.78</td>
</tr>
<tr>
<td>DA + (f_1 + f_3 + f_4 + f_5)</td>
<td>39.71†</td>
<td>47.77†</td>
</tr>
<tr>
<td>DA + (f_1 + f_3 + f_4 + f_5 + f_6 + f_7)</td>
<td>39.53</td>
<td>47.68†</td>
</tr>
</tbody>
</table>

Table 3: BLEU and TER scores of systems on English to Portuguese data set showing impact of sparse topic features against a weak (BA) and a strong (DA) baseline system (in bold). System performance marked with * and † show significant different results w.r.t baseline (BA) and domain adapted system (DA) with p-value < 0.05.

The second block shows results with the topic adapted system using the Joint Probability feature. Individually, the best features which give significant improvements in TER scores over the DA system are the crude-count, cosine similarity and the Hellinger’s divergence feature, i.e. DA+f1+f3, DA+f1+f4, DA+f1+f6. Concerning the BLEU scores, only the cosine similarity feature is significantly better than the domain adapted but on an average we observe 0.2 BLEU points improvement per feature. We also observe significant gains in TER over the domain adapted system when we combine all dense features together (see systems DA+f1+f2+f3+f4+f5 and DA+f1+f2+f3+f4+f5+f6+f7).

The results in the third block are using the GeoMean sparse feature. The component wise analysis shows that the cosine similarity, Hellinger’s divergence and the Sensitivity features give statistically significant improvements \((p < 0.05)\) over the DA system in terms of TER scores. Average improvements of 0.2 BLEU points are observed across individual feature systems. In
terms of BLEU, our best system is the one which combines all the features together without Hellinger’s divergence achieved statistically significant gains ($p < 0.05$) over the DA system (DA+f2+f3+f4+f5). In terms of TER, we observe statistically significant gains ($p < 0.05$) of 0.34 TER points in the best systems. These systems are the combination of GeoMean feature with Sensitivity feature (DA+f2+f7) and the one which combines all dense features together with the GeoMean feature (DA+f2+f3+f4+f5+f6+f7).

In Table 4 we show examples from the evaluation set where our system solved the problems of context disambiguation and proper rendering of proper names. In the first example the source title contains the words **endurance** which is a brand and **colander** which is the name of a cooking tool. Domain adapted system (DA) incorrectly translates **endurance** as **resistência** while the topic adapted system (TA) correctly took the verbatim translation. In the same sentence, DA translates **colander** as **escorredor** while TA correctly picked the more specific translation **escorredor de macarrão** (colander for pasta). In the second example the source contains: **columbia river crkt** which is a brand name. The DA system erroneously translates **river** as **rio** while the TA system produced the correct verbatim translation of the brand name.

<table>
<thead>
<tr>
<th>Source</th>
<th>rsvp international 5-qt. endurance colander 1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>rsvp international 5-qt. resistência escorredor 1024</td>
</tr>
<tr>
<td>TA</td>
<td>rsvp international 5-qt. endurance escorredor de macarrão 1024</td>
</tr>
<tr>
<td>Ref1</td>
<td>rsvp international 5-qt. coador endurance 1024</td>
</tr>
<tr>
<td>Ref2</td>
<td>escorredor de macarrão endurance de 5 quartos de galão da rsvp international 1024</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>columbia river crkt crawford kasper lawks zytel knife !</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>rio columbia crkt crawford kasper lawks zytel faca !</td>
</tr>
<tr>
<td>TA</td>
<td>columbia river crkt crawford kasper lawks zytel faca !</td>
</tr>
<tr>
<td>Ref1</td>
<td>faca columbia river crkt crawford kasper lawks zytel !</td>
</tr>
<tr>
<td>Ref2</td>
<td>faca canivetp columbia river crkt crawford kasper lawks zytel !</td>
</tr>
</tbody>
</table>

Table 4: Examples from the domain adapted (DA) and topic adapted (TA) systems.

### 7 Conclusion

An open problem in machine translation is how to effectively handle and incorporate the context information in the translation models that can help the system to properly disambiguate between competing translation alternatives. In this paper we presented methods for topic adaptation for phrase-based machine translation that have been experimented in an e-commerce application scenario. In particular, starting from state-of-the-art LDA topic modeling, we present several feature functions (some of them new and others derived from the literature) that combine topic information integrated in the phrase-table with topic information inferred on-the-fly on the input text. We report results on an English-Portuguese translation task of item titles that show consistent and statistical significant improvements through topic adaptation over both a generic baseline MT system and a domain adapted MT system. Our work shows that the use of sparse features permits to identify and cope with topic-translation inconsistencies in the training data, which from one side cannot be avoided when data from multiple and diverse sources are pooled together, but from the other side calls for more refined methods to label such training data with topic labels. In the future we plan to investigate feature regularization methods in order to select topic-features with high discrimination power.

### Acknowledgements

The first author received support through a financial gift by eBay Inc. to FBK. The second author received support by the EU H2020 funded MMT project (grant agreement No 645487),
by eBay Inc., and by FBK’s Mobility programme.

References


Machine Translation with Source-Predicted Target Morphology

Joachim Daiber J.Daiber@uva.nl
Khalil Sima’an K.Simaan@uva.nl
Institute for Logic, Language and Computation, University of Amsterdam
Science Park 107, 1098 XG Amsterdam, The Netherlands

Abstract

We propose a novel pipeline for translation into morphologically rich languages which consists of two steps: initially, the source string is enriched with target morphological features and then fed into a translation model which takes care of reordering and lexical choice that matches the provided morphological features. As a proof of concept we first show improved translation performance for a phrase-based model translating source strings enriched with morphological features projected through the word alignments from target words to source words. Given this potential, we present a model for predicting target morphological features on the source string and its predicate-argument structure, and tackle two major technical challenges: (1) How to fit the morphological feature set to training data? and (2) How to integrate the morphology into the back-end phrase-based model such that it can also be trained on projected (rather than predicted) features for a more efficient pipeline? For the first challenge we present a latent variable model, and show that it learns a feature set with quality comparable to a manually selected set for German. And for the second challenge we present results showing that it is possible to bridge the gap between a model trained on a predicted and another model trained on a projected morphologically enriched parallel corpus. Finally we exhibit final translation results showing promising improvement over the baseline phrase-based system.

1 Introduction

Translation into a morphologically rich language poses a challenge for statistical machine translation systems. Rich morphology usually goes together with relatively freer word order of the target language, which makes it difficult to predict morphology and word order in tandem. Technically speaking this difficulty could be due to data sparsity, but possibly also due to morphological agreement between words over long distances. In this paper we explore the idea of combating sparsity by conducting translation in a probabilistic pipeline (chain rule), whereby morphological choice may precede lexical choice and reordering.

Whenever the predicate-argument structures of the source and target strings are similar, we expect that the linguistic information required for determining the morphological inflection of a plausible translation resides in the source sentence and its syntactic dependency structure. Consequently, we explore target morphology as a source-side prediction task which aims at enriching the source sentence with useful target morphological information. Practically (see Figure 1), after word aligning the sentence pairs, we project a subset of the target morphological attributes to the source side via the word alignments, and then train a model to predict these attributes on predicate-argument aspects of source dependency trees (i.e., without the source word order).
Our approach differs from other approaches to predict target morphology (e.g. Chahuneau et al. (2013)) mainly in that we predict on the source side only. A related intuition underlies source-side reordering schemes, which have seen a surge of successful applications recently (e.g. Collins et al. (2005) or Lerner and Petrov (2013)). While syntax-driven source-side reordering assumes that source and target syntax are similar, here we make a weaker assumption, namely that the predicate-argument structures are similar.

We explore the prediction of target morphology on the source side because we see several benefits that could potentially be exploited for further improving machine translation into morphologically rich languages. Source-side prediction models can capitalize on the much reduced complexity of having to represent and process only the input source sentence instead of a large lattice of target hypotheses. Hence, morphological agreement can be enforced over long distances by morphological predictions for the full source sentence. Furthermore, while not pursued in the present work, we hypothesize that the morphological information predicted by our model can be exploited in the word alignment process.

Our contributions in this paper are three. Firstly, we report experiments to support the hypothesis that projecting morphology to the source side could be beneficial for translation, and then present a model for learning to predict target morphology on the source side (Section 4). Secondly, we address how to automatically learn the set of morphological attributes that fit with the parallel training data (Section 5). Finally, we introduce methods for integrating this new information into a machine translation system and evaluate on a translation task (Section 6).

2 Related work

Various approaches have been proposed to the problem of translating between languages of varying morphological complexity. Avramidis and Koehn (2008) enrich the morphologically impoverished source side with syntactic information and translate via a factored machine translation model. In spirit, this paper is closely related to the present work; however, while their decorations are source-side syntactic information (e.g. the noun is the subject), we directly predict target morphology and learn to select the most relevant properties. A similar approach, in which source syntax is reduced to part-of-speech tags is used successfully for translation into Turkish (Yeniterzi and Oflazer, 2010). Following the tradition of two step machine translation (Bojar and Kos, 2010), Fraser et al. (2012) translate morphologically underspecified tokens and add inflections on the target side based on the predictions of discriminative classifiers.

Carpuat and Wu (2007), Jeong et al. (2010), Toutanova et al. (2008) and Chahuneau et al. (2013) propose discriminative lexicon models that are able to take into account the larger context of the source sentence when making lexical choices on the target side. These proposals differ mostly in the way that the additional morphological information is integrated into the machine translation process. Jeong et al. (2010) integrate their lexical selection model via features in the underlying treelet translation system (Quirk et al., 2005). Toutanova et al. (2008) survey two basic methods of integration. In the first method, the inflection prediction model is...
allowed to change the inflections produced by the underlying MT system. The second method is a two step method, where the MT system translates into target-language stems, which are then inflected by the inflection model. Chahuneau et al. (2013) create synthetic phrases, i.e. phrases with inflections that have not been observed directly in the training corpus but have been created by an inflection model. These synthetic phrases are then added to the training data of the MT system and marked as such. This enables the MT system to learn how much to trust them.

Finally, Williams and Koehn (2011) add unification-based constraints to the target side of a string-to-tree model. The constraints are extracted heuristically from a treebank and violations are then penalized during decoding.

### 3 Morphology projection hypothesis

A morphological attribute is a morphological property of a word. Each morphological attribute can assume any of a predetermined set of values, such as \{nom, acc, dat, gen\} for the morphological attribute case in the German language. Further, the morphological attributes are refined based on a set of 9 atomic parts of speech, yielding a set of morphological attributes of the form noun:case, adj:case, verb:tense, etc.

In this paper, we are interested in the question whether target morphology can be addressed directly on the source. We hypothesize that projecting target morphological attributes and learning to predict them on source side trees can be beneficial to machine translation. To test this hypothesis we initially perform translation experiments with a standard phrase-based MT setup with and without projected morphological information. These experiments provide an indication for the potential of such an approach. They do, however, not answer the question to what extent target morphology can realistically be predicted on the source side. This question will be addressed in the next sections. We perform translation experiments with translation systems decorated with projected morphological attributes. In these translation systems, the target side of the test set was processed with a morphological tagger and subsets of the resulting morphological attributes were projected to the source side via alignments. These experiments provide a conservative indication of the potential of this approach. They are not oracle translation experiments, but simulate an optimal target morphology prediction model. The three systems listed in Table 1 differ only in the subset of morphological attributes they use.

The experiment is documented in Table 1. We evaluate translation quality with METEOR and BLEU (Denkowski and Lavie, 2011; Papineni et al., 2002), word order with Kendall’s Tau (Kendall, 1938) and lexical choice with unigram BLEU. Statistical significance tests are performed for the translation scores (METEOR and BLEU) using the bootstrap resampling method (Koehn, 2004). The results show that projecting target morphological attributes improves trans-

<table>
<thead>
<tr>
<th>Training and test decor.</th>
<th>Tags</th>
<th>MTR</th>
<th>BLEU</th>
<th>Word order</th>
<th>Lexical choice</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (baseline)</td>
<td>-</td>
<td>35.74</td>
<td>15.12</td>
<td>45.26</td>
<td>49.86</td>
</tr>
<tr>
<td>Projected manual set</td>
<td>77</td>
<td>36.34</td>
<td>15.86</td>
<td>45.79</td>
<td>51.30</td>
</tr>
<tr>
<td>Projected automatic set</td>
<td>225</td>
<td>36.50</td>
<td>15.73</td>
<td>46.45</td>
<td>51.24</td>
</tr>
<tr>
<td>Projected full set</td>
<td>846</td>
<td>36.67</td>
<td>15.96</td>
<td>46.27</td>
<td>51.52</td>
</tr>
</tbody>
</table>

Table 1: Translation with various subsets of projected morphology.

---

1Details of the experimental setup are provided in Section 6.3.
Figure 2: Morphology projection and a source dependency chain.

4 Modeling target-side morphology

Since the word order of the source and target language may differ significantly, predicting morphology in a sequential, word-by-word fashion could be inadequate. We think that source syntax and the source predicate argument structure should be informative for predicting target morphology. Hence, we propose a source-side dependency chain model \( P(s_m' | \tau, s) \) to predict the morphologically enriched source string \( s_m' \) given a lexical dependency tree \( \tau \) of \( s \).

4.1 Source-side dependency chains

A source-side dependency chain is any path from the root of the source dependency tree to any of its leaf nodes, such as \( \text{escaped} \rightarrow \text{from} \rightarrow \text{police} \rightarrow \text{the} \) in Figure 2. Every node with a 1-to-1 alignment to a target node is decorated with the target node’s morphological attributes. A standard morphological tagger, such as the \( n \)-th order linear chain CRF model (e.g. Mueller et al., 2013), would predict the attribute–value vector for each word left-to-right with a history of \( n - 1 \) tags. Modeling with source-side dependency chains instead, gives various advantages: Besides providing access to the morphological tags assigned to the dependency tree parent and grandparent nodes, it implicitly encourages morphological agreement between a node and its \( n - 1 \) ancestor nodes. The model benefits from access to the node’s syntactic role, for example

\[2\] The difference is statistically significant at \( p < 0.05 \).
Table 2: Impact of attribute selection and model parameters on prediction quality ($F_1$ score).

<table>
<thead>
<tr>
<th></th>
<th>Manual</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>50k</td>
<td>68.50</td>
<td>70.13</td>
<td>68.86</td>
<td>70.84</td>
<td>69.73</td>
<td>70.97</td>
<td>58.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100k</td>
<td>67.08</td>
<td>67.38</td>
<td>67.01</td>
<td>69.33</td>
<td>71.15</td>
<td>69.52</td>
<td>58.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200k</td>
<td>67.40</td>
<td>67.40</td>
<td>68.55</td>
<td>69.58</td>
<td>71.15</td>
<td>70.06</td>
<td>57.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50k</td>
<td>72.67</td>
<td>70.36</td>
<td>72.86</td>
<td>74.67</td>
<td>71.42</td>
<td>71.83</td>
<td>62.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100k</td>
<td>70.01</td>
<td>71.89</td>
<td>69.82</td>
<td>72.63</td>
<td>72.04</td>
<td>72.61</td>
<td>62.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200k</td>
<td>69.40</td>
<td>69.46</td>
<td>69.99</td>
<td>71.44</td>
<td>70.80</td>
<td>69.83</td>
<td>60.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Best overall $F_1$ score highlighted in bold.

4.2 Model estimation
We estimate the source dependency chain model using the general CRF framework. In a linear-chain CRF model, the probability of a tag sequence $y$ given a sentence $x$ is:

$$P(y | x) = \frac{\exp \sum_i \lambda_i \cdot \phi_i(y, x, t)}{\sum_y \exp \sum_i \lambda_i \cdot \phi_i(y, x, t)}$$

where $t$ is the index of a token, $i$ is the index of a feature and $\lambda_i$ is the weight corresponding to the binary feature $\phi_i(y, x, t)$. To improve training and inference time, we use a coarse-to-fine pruned CRF tagger (Mueller et al., 2013). The training procedure is identical to the linear-chain case, except that we use dependency chains instead of left-to-right chains as training examples.

The dependency chain model’s feature set is based on the set used in the linear chain CRF for morphological tagging (Mueller et al., 2013). Additionally to the features used by Mueller et al. (2013), we add the following feature templates: the dependency label of the current token, the dependency label of the parent token, the number of children of the current token, the source-side part-of-speech tag of the token, and the current token’s child tokens if they are a determiner ($AuxA$), auxiliary verb ($AuxV$), subject ($Sb$) or a preposition ($AuxP$).

4.3 Intrinsic evaluation
To evaluate the quality of the source dependency chain predictions, we perform experiments on a heldout dataset. Models are trained on a subset of the parallel Europarl data. Evaluation is performed using the $F_1$ score of the predictions compared to the projected morphological attributes obtained by automatic alignment of the source and target side of the evaluation set.

Impact of model parameters Table 2 shows prediction performance of the dependency chain model in relation to a selection of model parameters. For each morphological attribute set, we train models of order 5, 6 and 7. All models are trained on sets of 50k, 100k and 200k dependency chains, which are randomly sampled from the training data. In strict training mode, we require that target words and source words connected by alignment links agree in their coarse part of speech tags. This restriction enforces a weak form of isomorphism between the source and the target sentence and hence limits the training set to training instances of potentially higher quality. In the relaxed setup, no such agreement is enforced.

Up to a certain point, higher order models perform better than models with shorter dependency histories; however, these models are also prone to the issues of data sparsity and overfit-
The results show that strict training performs worse than the relaxed training regime. The strict training regime could possibly produce cleaner training examples; however, since it also enforces a potentially unrealistic isomorphism between the two sentences, those examples may also be less helpful for the final prediction.

Impact of morphological attribute selection As shown in Section 3, it is possible to reduce the set of morphological attributes without major losses in translation quality. For the dependency chain model, smaller attribute sets are preferable since they lead to less complex models and faster training times. Individual attributes may be difficult to predict; hence, the exact selection of attributes is also important for prediction quality.

<table>
<thead>
<tr>
<th></th>
<th>Manual</th>
<th>Automatic</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training time, 50k</td>
<td>36m</td>
<td>45m</td>
<td>77m</td>
</tr>
<tr>
<td>Training time, 100k</td>
<td>58m</td>
<td>82m</td>
<td>2h51m</td>
</tr>
<tr>
<td>Training time, 200k</td>
<td>1h54m</td>
<td>3h5m</td>
<td>6h44m</td>
</tr>
<tr>
<td>Tags</td>
<td>77</td>
<td>225</td>
<td>846</td>
</tr>
<tr>
<td>Best F1</td>
<td>72.86</td>
<td>74.67</td>
<td>62.18</td>
</tr>
</tbody>
</table>

Table 3: Training times and best scores for the three attribute sets.

Table 3 summarizes training times and prediction performance of the three morphological attribute sets. Larger attribute sets and more training examples lead to longer training times. Overall, the automatic set produces more accurate results than the manual selection. Our analysis shows that this is largely due to difficult to predict verb attributes, which are included in the manual selection but are not part of the automatically learnt set. The finding that these attributes are hard to predict is in line with Fraser et al. (2012), who equally dropped the prediction of verb attributes in later work.

5 Learning salient morphological attributes

Decorating the source language with all morphological properties of the target language will lead to data sparsity and will complicate the prediction task. Therefore, it is necessary to reduce this set to only morphological attributes which are helpful for a given language pair. We consider a morphological attribute to be salient if it enables the machine translation system to perform better lexical selection. It is computationally infeasible to test all possible combinations of morphological attributes in a full machine translation system; hence, we approximate the machine translation system’s ability to perform lexical selection with a word-based translation system given by IBM model 1 (Brown et al., 1993). Based on this simplified translation model, the set of salient features which improve the translation performance can be chosen by a clustering procedure.

5.1 Learning procedure

Let \((s, t)\) be a pair of parallel sentences in source and target language. IBM model 1 provides an iterative method for estimating the translation model \(P(t \mid s)\) from a set of parallel sentences. We add the morphological decoration \(s'_{m}\) to this model. The translation model now takes the following form:

\[
P(t \mid s) = \sum_{s'_{m} \in \Theta_{m}(s)} P(s'_{m} \mid s)P(t \mid s'_{m})
\]
where \( P(t \mid s'_m) \) is the standard IBM model 1 formulation applied to morphologically decorated source tokens. In this simple machine translation model, the morphological attributes are directly attached to the source words. For example, if the English token police is decorated with grammatical case, gender and number, it would be replaced by the string police\( /\)case=dat+gender=female+number=singular. We define the log-likelihood of a set of parallel sentences \( X \) to be:

\[
\mathcal{L}(X) \equiv \log \prod_{(s,t) \in X} P(t \mid s) P(s) = \sum_{(s,t) \in X} \log P(t \mid s) + \log P(s)
\]

Let \( M_0 \) be the initial set of all morphological attributes observed in the training corpus. Our goal is to find the set \( M_n \subseteq M_0 \) which maximizes the likelihood of a heldout dataset. By \( s'_m^{(i)} \) we denote the decorated source sentence containing only the morphological attributes in \( M_i \). We formulate the search for the set \( M_n \) as follows:

\[
M_n = \arg \max_{M_i \subseteq M_0} \sum_{(s,t) \in X} \log P(t \mid s) + \log P(s)
\]

\[
= \arg \max_{M_i \subseteq M_0} \sum_{(s,t) \in X} \log P(t \mid s)
\]

\[
= \arg \max_{M_i \subseteq M_0} \sum_{(s,t) \in X} \log \left( \sum_{s'_m \in \Theta_m(s)} P(s'_m \mid s) P(t \mid s'_m^{(i)}) \right)
\]

We found the estimates for \( P(s'_m \mid s) \) using the full set of attributes \( M_0 \) to be reasonable, with sufficient probability mass assigned to the most likely path. Therefore, we approximate this model by only using the first-best (Viterbi) assignment \( s'_m^{(i)} \). The final, simplified search objective is therefore:

\[
M_n = \arg \max_{M_i \subseteq M_0} \sum_{(s,t) \in X} \log \left( P(s''_m \mid s) P(t \mid s''_m^{(i)}) \right)
\]

\[
= \arg \max_{M_i \subseteq M_0} \sum_{(s,t) \in X} \log P(t \mid s''_m^{(i)})
\]

The optimal set of attributes can now be determined with a clustering procedure starting from the full set of morphological attributes \( M_0 \). This procedure is reminiscent of Petrov et al. (2006) since as in their work, we can simulate the removal of a morphological attribute by merging the statistics of each of its occurrences.\(^3\)

1. Initialization:
   - Estimate the source dependency chain model \( P(s'_m^{(0)} \mid s) \), apply it to decorate the training and heldout set, producing \( T_0 \) and \( H_0 \) (datasets \( T \) and \( H \) decorated with \( M_0 \)).
   - Estimate \( P(t \mid s''_m^{(0)}) \): perform 5 iterations of IBM Model 1 training on \( T_0 \).
2. Start with \( i = 0 \).
3. Calculate \( P(t \mid s''_m^{(i)}) \) for each sentence pair in the heldout set \( H_i \).

\(^3\)For example, to simulate the removal of the attribute gender, we would merge the statistics of every occurrence of the attribute (either gender=male or gender=female). The two tags case=nom+gender=female and case=nom+gender=male would therefore be merged into one tag case=nom.
Table 4: Salient attributes for English–German.

<table>
<thead>
<tr>
<th>Noun</th>
<th>Adjective</th>
<th>Verb</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>gender</td>
<td>number</td>
<td>number</td>
</tr>
<tr>
<td>number</td>
<td>number</td>
<td>case</td>
<td>case</td>
</tr>
<tr>
<td>synpos</td>
<td>synpos</td>
<td>degree</td>
<td>degree</td>
</tr>
<tr>
<td>gender</td>
<td>gender</td>
<td>number</td>
<td>number</td>
</tr>
<tr>
<td>number</td>
<td>number</td>
<td>case</td>
<td>case</td>
</tr>
<tr>
<td>synpos</td>
<td>synpos</td>
<td>degree</td>
<td>degree</td>
</tr>
</tbody>
</table>

† Transferred with lemma. ‡ Propagated from noun. * Dropped in later work.

4. Find the attribute $\hat{m} \in M_i$, such that:

$$\hat{m} = \arg \min_{m \in M_i} \left( \sum_{(s,t) \in H_i} \log P(t \mid s''_{m}^{i}) - \log P(t \mid s''_{m}^{i} \backslash m') \right)$$

where $s''_{m}^{i} \backslash m'$ denotes a sentence with the attributes in $M_i$ minus attribute $m'$.

5. Merge all values of $\hat{m}$ in $T_i$ and $H_i$, producing $T_{i+1}$ and $H_{i+1}$.

6. Estimate $P(t \mid s''_{m}^{i+1})$: Merge the t-tables containing $\hat{m}$ and perform IBM Model 1 iteration on $T_{i+1}$.

7. Repeat from (3) with $i = i + 1$. Stop if no possible merge improves $L(H_i)$.

5.2 Intrinsic evaluation

The complexity of the clustering procedure is $O(|M| \times k \times l^2)$ for $k$ sentences of length $l$. In practice, the procedure runs for several hours on a standard machine. Table 4 shows the attributes determined by the learning procedure. The column Auto shows the procedure’s selection and the column Manual shows the manually determined set of morphological attributes for the same language pair, as used by Fraser et al. (2012).

Quality of the selection From inspection of these attributes, we find that our method learns a reasonable set of salient attributes. The manual and automatic selections differ mainly in the verb attributes, which our learning procedure removed from the final set. Morphological attributes in the manual selection which are marked with ‡, are attributes that in the work of Fraser et al. (2012) were transferred as part of the translated stem by their MT system. The symbol † marks morphological attributes that they propagated from the noun (for example, an adjective’s case is copied from the noun it modifies). Finally, the verb attributes, which are marked with * are used by Fraser et al. (2012) but found to be problematic by Cap et al. (2014b) and dropped in later work (Cap et al., 2014a). Likewise, inspection of our model showed that verb attributes perform badly as they may be difficult to predict. Hence, our procedure successfully learnt not to model these attributes while retaining the beneficial noun and adjective attributes.

Granularity of the morphological attributes When simulating the removal of a morphological attribute with this learning algorithm, all of its values are merged. In some language pairs, however, it would be useful to merge the individual values of the attributes instead. For example, from the spelling of German nouns it is usually not recognizable whether the noun is case=nominative or case=accusative. Hence, the algorithm should ideally be able to also merge individual values. Since this is a straight-forward extension of our current algorithm, we plan to evaluate this aspect in future work.
6 Morphology-informed translation

To leverage the morphology predictions in a machine translation decoder, we integrate this additional information into the translation model. During training and tuning, the translation model is decorated with morphological attributes either projected from the target side or predicted by our dependency chain model.

6.1 Integration of target morphology predictions

In practice, the predicted morphological attributes on the source side can be integrated into the machine translation system as arbitrary features based on source morphology and target strings. In our experiments, we opted for a feature representation in which this information is encoded as source morphology-to-target affix features. We chose this simple representation because it is generic enough to produce improvements on the one hand and it is not prone to overfitting on the other hand. For each phrase candidate on the source side, sparse features fire for a given sequence of source-side morphology tags and target-side string affixes. As an example, consider the sentence Peter entkam der Polizei (Peter escaped from the police) from Figure 2. In this case, the morphological attributes gender (female), number (singular) and grammatical case (dative) would have been projected from the target to the source side for the phrase the police/der Polizei. When translating the source segment the police, the feature gender=fem+number=sing+case=dat X → -er X would fire based on the predicted morphology. This hint would help the machine translation system choose the correct German determiner der.\(^4\)

6.2 Inference strategies

At test time, the morphological decoration of the source sentence needs to be selected. This decision should ideally take into account both the predictions of our source-side dependency chain model and the content of the phrase table, which may be decorated with projected morphology.

We compare several inference strategies. The major distinction between these strategies is whether the machine translation system is trained and tuned on projected morphology or predicted morphology. Training on predicted morphology has the benefit that it lets the MT system learn how much it can trust the predictions made by the dependency chain model. However, this method is also more laborious in system development, since it requires retraining and tuning the whole translation system for every change in the prediction model.

Training and decoding with Viterbi predictions In the first decoding setup, which is similar to the most common setup used in preordering, we decorate both training and test set with the Viterbi decorations extracted from the dependency chain model. Specifically, for each possible dependency chain in the source dependency tree, we perform standard CRF Viterbi tagging starting from the root of the tree. The full training and tuning set is decorated with these single-best predicted decorations. System training and tuning is then performed on these sets. During test time, only the single-best Viterbi prediction is considered by the MT system.

Training on projected morphology and decoding with Viterbi predictions The projected training setup differs from the previous setup in that the morphological decorations on the training and tuning set are not predicted but projected from the target side via alignments. During test time, the decorations are predicted using single-best Viterbi predictions as in the previous setup. While this strategy is advantageous since it simplifies the system training, the main downside of this strategy is that it cannot take into account possible shortcomings of the prediction model.

\(^4\)This feature example is taken from the weights of the system trained with the automatic morphological attribute set and predicted training and test decoration.
Table 5: Translation with predicted test decorations.

At training time, only projected decorations are observed, which might not be realistic when taking into account the prediction model.

6.3 Evaluation

Having introduced and evaluated the attribute selection process and the prediction of target-side morphological attributes based on source-side dependency chains, we now turn to the evaluation of the predicted morphological information within a full machine translation pipeline.

Experimental details We use a standard phrase-based machine translation system (Cer et al., 2010) with a 5-gram language model and distortion-based reordering (dl=5). Features based on the source morphology predictions are learnt on either the projected morphology or the predictions of the source dependency chain model. Experiments are conducted on English–German. Source-side dependency trees are predicted based on the HamleDT treebank (Zeman et al., 2012) using TurboParser (Martins et al., 2010). The dependency parser is trained to produce pseudo-projective dependency trees (Nivre and Nilsson, 2005).

The system is trained on the full parallel sections of Europarl (Koehn, 2005) and tuned and tested on the WMT 2009 and WMT 2010 newstest sets respectively.

Monolingual morphological tagging is performed using the Marmot CRF-based tagger (Mueller et al., 2013). The tagger is trained on the English and German parts of the HamleDT treebank. The morphological attributes of both languages follow the Interset standard (Zeman, 2008), which contains 45 unique attribute vectors (tags) for English and 958 for German.

Discussion Table 5 shows the outcomes of using the inference strategies presented in Section 6.2. We evaluate translation quality with METEOR and BLEU (Denkowski and Lavie, 2011; Papineni et al., 2002), word order with Kendall’s Tau (Kendall, 1938) and lexical choice with unigram BLEU. Statistical significance tests are performed for the translation scores (METEOR and BLEU) using the bootstrap resampling method (Koehn, 2004).

The results show that both attribute selections show improvements over the baseline when training and testing on predicted morphology. On the other hand, when training on projected morphology and performing Viterbi predictions, a visible gap between the manual set and the automatic set can be observed. This gap indicates that with the automatic set, the predictions by the dependency chain model are closer to the projected predictions so that the machine trans-
lation system learns realistic weights for the prediction part. Additionally, the system based on the automatic selection produces a significantly better METEOR score than the system using the manual selection. As in the experiments with projected morphology, the results of this evaluation indicate that the improvements stem from both word order choices as well as better lexical selection. In terms of time performance, we found that the additional information does not significantly affect the speed of the translation system. The Viterbi algorithm for predicting the target morphology is efficient and as the information is passed to the MT system as sparse features, no additional complexity is added. While we have focused on the language pair English–German, the methods presented in this paper are applicable to many other language pairs. We therefore aim to perform additional experiments for morphologically-rich target languages such as Turkish, Arabic and Czech.

7 Conclusion

In this paper, we have explored the novel approach of target morphology projection. After testing the idea empirically, we have put forward three proposals to realize this idea: First, we introduced the dependency chain model for predicting arbitrary target morphology attributes based on source dependency trees. Second, we introduced a learning procedure to determine a language pair’s set of salient morphological attributes. And finally, we have introduced and compared various strategies for integrating this new information into a machine translation system. The experiments we have performed have provided us with important insights. They have demonstrated that projecting a small subset of morphological attributes to the source side can provide major translation improvements, while reducing the complexity of prediction. Furthermore, the approach for learning the useful subset performs well based both on the intrinsic evaluation and the empirical results during prediction and translation. Given that previous work has found it rather difficult to achieve improvements in German morphology, we consider the improvements in METEOR score and the modest improvements in BLEU score encouraging.

While the prediction performance of the dependency chain model leaves room for improvement, we submit that our experiments sufficiently demonstrate the potential of this approach. We plan to further improve the prediction performance of the dependency chain model with extensions such as the use of (bilingual) word embeddings that could help resolve ambiguous cases. In addition, to let the machine translation system better exploit this new knowledge, deeper integration (e.g. into the language model) is necessary. Both ideas constitute the main topics for extending the current work in the future.

Acknowledgements

We thank the three anonymous reviewers for their constructive comments and suggestions. The first author is supported by the EXPERT (EXPloting Empirical appRoaches to Translation) Initial Training Network (ITN) of the European Union’s Seventh Framework Programme.
References


Improved Beam Search with Constrained Softmax for NMT

Xiaoguang Hu  
huxiaoguang@baidu.com  
Wei Li  
liwei08@baidu.com  
Xiang Lan  
lanxiang@baidu.com  
Hua Wu  
wu_hua@baidu.com  
Haifeng Wang  
wanghaifeng@baidu.com  
Baidu Inc, Beijing, China

Abstract

We propose an improved beam search decoding algorithm with constrained softmax operations for neural machine translation (NMT). NMT is a newly emerging approach to predict the best translation by building a neural network instead of a log-linear model. It has achieved comparable translation quality to the existing phrase-based statistical machine translation systems. However, how to perform efficient decoding for NMT is still challenging, especially for commercial systems which provide real-time translation service. Unlike the standard beam search algorithm, we use a priority queue to choose the best hypothesis for the next search, which drastically reduces search space. Another time consuming factor is the softmax operation in the output layer because of the large target vocabulary size. To solve this problem, we introduce a limited word set of translation candidates to greatly reduce the computation complexity. Our experiments show that, under the GPU environment, our method achieves a speed about 3.5 times faster than the well optimized baseline system without sacrificing the translation quality. Our method translates about 117 words per second, beating the real-time translation requirements for practical MT systems.

1 Introduction

Neural network models have become increasingly popular in recent machine translation tasks. Initially, the neural networks were designed for language models, such as neural network language model (Bengio et al., 2003) and recurrent neural network language model (Mikolov et al., 2010). Devlin et al. (2014) proposed a neural network joint model to improve translation by considering the source contexts jointly. The above work integrated the neural network models into the traditional translation decoders as additional features in a log-linear framework (Och and Ney, 2002). Rather than applying a neural network as a part of the existing system, Cho et al. (2014) proposed a whole new RNN Encoder-Decoder approach which generates the target translation with a neural network directly. Bahdanau et al. (2014) extended the above approach by allowing an RNN model to automatically (soft-)search for parts of a source sentence to predict a target word, and achieved a translation performance comparable to the existing state-of-the-art phrase-based systems (Koehn et al., 2003).

Although NMT shows high potential, its decoding efficiency is still challenging. Cho et al. (2014) implemented a standard beam search decoding algorithm (Koehn, 2004) for an RNN...
Encoder-Decoder system published as GroundHog\textsuperscript{1}. The beam search algorithm successfully reduces the search space from exponential size to polynomial size, and it is able to translate about ten words per second. Even though the speed is acceptable for most research tasks, it is not yet efficient enough to meet the requirement of commercial systems for providing real-time translation service. Huang and Chiang (2007) proposed the forest rescoring algorithm which succeed to accelerate the decoding by using cube pruning. The main improvement comes from the reduction of language model calculation which is the most time-consuming part in both phrase-based and syntax-based system. However, there is no separable language model feature in NMT, which makes it difficult to use the cube pruning algorithm in NMT. Furthermore, we found that one of the most time-consuming part in NMT decoding is the softmax operation over the output layer because of the large vocabulary size. The hierarchical softmax (Mikolov et al., 2011) and Noise-contrastive estimation (Gutmann and Hyvarinen, 2010) succeed to accelerate softmax operation effectively both for training and predicting in the RNN language model. However, they don’t work well for the decoding application. In the language model task, we only need to compute the probability of the certain next word which we have known in advance. But in the decoding task, we have to compute all probabilities of candidates in the target vocabulary to know which one is the best candidate.

In this work, in order to make the NMT system meet the real-time translation requirements, we propose an improved beam search decoding algorithm with constrained softmax operation over the output layer for the RNN Encoder-Decoder machine translation. We found that many hypotheses can be pruned safely without affecting the final translation result, so that we apply a priority queue to choose the best hypothesis to be extended. Furthermore, we build a limited word set of translation candidates which is used to effectively reduce the computation complexity in softmax operation. The experimental results show that, our optimized algorithm achieves a speed about 3.5 times faster than well optimized baseline system at the same level of translation quality. The improved beam search with priority queue helps to reduce about 37.6\% hypothesis extension operations which results in 1.7 times speedup. The constrained softmax strategy succeeds to reduce the output layer size from 30,000 to about 300 which leads to another 2.1 times speedup.

In section 2, we introduce the standard beam search algorithm of RNN Encoder-Decoder approach in detail. In section 3, we present our improved beam search algorithm with constrained softmax operation. We describe our experimental settings in section 4 and report the experimental results in section 5. Finally, we summarize our conclusions and future work in section 6.

2 Neural Machine Translation

The end-to-end neural machine translation has been newly proposed in recent years. We first introduce the general RNN Encoder-Decoder translation framework (Bahdanau et al., 2014), and then we elaborate the standard beam search decoding algorithm for NMT in detail.

2.1 Preliminary Definitions

In the neural machine translation framework, a source sentence is represented as a sequence of 1-of-K coded word vectors

\[
x = (x_1, \ldots, x_T), x_i \in \mathbb{R}^{K_x}
\]

and a target sentence is also represented as a sequence of 1-of-K coded word vectors

\[
y = (y_1, \ldots, y_T), y_i \in \mathbb{R}^{K_y}
\]

\textsuperscript{1}https://github.com/lisa-groundhog/GroundHog
where $K_x$ and $K_y$ are the vocabulary sizes of source and target languages respectively. $T_x$ and $T_y$ are the lengths of source and target sentences respectively.

Given an input sentence $x$ and $y$, we define some important concepts and symbols in advance.

- **encoder**: RNN to convert the input sentence $x$ into a *sequence of hidden states* $h$.

- **decoder**: RNN to predict the probability distribution over $y_i \in \mathbb{R}^{K_y}$ from the given *context vector* and *hidden states*.

- **hidden state**: a vector from RNN hidden layer at a certain step $t$. Where $t$ is defined as the number of accepted or generated words right now. We use $h_t$ to represent the hidden state for the encoder, and $s_t$ for the decoder.

- **context vector**: a vector $c_t$ used to predict translation probability at step $t$ generated from $h$ and $s_t$.

- **hypothesis**: a data structure used to store search states includes $s_{t-1}, c_t, y_t$, and the translation probability at step $t$.

- **extend**: an operation to construct several new hypotheses from a hypothesis with translation candidates generated from the decoder.

### 2.2 Encoder-Decoder Framework

![RNN Encoder-Decoder Framework](image)

*Figure 1: RNN Encoder-Decoder Framework*
An RNN Encoder-Decoder translation system often contains an encoder and a decoder. Figure 1 shows how this system works to translate a Chinese sentence into an English sentence.

Given a Chinese sentence \( x = "beijing de chuzuche siji hen reqing" \) which means "the taxi driver in beijing is enthusiastic", the encoder converts \( x \) into a series of hidden states \( h = (h_1, \ldots, h_6) \) sequentially via equation (1).

\[
h_j = \begin{cases} 
  f(x_j, h_{j-1}) & j > 0 \\
  0 & j = 0
\end{cases}
\]

where \( h_j \in \mathbb{R}^n \) is a hidden state of the encoder after reading input word \( x_j \), \( n \) is the hidden layer size of the encoder, and \( f \) is a nonlinear function which is sometimes complicated such as in the long short term memory model (Hochreiter and Schmidhuber, 1997). Furthermore, a bidirectional RNN (Schuster and Paliwal, 1997), which consists of forward and backward RNNs, is used here.

Given the encoder hidden states \( h \) generated from \( x \), and all the previously predicted history words \( (y_1, \ldots, y_{i-1}) \), the decoder predicts the probability distribution over \( y_i \in \mathbb{R}^K \). For a translation candidate \( y \), the translation probability is computed in equation (2).

\[
p(y) = \prod_{i=1}^{T} p(y_i | y_1, \ldots, y_{i-1}, h)
\]

The translation history \( (y_1, \ldots, y_{i-1}) \) is represented by hidden state \( s_{i-1} \) and \( y_{i-1} \). The encoder hidden states \( h \) is represented as \( c_i \). The output probability is computed in equation (3).

\[
p(y_i | y_1, \ldots, y_{i-1}, h) = g(s_{i-1}, y_{i-1}, c_i)
\]

where \( g \) is a non-linear function, of which we will give an example later in section 3.2.

### 2.3 Standard Beam Search

We describe the beam search algorithm implemented in GroundHog as in Algorithm 1. Given the encoder, decoder and input sentence \( x \), we try to find the best translation \( \hat{y} = \arg\max_{y} p(y|x) \). A group of stacks are used to store hypotheses during searching. Beam size \( N \) is used to control the search space by extending only the top-\( N \) hypotheses in the current stack. With the above settings, the translation \( y \) is generated word by word from left to right.

We define *complete hypothesis* as hypothesis which outputs \( EOS \), where \( EOS \) is a special target word indicating the end of sentence.

As compared to the standard decoding algorithm (Koehn, 2004) in phrase-based machine translation, there are some differences. Firstly, a hypothesis is always of less score than the hypothesis from which it extended. Because after every extension in NMT, the total score will always be multiplied with a probability which is less than 1. But in phrase-based system, the score of a extension is not always less than 1 because the features such as word penalty score and phrase penalty score may be greater than 1 in the log-linear framework. Secondly, the stop conditions are different. The phrase-based system will stop searching when all the words in the source sentence are translated. In NMT, there is no exactly word alignment information to tell which word is translated or not. It will stop searching when it has generated \( N \) complete hypotheses.

### 3 Improved Beam Search

In this section, we introduce our improvement over the standard beam search algorithm for NMT. We observed that not all the hypothesis extensions are necessary, and some extensions
Algorithm 1 Standard Beam Search

1: function STANDARDSEARCH(enc, dec, x, y)
2:    define stacks s
3:    define set c
4:    create initial hypo and put it into s[0]
5:    \(i \leftarrow 0\)
6:    \(N \leftarrow \text{beam size}\)
7:    while \(s[i] \neq \emptyset\) do
8:        for all \(h \in s[i]\) do
9:            extend new hypos from \(h\)
10:           put new hypos into \(s[i + 1]\)
11:        end for
12:        prune \(s[i + 1]\) to keep \(N - \text{c.size}\) hypos
13:        move complete hypo in \(s[i + 1]\) to \(c\)
14:        \(i \leftarrow i + 1\)
15:    end while
16:    \(y \leftarrow \text{trace back from best } h \in c\)
17: end function

of inferior hypotheses could be skipped safely without affecting the translation results. Furthermore, we found that the operation to compute the softmax score over the output layer is very slow because the target vocabulary size \(K_y\) is often large. We can accelerate beam search by reducing the vocabulary size in advance with translation knowledge learned from the phrase-based model.

3.1 Hypotheses Extension with Priority Queue

The standard beam search algorithm extends the hypotheses from left to right stack by stack as shown in Figure 2 part (a). At certain step \(t\), all the hypotheses in the stack before \(t\) have to be extended. Actually, not all the extended hypotheses will succeed to generate a complete hypothesis with \(EOS\), because plenty of extended hypotheses will be pruned in the future.

Figure 2: (a) standard beam search (b) improved beam search: ■ extended hypothesis, △ best hypothesis to be extended, ○ unextended hypothesis
In our improved beam search algorithm, we use a priority queue $q$ to record the best hypotheses of each stack as shown in Figure 2 part (b). In a stack, there are at most $N$ hypotheses, which are ranked in descending order by their translation probabilities. If there is any unextended hypothesis in a stack, the one with the highest translation probability will be inserted into queue $q$. With above settings, the improved beam search works as shown in Algorithm 2.

- Firstly, we define the stacks $s$ and priority queue $q$ to manage hypotheses, and we create the initial hypothesis and insert it into $s[0]$ and $q$.
- Secondly, we select the best hypothesis in the queue $q$ which is the initial hypothesis by now, and extend at most $N$ new hypotheses from the initial hypothesis. We insert new hypotheses into $s[1]$ in descending order, then insert the best hypothesis in $s[1]$ into queue $q$. We update the queue $q$ by eliminating the initial hypothesis.
- After that, we select the best hypothesis $h$ in the queue $q$ which is of stack index $i$, and extend at most $N$ new hypotheses from $h$. We insert new hypotheses into stack $s[i + 1]$ in descending order, and prune the inferior hypotheses to make sure the number of hypotheses is less than $N$. For stack $s[i]$ and $s[i + 1]$, we update the hypotheses in queue $q$ to make sure that $q$ contains only the best unextended hypothesis of each stack. We adjust the queue $q$ to be in the priority order.
- We repeat the process until the best hypothesis $h$ to be extended in the queue is a complete hypothesis. It’s safe to stop searching and output the best translation $\hat{y}$ by tracing back from $h$, because $h$ is associated with the highest translation probability among any other unextended hypotheses by now, and no hypothesis in the searching space with beam size $N$ will get higher translation probability than $h$ in the future since $p(y_i|y_1, \ldots, y_{i-1}, h) \leq 1$.

Instead of extending the hypothesis stack by stack sequentially, our improved beam search algorithm only extend the best unextended hypothesis in all stacks. The standard algorithm stops searching after generating $N$ complete hypotheses, while our algorithm stops when a complete hypothesis $h$ is selected to be extended. Because $h$ is better than any other hypotheses in $q$, and the hypothesis in $q$ is better than any other unextended hypotheses in the same stack, and the unextended hypotheses in the stack is better than any other hypotheses extended from it in the future. Therefore our algorithm extends less hypotheses without changing translation result. All the unextended hypotheses inferior to $h$ will not be extended. Considering other breadth-first searching algorithms with a queue such as the classical Dijkstra’s algorithm, the size of the queue grows exponentially. In our algorithm, there is at most one hypothesis in the priority queue for each stack, and it costs less memory and more efficient to manage the queue.

Furthermore, we find that it’s unfair to compare the translation probabilities of hypotheses in the priority queue directly. Because the hypotheses with longer translation often tend to get lower probability. We design a parameter $\alpha$ to alleviate the influence of translation length, which is similar to the weight of word penalty feature in the phrase based system.

$$p(y|x) = p(y|x \overset{\text{geometry average}}{\sim} y_{i-1})$$  \hspace{1cm} (4)

Where $i \geq 1$ is the stack index of current hypothesis. If $\alpha = 0$, then $\tilde{p} = p$; if $\alpha = 1$ the $\tilde{p}$ is the geometry average of $p$ over $i$. The lower the $\alpha$ is, the less hypothesis will be pruned. After normalization, the translation result will be changed.

### 3.2 Constrained Softmax Over Output Layer

For a hypothesis extension, the decoder predicts the next word probability $p(y_i|s_{i-1}, y_{i-1}, c_i)$ by executing RNN feed forward operation. This procedure comprises some matrix multiplication operations and a final softmax operation over the output layer as shown in Figure 3. Here,
Algorithm 2 Improved Beam Search

1: function IMPROVEDSEARCH(enc, dec, x, y)
2:   define stacks s
3:   define priority queue q
4:   create initial hypo and put it into s[0], q
5:   \( N \leftarrow \) beam size
6:   \( h \leftarrow \) find the best hypo in q
7:   while \( h \) is not complete hypo do
8:      \( i \leftarrow \) get stack index of \( h \)
9:      extend new hypos from \( h \)
10:      put new hypos into \( s[i+1] \)
11:     prune \( s[i+1] \) to keep \( N \) hypos
12:     UPDATEQUEUE(\( q, s[i] \))
13:     UPDATEQUEUE(\( q, s[i+1] \))
14:   \( h \leftarrow \) find the best hypo in \( q \)
15: end while
16: \( y \leftarrow \) trace back from \( h \)
17: end function

18: function UPDATEQUEUE(q,s)
19:   \( h \leftarrow q \cap s \) \( \triangleright \) hypo both in \( q \) and \( s \)
20:   if \( \exists \) best unextended hypo \( uh \in s \) then
21:      replace \( h \) with \( uh \) in \( q \)
22:   else \( \triangleright \) no unextended hypo in \( s \)
23:      delete \( h \) from \( q \)
24:   end if
25:   adjust \( q \) to be a priority queue
26: end function

we compute word probability using the same \( g \) function proposed by Bahdanau et al. (2014) as follows.

\[
p(y_i|s_{i-1}, y_{i-1}, c_i) \propto \exp (y_i^T W_o u_i)
\]

\[
u_i = [\max (\tilde{u}_{i,2j-1}, \tilde{u}_{i,2j})]_{j=1,...,l}
\]

\[
\tilde{u}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i
\]

Where \( W_o \in \mathbb{R}^{K_y \times l} \), \( U_o \in \mathbb{R}^{2l \times n} \), \( V_o \in \mathbb{R}^{2l \times m} \) and \( C_o \in \mathbb{R}^{2l \times 2n} \) are weight matrices. And \( l \) is the size of maxout hidden layer \( t_i \); \( n \) is the size of hidden layer \( s_i \); \( m \) is the word embedding dimensionality of \( E y_{i-1} \).

Actually, we find that the matrix multiplication to compute the softmax over the output layer is often the most time-consuming operation. For most translation tasks, the size of output layer (\( K_y \geq 10,000 \)) is usually over ten times larger than sizes of other layers \((l, n, m \leq 1,000)\). The computation time complexity is close to \( O(l \times K_y) \). If we can reduce \( K_y \), the hypothesis extension will become faster. RNN with output layer factorized by class layer (Mikolov et al., 2011) and Noise-contrastive estimation (Gutmann and Hyvärinen, 2010) is able to accelerate softmax operation effectively both for training and predicting in RNNLM application. However, unlike estimating the translation probability for only one certain target word \( y_i \), we have to estimate the probability distribution for all target words to get the best candidate.

Given an input source sentence \( x \), we note that the number of possible translation candidates \( T_x \) is much smaller compared to the target vocabulary size \( K_y \). Instead of calculating the
translation probability for every word in the target vocabulary, we only calculate the probability for a limited set of target candidates which are relevant to the given x. The problem is how to get such a candidate set. Intuitively, we can make use of the word alignment information generated with IBM model (Brown et al., 1993). And we found that there are often too many candidates especially for the common words, which makes the improvement limited.

Actually, we can get more accurate translation candidates from the phrase pairs of the phrase-based translation model (Koehn et al., 2003). Firstly, we train a phrase-based translation model with the word aligned bilingual sentence pairs. Then we segment the input sentence x into a number of continuous source phrases. Next, we search the corresponding target phrases for all source phrases from the phrase-based translation model, namely phrase table. After that, we build a candidate word set by enumerating all the words in the target phrases. Finally, we constrain the softmax operation with the candidate word set for all the hypothesis extension of x.

One disadvantage of above approach is that it is too costly to load the whole phrase table into memory. Fortunately, we can reduce the memory cost by pruning phrase table carefully. The phrase table filtering techniques have been widely used in machine translation. Instead of considering both the phrase coverage and translation features in the filtering technique, our system only considers the word coverage, which is much easier to implement. We investigated the histogram pruning (Zens et al., 2012) and length-based pruning method to reduce the memory used in our system. For histogram pruning, we preserves the X target phrases with highest probability for each source phrase. For length-based pruning, we prune the phrase pairs which contain more than X words in source or target side. Where X is the threshold. We found that even the basic length-based pruning method works well enough in our task.

4 Experimental Settings

In this section, we evaluate the improved beam search algorithm on the task of Chinese-to-English translation.

4.1 Dataset

In this paper, we use Chinese-to-English bilingual corpora from LDC which is a subset of corpus for NIST08 task, containing 67M Chinese words and 74M English words, to train our models. No monolingual corpus are used to help the model training. The NIST MT03 Chinese-to-English test set is used as the development set, and NIST MT08 Chinese-to-English test set is used to evaluate our translation results. The development set is used to choose the best RNN
model in history, because the performance of RNN model will fluctuate during training.

4.2 Toolkits
The open source SMT system Moses (Koehn et al., 2007) is used to train a phrase-based machine translation system. The phrase table trained with Moses will be used to constrain the softmax translation. We use the open source RNN Encoder-Decoder toolkits GroundHog which is implemented with Theano (Bergstra et al., 2010; Bastien et al., 2012) to train a neural machine translation model. It is re-implemented with C++ in the GPU Environment and named NetTrans, which is well optimized and faster than GroundHog which is implemented in Python. Given the same model, there is no translation differences between GroundHog and NetTrans. We will report our experimental results on NetTrans instead of GroundHog.

4.3 Settings
We train a phrase-based model from the bilingual corpus with Moses using “grow-diag-final” word alignments. We train two RNN encoder-decoder models (set $K_x=30,000$ and $K_y=60,000$ separately) with GroundHog toolkit written with Theano by using the default settings proposed by Bahdanau et al. (2014). A multilayer network with a single maxout (Goodfellow et al., 2013) hidden layer is used to compute the conditional probability of each target word. The size of a hidden layer is 1000, the word embedding dimensionality $m$ is 620 and the size of maxout hidden layer in the deep output $l$ is 500. We use a minibatch stochastic gradient descent (SGD) algorithm together with Adadelta (Zeiler, 2012) to train all the RNN models. Each SGD update direction is computed using a mini-batch of 50 sentences. The training time for each model is approximately 22 days. We set beam size $N = 10$ for beam search decoding. We run both the training and decoding programs on a machine with GPU cards (NVIDIA Tesla K10).

5 Results and Analysis
5.1 Baseline
In this section, we report the translation results of Moses, GroundHog and NetTrans. Our experimental results in table 1 shows that the NetTrans runs about 3 times faster than GroundHog with the same translation results. GroundHog is implemented with Theano, which is a very flexible toolkit for general deep learning tasks. NetTrans is well designed especially for the RNN Encoder-Decoder framework on the GPU architecture with pure C++. In NetTrans, we execute the operations between the same matrices continuously to avoid additional GPU initialization and we use memory pool to avoid allocating memory frequently and store the matrix in memory continuously.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>Speed (words/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses</td>
<td>24.19</td>
<td>-</td>
</tr>
<tr>
<td>GroundHog</td>
<td>24.50</td>
<td>11.2</td>
</tr>
<tr>
<td>NetTrans</td>
<td>24.50</td>
<td>33.4</td>
</tr>
</tbody>
</table>

Table 1: Baseline Systems

5.2 Improved Beam Search with Constrained Softmax
We describe our experimental settings as follows. Baseline is the standard beam search algorithm implemented in NetTrans. Priority represents the improved beam search algorithm with

---

2http://www.statmt.org/moses/
priority queue, and the parameter $\alpha$ ($0 \leq \alpha \leq 1$) is set as in Equation (4). \textit{CandX} represents the histogram pruning with threshold $X$. Our experimental results under different \textit{Alpha} ($0 \leq \alpha \leq 1$) are shown in Figure 4. Curve \textit{Priority} tells the results of improved beam search with the priority queue. Curves \textit{Cand50} \textsuperscript{3}, \textit{Cand20} \textsuperscript{4} report the results of constrained softmax with different \textit{CandX}. As shown in Figure 4, both improved beam search and constrained softmax can accelerate the decoding speed effectively.

![Figure 4: Results of Improved Beam Search](image)

As shown in Table 2, our method achieves a speed more than 3.5 times faster than the baseline at the same level of translation quality with \( \alpha = 0.2 \). The improved beam search with priority queue achieves in 1.7 times speedup by reducing 37.6\% hypothesis extension operations. The hypothesis extension times both for our method and the baseline method (\textit{ext}_o and \textit{ext}_b) are counted respectively. The \textit{pruning ratio} is computed with equation \( (\text{ext}_b - \text{ext}_o)/\text{ext}_b \). The results show that the speed acceleration with different \( \alpha \) is roughly consistent with the \textit{pruning ratio}. The constrained softmax strategy leads to another 2.1 times speedup by reducing the output layer size from 30,000 to about 300. Theoretically, the speedup should be \( K_y \times n / \max\{n \times n, K_y \times n\} \) and thus 30 times faster with hidden layer dimension \( n = 1000 \). Actually, the standard softmax costs about half of the decoding time under the GPU environment. After constraining it with a candidate word set, it makes the constrained softmax no

\textsuperscript{3}We use the default settings in Moses that performs well for phrase-based MT with \textit{CandX}=50.

\textsuperscript{4}We use a reduced candidate set to examine how different candidate sizes affects translation quality.
longer the bottleneck of translation. At the same level of translation speed, our method performs more than 1 BLEU scores better than the method to reduce the beam size $N$ from 10 to 3 directly.

<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>Speed</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>24.50</td>
<td>33.4</td>
<td>-</td>
</tr>
<tr>
<td>$\alpha=0$</td>
<td>24.50</td>
<td>41.11</td>
<td>14.6%</td>
</tr>
<tr>
<td>$\alpha=0.2$</td>
<td>24.29</td>
<td>55.4</td>
<td>37.6%</td>
</tr>
<tr>
<td>$\alpha=0.2,Cand50$</td>
<td>24.29</td>
<td>111.6</td>
<td>42.8%</td>
</tr>
<tr>
<td>$\alpha=0.2,Cand20$</td>
<td>24.44</td>
<td>117.2</td>
<td>45.1%</td>
</tr>
<tr>
<td>Beam Size=3</td>
<td>23.37</td>
<td>104.4</td>
<td>66.6%</td>
</tr>
</tbody>
</table>

Table 2: Optimized Search with Pruning Ratio

5.3 Constrained Candidates Selection

We define $\text{PhraseX}$ as the length-based pruning with threshold $X$. $\text{Cands}$ is the average word size of translation candidates set for a given sentence. $\text{Coverage}$ represents the recall ratio of the real candidates generated by the baseline RNN system ($\alpha=0.2$). $\text{Memory}$ means the additional memory used to store the whole phrase table for candidates selection.

As shown in Table 3, the $\text{Coverage}$ decreases when reducing $\text{CandX}$, while $\text{Coverage}$ almost remains the same when reducing $\text{PhraseX}$. For example, when we only select those phrase pairs whose length is less than or equal to 2 ($\text{PhraseX}=2$), the coverage is almost the same as that when $\text{PhraseX}=7$. We find that our method works well with very limited memory (170MB) in setting $\text{Cand50}$ and $\text{Phrase2}$. This result is consistent with that for phrase-based SMT system, where phrase pairs with 2 or 3 words are frequently used and the target candidates of the source phrase is set to 50 by default. Moreover, with such setting, the translation quality is not degraded in BLEU score metrics. This result shows that our method for constrained candidate selection is effective to speed up the decoder without degrading of translation quality and with limited memory.

<table>
<thead>
<tr>
<th>Selection</th>
<th>Cands</th>
<th>Coverage</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cand5,Phrase7</td>
<td>83.8</td>
<td>87.3%</td>
<td>2.26GB</td>
</tr>
<tr>
<td>Cand10,Phrase7</td>
<td>126.6</td>
<td>91.4%</td>
<td>2.29GB</td>
</tr>
<tr>
<td>Cand20,Phrase7</td>
<td>195.4</td>
<td>94.0%</td>
<td>2.31GB</td>
</tr>
<tr>
<td><strong>Cand50,Phrase7</strong></td>
<td><strong>352.5</strong></td>
<td><strong>96.2%</strong></td>
<td><strong>2.33GB</strong></td>
</tr>
<tr>
<td>Cand50,Phrase3</td>
<td>350.9</td>
<td>96.2%</td>
<td>571MB</td>
</tr>
<tr>
<td>Cand50,Phrase2</td>
<td>337.6</td>
<td>96.1%</td>
<td>170MB</td>
</tr>
<tr>
<td>Cand50,Phrase1</td>
<td>276.3</td>
<td>95.3%</td>
<td>12MB</td>
</tr>
</tbody>
</table>

Table 3: Constrained Candidates Selection

5.4 The Effect of Vocabulary Size

We train a RNN Encoder-Decoder model with the target word size set to $K_y=60,000$, to examine how our method affects the decoding speed under different settings. As shown in Table 4, as the vocabulary size becomes larger, our approach performs even better, achieving a speed about 5.6 times faster than the baseline. The result is consistent with our intuition.
<table>
<thead>
<tr>
<th>System</th>
<th>BLEU</th>
<th>Speed (words/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>24.35</td>
<td>21.3</td>
</tr>
<tr>
<td>$\alpha=0$</td>
<td>24.35</td>
<td>27.1</td>
</tr>
<tr>
<td>$\alpha=0.2$</td>
<td>24.19</td>
<td>38.6</td>
</tr>
<tr>
<td>Cand50</td>
<td>24.05</td>
<td>113.8</td>
</tr>
<tr>
<td>Cand20</td>
<td>24.41</td>
<td>119.8</td>
</tr>
</tbody>
</table>

Table 4: Results of Vocabulary Size = 60,000

6 Conclusion and Future Work

In this paper, we propose an improved beam search decoding algorithm with constrained softmax operation for the neural network translation. It accelerates the translation speed more than 3.5 times compared to the standard beam search algorithm as in our well optimized baseline while keeping the same translation quality. Our experiments show that the improved beam search algorithm and the constrained softmax operation over the output layer are very effective for neural network translation.

In the future, we will try to accelerate the training for NMT using constrained softmax strategy. Furthermore, we will try to reduce the searching errors by selecting more accurate bilingual translation candidates.

Acknowledgments

This work is supported by the National Basic Research Program of China (973 program No. 2014CB340505).

References


Bilingual Distributed Phrase Representations for Statistical Machine Translation

Peyman Passban  ppassban@computing.dcu.ie
Chris Hokamp    chokamp@computing.dcu.ie
Qun Liu         qliu@computing.dcu.ie
ADAPT Centre, School of Computing, Dublin City University, Ireland

Abstract

Phrase–based machine translation (PBMT) relies upon the phrase-table as the main resource for bilingual knowledge at decoding time. A phrase table in its basic form includes aligned phrases along with four probabilities indicating aspects of the co-occurrence statistics for each phrase pair. In this paper we add a new semantic similarity score as a statistical feature to enrich the phrase table. The new feature is inferred from a bilingual corpus by a neural network (NN), and estimates the semantic similarity of each source and target phrase pair. We observe a significant increase in system performance with the addition of the new feature. We evaluated our model on the English–French (En–Fr) and English–Farsi (En–Fa) language pairs. Experimental results show improvements for all translation directions of En↔Fr and En↔Fa.

1 Introduction

Phrase–based machine translation begins by segmenting a source sentence into phrases and looking up candidate translations for the sub-sentential phrases in a phrase table. The goal of PBMT is to assemble these translation candidates into the optimal target sequence. Finding the best source segmentation, translation candidates, and phrase ordering is a search problem which is typically formulated as a log-linear model using both dynamic and static features, whose weights are optimized via a search heuristic on held-out development data. The inverse phrase translation probability $\varphi(f|e)$, inverse lexical weighting $\text{lex}(f|e)$, direct phrase translation probability $\varphi(e|f)$ and direct lexical weighting $\text{lex}(e|f)$ are four of the standard static features used in phrase tables. These features are computed directly from the co-occurrence of aligned phrases in the training corpora. However, co-occurrence information alone cannot capture semantic information about phrases, especially when they are taken out of context. Therefore, many techniques have been proposed to enrich the feature list by including features which contain syntactic and/or semantic information (Banchs and Costa-jussà, 2011).

Most work evaluating the inclusion of semantic information into SMT decoders has focused upon adding dynamic features (those which must be computed at decoding time). However, dynamic features require the implementation of a new feature function which depends upon the hypothesis data structure of the particular decoder. The implementation of dynamic features typically requires significantly more engineering effort than simply augmenting the phrase table. In this paper, we add a new static feature and show how a good static feature alone can significantly boost translation quality. The basic idea behind our work is to use the vector representation, or semantic embedding of phrases, which is generated by an NN. The semantic features of the source and target languages are projected into a shared bilingual space,
which preserves both semantic and syntactic information about the phrases. The supervised
approach to generating embeddings from neural networks optimizes the network to produce
vectors which are good with respect to a specific objective. As an example, if the goal is to do
the sentiment analysis or sentence clustering, a vector should reflect the polarity of a sentence —
whether is positive or negative — or its closeness to a specific distribution (Kalchbrenner et al.,
2014). In tasks like translation, a good vector should reflect semantic and syntactic information
about original constituent (sentence, phrase or word) in addition to contextual knowledge from
its surrounding words, and ideally some information which will make it easier to map into the
target language.

Methods like word2vec\(^1\) (Mikolov et al., 2013a) or that of (Le and Mikolov, 2014) produce
general-purpose vector representations which can be leveraged for a variety of downstream
applications. As the word and sentence vectors encode syntactic and semantic information, they
are potentially useful for translation tasks. In pure neural MT engines, the word embeddings
are trained as parameters of the model, which generally attempts to maximize the likelihoods
(Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2014), and then used directly to perform
translation. Another line of work has tried to make use of distributed representations within the
classic MT pipeline (Gao et al., 2013; Devlin et al., 2014).

In this work, we prepare an enhanced bilingual corpus which includes the source and target
phrases extracted by the alignment model, along with the original source and target sentences,
and a set of word pairs which are direct translations of one another. An NN is used to generate
embeddings for the each of the phrases. The generated vectors reflect the similarity of phrases in
the same language as well as their relevancy to the phrases of the other language. In the training
data, monolingual distributional information for each language is contained in the sentences,
while bilingual information is conveyed by the bilingual phrase pairs and word pairs.

Our contributions in this paper are twofold. a) We extract a novel static feature from a
bilingual corpus which boosts the translation quality. Most previous work has added new dy-
namic features which significantly increase computational overhead — our simple static feature
can achieve comparable improvements with less effort. b) Our network extracts the vectors
from a bilingual corpus. To the best of our knowledge, related research has modeled the source
and target vectors separately in isolated spaces and focused upon finding a means to transform
the source & target representations into comparable forms. We extracted vectors in the joint
space and tried to capture information of the both target and source sides in a single vector.
Inferred feature caused considerable improvements and showed that, a good static feature can
perform as efficiently as a dynamic one. The structure of paper is as follows. Section 2 gives an
overview of related work, and tries to show why word, phrase, and sentence vectors are useful
for MT purposes. Section 3 explains the network architecture in detail. In Section 4, experi-
mental results are reported for two language pairs: En–Fr and En–Fa. Finally, in the last section
we present our conclusions and discuss some avenues for future work.

2 Background

Using vector representations for textual data (word, phrase, sentence, etc...) is not a new
idea. The concept of continuous, distributed representations for text tokens was introduced
by the Vector Space Model of Salton et al. (1975) and expanded by techniques such as like
Latent Semantic Analysis (Deerwester et al., 1990), Latent Dirichlet Allocation (Blei et al.,
2003), and Random Indexing (Sahlgren, 2005). Real valued, continuous representations are
straightforward to use within Machine Learning models, and have contributed to the current
state-of-the-art in many NLP tasks. However, a disadvantage of distributed representations is

\(^1\)http://code.google.com/p/word2vec/
that they are typically not decomposable — i.e. columns often do not correspond to intuitive features, thus models must be evaluated on a task-by-task basis.

The techniques mentioned in the previous paragraph build vectors from bag-of-words (BOW) representations, discarding structural dependencies and word order, so they cannot not reflect semantic information which depends upon syntactic structure. To compensate for these deficiencies, a more recent line-of-work uses neural networks to generate vectors which can encode local distributional information, as well as syntactic structure and word order in some cases. Hinton (1986) used NN for text modelling for the first time. Recently, neural networks have achieved state-of-the-art performance in many areas of NLP due both to the development of new learning algorithms and to the availability of computational resources such as GPUs. Word2vec (Mikolov et al., 2013a), and word2vec inspired works (Wolf et al., 2014) have been successfully applied to a wide variety of NLP tasks. The following paragraph focuses upon work that leverages these vectors for MT purposes.

A basic but successful application of NN-based word vectors for MT was reported in (Mikolov et al., 2013b). They project words of the source language into vectors and do the same with words of target language. Then try to find a transformation function which maps the source semantic space into the target semantic space using a small set of word pairs known to be high-quality translations. The model significantly reduces the volume of bilingual data required to train such systems and this is the main advantage of their approach. The model works on monolingual data and only needs a small number of parallel words to make the bridge between languages. The cross-lingual transformation allows an MT system to search for translations for OOV (out-of-vocabulary) words by consulting a monolingual index which contains words that were not observed in the parallel training data for the MT system. Garcia and Tiedemann (2014) and Dinu and Baroni (2014) are other examples of approaches which leverage NN-based word vectors for translation tasks. They focus upon exploiting similarities at the word level, but MT encompasses more than just word-level translation. To extend the application of text embeddings beyond single words, Gao et al. (2013) proposed learning embeddings for source and target phrases by training a network to maximize the sentence-level BLEU score. The outcome is a set of vectors for phrases, and the similarity between each phrase pair vectors is used as a dynamic feature function in the log–linear model at decoding time. In another work Costa-Jussà et al. (2014) tried to find the similarities among source sentences and incorporate source side contextual information into the decoding. Some other models try to re-score the phrase table or infer new phrase pairs to address the OOV word problem in order to improve translation quality (Alkhouli et al., 2014; Costa-Jussà and Banchs, 2011). Zhao et al. (2015) did the same using monolingual datasets and extended the phrase table.

3 Learning Embeddings for Phrases

Our model extends the document vectors of Le and Mikolov (2014) to bilingual texts. By including both monolingual and bilingual ‘documents’ into the training data, we learn a distributed representation for both languages simultaneously. In the method proposed by Le and Mikolov (2014), documents are treated as atomic units in order to learn an embedding with the same dimensionality as the vectors for the individual words in the model. We adapt this idea to sentences and phrases, where phrases are presented both as monolingual and as bilingual documents.

Le and Mikolov (2014) create a new vector for each document, which in our case may be a monolingual sentence, a monolingual phrase, a bilingual phrase pair, or a bilingual word pair. During training, the document vector is concatenated with the vectors for individual words to predict the surrounding words in the given unit of text. Intuitively, we expect document vectors to be representative of the semantic content of the entire unit of text, while word vectors are
representative of all of the contexts where the word occurs. During training, word vectors and sentence/phrase vectors are updated until the cost is minimized. The model learns a semantic space where constituents with similar distributional tendencies tend to have similar vectors. More formally, given a sequence of tokens $c_1, c_2, ..., c_n$ the objective is to maximize the average log probability of a word given its context:

$$\frac{1}{n} \sum_{i=l}^{n-l} \log p(c_i | c_{i-l}, ..., c_{i+l})$$

Using the standard terminology for NN models, the training objective can be explained as follows. The cost function is defined according to the average log probability. Values for the probabilities can be calculated using a multiclass classifier such as softmax:

$$p(c_i | c_{i-l}, ..., c_{i+l}) = \frac{e^{y_{w_i}}}{\sum_j e^{y_j}}$$

where $y_j$ is the output for the input word $w_j$ in which:

$$y = b + Wh(c_{i-l}, ..., c_{i+l})$$

$W$ and $b$ are parameters for the Softmax function, and $h$ represents the output of one or more hidden layers. The network is trained by stochastic gradient decent and back-propagation (Rumelhart et al., 1988) to obtain the set of constituent vectors $C$, and network parameters, $W$ and $b$.

In our case, the network includes one hidden layer with 100 nodes which is fed by a bilingual corpus. The Corpus includes a) source and target sentences which are those we used to train our SMT engine, b) phrase pairs and c) bilingual lexicon. Both the phrase and lexicon sets are extracted by Moses (Koehn et al., 2007). Specifically, each line of corpus contains a sentence in one of the languages, a phrase pair, or a tuple of words.

As it has been shown in (Huang et al., 2012), word vectors can be affected by the word’s surroundings as well as by the global structure of a text. Accordingly, by using a training corpus with some bilingual examples we expect to learn a semantic space which contains both languages. Each unique word has a specific vector representation. Clearly similar words in the same language would have similar vectors (Mikolov et al., 2013a). Words that are direct translations of each other (same meaning with different languages) should also have similar vectors. As the corpus contains the tuples of $<word_{L1}, word_{L2}>$, equal words are connected together. By the same logic, phrasal units are also connected together. During decoding, the sentences in a good translation pair should be built from similar sub-units, indicating the semantic compatibility of the constituent phrases. Table 1 shows the most similar phrases and words for two
examples. The items that were originally in Farsi have been translated into English, and are indicated with *italics*.

**Table 1:** The top 10 most similar vectors for two queries: an English phrase, and a Farsi word. Recall that the index includes vectors for words in both languages, phrases in both languages, sentences in both languages, bilingual phrase pairs, and bilingual lexical pairs. Farsi words are indicated in *italics*.

<table>
<thead>
<tr>
<th>Query</th>
<th>we can’t let them win</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>you could ever</td>
</tr>
<tr>
<td>2</td>
<td>what the bloody hell is that .</td>
</tr>
<tr>
<td>3</td>
<td>as you know it .</td>
</tr>
<tr>
<td>4</td>
<td>let her go .</td>
</tr>
<tr>
<td>5</td>
<td>he is lying . no i am not .</td>
</tr>
<tr>
<td>6</td>
<td>to be</td>
</tr>
<tr>
<td>7</td>
<td>he won</td>
</tr>
<tr>
<td>8</td>
<td>you !?</td>
</tr>
<tr>
<td>9</td>
<td>you got that .</td>
</tr>
<tr>
<td>10</td>
<td>they are ahead</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query</th>
<th>sadness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>&lt;apprehension, nervous&gt;</em></td>
</tr>
<tr>
<td>2</td>
<td>emotion</td>
</tr>
<tr>
<td>3</td>
<td><em>&lt;ill, sick&gt;</em></td>
</tr>
<tr>
<td>4</td>
<td>pain</td>
</tr>
<tr>
<td>5</td>
<td><em>&lt;money, money&gt;</em></td>
</tr>
<tr>
<td>6</td>
<td>benignity</td>
</tr>
<tr>
<td>7</td>
<td><em>&lt;may he was punished, punished harshly&gt;</em></td>
</tr>
<tr>
<td>8</td>
<td>is really gonna hurt</td>
</tr>
<tr>
<td>9</td>
<td>i know toms dying</td>
</tr>
<tr>
<td>10</td>
<td><em>&lt;bitter, angry&gt;</em></td>
</tr>
</tbody>
</table>

Table 1: The top 10 most similar vectors for two queries: an English phrase, and a Farsi word. Recall that the index includes vectors for words in both languages, phrases in both languages, sentences in both languages, bilingual phrase pairs, and bilingual lexical pairs. Farsi words are indicated in *italics*.

The pipeline for adding our semantic feature to the phrase table is very straightforward. We have a set of vectors for the phrases of both languages. Each phrase is modelled with a 100-dimensional vector. The phrase table is scanned sequentially, and for each phrase pair, related vectors are fetched, then their similarity is estimated. To measure the similarities we use the cosine metric, which is:

\[
similarity(v_1, v_2) = \frac{v_1 \cdot v_2}{||v_1|| \ast ||v_2||}
\]

Values from the cosine similarity are in the range [-1,1]. We map the similarity scores to the range [0,1] before adding them to the phrase table. After including the new feature, the weights for the log-linear model are learned by using held-out data.

**4 Experimental Results**

To study the impact of our new feature, we selected two datasets. One is the TEP++ corpus (Passban et al., 2015) which is a collection of 600K parallel En–Fa sentences. Farsi is a particularly interesting language for new MT research because it is both low resource and morphologically rich. The state-of-the-art MT quality for Farsi is not advanced relative to languages with
more training data available. The other dataset is the WMT\(^2\) Fr-En corpus. The WMT corpora are frequently used as the de facto standard test datasets for SMT systems.

In all of the experiments Moses (Koehn et al., 2007) is used to build the SMT engines. BLEU (Papineni et al., 2002) is the evaluation metric and the feature weights are tuned by MERT (Och, 2003). All language models are built using SRILM (Stolcke et al., 2002). The improvements in translation quality are statistically significant according to the results of paired bootstrap re-sampling (Koehn, 2004).\(^3\) The experimental setup is shown in the Table 2 and results from the baseline and extended systems are in Table 3.

### Table 2: Experimental setup

<table>
<thead>
<tr>
<th>Language pair</th>
<th>Dataset</th>
<th>MT engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>En–Fr</td>
<td>Europarl corpus (Koehn, 2005) version 7 is used as a bilingual training set. As a dev set 2000 sentences of news-test2014 were used and the test set is the test set of the WMT2015 shared translation task.</td>
<td>statistical phrase based engine with default Moses configuration</td>
</tr>
<tr>
<td>En–Fa</td>
<td>The training, test and dev sets are subsets of the TEP++ corpus consisting of 575191, 2000 and 1000 sentences respectively. All the sentences have been selected randomly.</td>
<td></td>
</tr>
<tr>
<td>Fa–En</td>
<td>5-gram language models trained on the monolingual part of the training sets.</td>
<td></td>
</tr>
</tbody>
</table>

As the results show, the new feature leads to improvements in all directions. We anticipated these improvements, as the positive impact of distributed phrase representation has already been shown by Gao et al. (2013). However, our work consider two new aspects of the problem. We train the vectors in a shared bilingual space, and show that proposed model can generate similar vectors for similar/equivalent constituents, even for languages pairs such as En-Fa, which are not typologically similar. We also show that a simple but efficient static feature can improve translation quality.

### Table 3: Results for base-line and extended systems

<table>
<thead>
<tr>
<th></th>
<th>En–Fr</th>
<th>Fr–En</th>
<th>Fa–En</th>
<th>En–Fa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>29.91</td>
<td>27.31</td>
<td>29.21</td>
<td>21.03</td>
</tr>
<tr>
<td>Extended</td>
<td>30.62</td>
<td>27.95</td>
<td>29.72</td>
<td>21.44</td>
</tr>
<tr>
<td>Improvement</td>
<td>+0.71</td>
<td>+0.64</td>
<td>+0.51</td>
<td>+0.41</td>
</tr>
</tbody>
</table>

The new semantic similarity feature causes, on average, +0.56 enhancement in terms of BLEU for all of the directions of En$\leftrightarrow$Fr and En$\leftrightarrow$Fa, and as the size of training data increases the method provides even better performance. Improvements for the En–Fr pair demonstrate that achievements of the model are valid for large datasets, and improvements for the En–Fa pair show that the model can be used to translate distant language pairs. The word order and structure of the Farsi and English languages are quite different from each other, and Farsi is a morphologically rich language, making translation more difficult than for closely related language pairs such as En-Es.

\(^2\)http://www.statmt.org/wmt15/translation-task.html

\(^3\)We used ARK research group codes for statitical significance testing for 1000 samples with parameter of 0.05, http://www.ark.cs.cmu.edu/MT/
5 Conclusion and Future work

In this work we presented a new bilingual semantic similarity feature obtained from a neural network that is trained on a bilingual corpus, and computes the distributed representation of phrases in a shared semantic space. Each phrase is projected into a vector and the similarity of the vectors for each phrase pair is estimated. The similarity score for the phrase pair is added as a new phrase table feature, and the MT engine is tuned according to the default features in addition to new one. This augmentation of the information in the phrase table provides improvements in translation quality.

The method is quite straightforward and does not impose any significant overhead to the baseline SMT pipeline. Distributed vector representations preserve the semantic information of the constituents as well as their order and structural dependencies. The bilingual examples in the training data create dependencies between the equivalent constituents from different languages. As the model connects the phrases of two different languages to each other, it implicitly includes contextual information about the phrase pair into the MT process. Our next goal is to incorporate information from the source and target sides at decoding time. Although our model provides a global measure of the quality of a phrase pair, we cannot use the current framework to do tasks like disambiguation, because our features are static. We hope to incorporate the knowledge from paragraphs and text segments that the source and target phrases are extracted from, and compare this information to the context of the phrase at decoding time in order to provide a dynamic means of computing cross-lingual similarity.

Acknowledgement

We would like to thank the three anonymous reviewers and Rasul Kaljahi for their valuable comments. This research is supported by Science Foundation Ireland through the CNGL Programme (Grant 12/CE/I2267) in the ADAPT Centre (www.adaptcentre.ie) at Dublin City University.

References


Improved Search Strategy for Interactive Predictions in Computer-Assisted Translation

Fatemeh Azadi
Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
ft.azadi@aut.ac.ir

Shahram Khadivi
eBay Inc, Aachen, Germany
skhadivi@ebay.com

Abstract
The statistical machine translation outputs are not error-free and in a high quality yet. So in the cases that we need high quality translations we definitely need the human intervention. An interactive-predictive machine translation is a framework, which enables the collaboration of the human and the translation system. Here, we address the problem of searching the best suffix to propose to the user in the phrase-based interactive prediction scenario. By adding the jump operation to the common edit distance based search, we try to overcome the lack of some of the reorderings in the search graph which might be desired by the user. The experiments results shows that this method improves the base method by 1.35% in KSMR$^2$, and if we combine the edit error in the proposed method with the translation scores given by the statistical models to select the offered suffix, we could gain the KSMR improvement of about 1.63% compared to the base search method.

1. Introduction
Although the significant improvements achieved in the field of statistical machine translation, the current models and therefore the systems which can be built from them are still far from perfect. So, in order to achieve good or even acceptable translations, manual post editing is needed. An alternative to this approach is given by the interactive predictive machine translation paradigm (Barrachina et al., 2009). Under this paradigm, translation is considered as an iterative process where the human translator and the computer collaborates to generate the final translation, and in this way both the human and the computer can help each other.

In the interactive predictive MT scenario, the system initially offers a translation for a source sentence. From this translation, the user could mark a prefix as correct and begins to type the rest of the target sentence as it’s desirable for him. By typing the single next character (or maybe the full next word, according to the system settings) the interactive MT system would suggest a new suffix that completes the user’s confirmed prefix with the new character the user has just typed. This procedure continues iteratively until the user accepts the translation to be complete and correct.

One of the important issues in the development of interactive predictive MT systems is the strategy to search and offer the best suffix that completes the prefix confirmed by the user. One of the most common ways for phrase-based translation systems is to use the search graph that is obtained once during the translation process, and complete each user’s confirmed pre-
fix by searching the best path which has a prefix with highest matching to the user’s prefix, and offering the rest of it to the user. The most important advantage of using the pre-built search graph in each interaction is that it leads to an efficient response time for each interaction.

The problem in searching the best system offer arises when the prefix confirmed by the user is not producible by the statistical models of the translation system. That is it could not be produced by any of the paths in the search graph, so that the interactive predictive MT system could not offer any completion to the user. A common solution to this problem is that instead of searching a path in the graph whose prefix matches exactly with the user’s prefix, we search more freely, for a path which has the most similarity with the user’s prefix. Usually, the edit distance similarity measure is used for this purpose.

The search graph may not contain all of the possible reorderings for a certain set of words, due to the weakness of translation and reordering models or the pruning done during the generation of the search graph. So, the user’s desired translation might have a reordering which doesn’t exist in any of the search graph paths and the search based on edit distance could not perform well in these cases, since it aligns the user’s prefix and a prefix of each path in the search graph monotonically and with no reordering.

In this paper we try to solve this problem for phrase-based interactive MT systems, by adding the jump operation in computing edit distance, which allows jumping over some parts in each path while aligning with the user’s prefix, and then offering those parts as the suffix to the user; i.e. shifting some parts to a position after user’s prefix, so that it changes the reordering of the system’s translation.

Also, we compare and combine this method with another improvement done to the edit distance based search in (Vakil and Khadivi, 2012), which uses the weighted sum of the edit distance and the translation score for each path for selecting the best suffix.

In section 2 the related works done in the computer assisted translation field are reviewed. In section 3, we briefly explain the statistical framework for the interactive predictive MT. Then in section 4, the searching algorithms for finding the best suffix are discussed and the proposed method in this paper is presented. Finally, the experiments done for the evaluation of our method is presented in section 5.

2. Related Work

Interactivity in CAT\(^3\) has been explored for a long time. Systems have been designed to interact with human translators in order to solve different types of (lexical, syntactic, or semantic) ambiguities (Slocum 1985; Whitelock et al. 1986).

A major change in the interactive CAT technology was done within the TransType project in (Foster et al. 1997; Langlais et al., 2000). A new approach called Target-Text Mediation was proposed, in which the focus of the interaction was shifted from the source text to the production of the target text directly. This project with TransType2 (TT2) (Bender et al., 2005; Tomás and Casacuberta 2006; Barrachina et al., 2009) proposed to embed an MT system in an interactive scenario. This way, the human translator can ensure a high quality output while the MT system ensures the significant gain of productivity.

Particularly the interactive-predictive machine translation paradigm was proposed in (Barrachina et al., 2009). In this paradigm a statistical MT system uses the source sentence and a prefix of it’s translation which is confirmed by the user to propose a suitable continuation. And the text would be translated in an iterative process of interactions between

---

\(^3\) Computer Assisted Translation
the user and the interactive MT system. The results showed that interactive MT can save a significant amount of human effort. Also different translation models were tested in (Barrachina et al., 2009), and the results showed that the phrased based models have a better performance than the other ones.

After TT2 some sporadic works were done on developing CAT systems (Ortiz-Martínez et al., 2009; Huang et al., 2012), but the more famous one was the Caira system (Koehn 2009; Koehn and Haddow, 2009). It was a web-based system, which could be accessed online by the users. The search method used as the base method in this paper, which will be discussed in section 4.1, is the method proposed and implemented in this system.

Recently two other CAT projects, CASMACAT⁴ and MATECAT⁵ were done, which were committed to develop an open source workbench targeted both at researchers to investigate novel and enhanced types of assistance and at professional translators for actual use (Federico et al., 2014; Alabau et al., 2014). One of the goals of these projects was to gain insight into the cognitive processes involved in human translation, by using eye tracking and key logging. The cognitive analysis will determine what types of assistance are offered to the translator, what information should be displayed on the screen, and what information should be hidden as it would be distracting. Also they try to add the capability of the automatic adaptation to the translated content to the CAT tools, which is not related to the subject of this paper. Also the interactive-predictive CAT scenario when the user pronounce the correct translation instead of typing it has been studied in (Vakil and Khadivi, 2012b).

The other work done to improve the searching methods for the interactive prediction is (Vakil and Khadivi, 2012a), which tries to consider both the edit distance and the translation score of each path in the search graph instead of just using the edit distance. This approach, which is called the weighted edit distance, is discussed more in section 4.2.

Also in (Ortiz-Martínez, 2011) and (González-Rubio et al., 2013) a search approach is proposed, which is somewhat alike the weighted edit distance approach. But they introduce a unified statistical framework, which integrates all of the statistical models used in the translation machine with the edit distance error, which is called error correction model here. Also the proposed approach, models the edit distance as a Bernoulli process where each character of the candidate string has a probability of being erroneous.

The jump operation that we used here is somehow like the shift operation used in Translation Error Rate (Snover et al., 2006), which is an evaluation metrics for machine translation and measures the number of edits required to change a system output into one of the references.

3. Interactive-Predictive Machine Translation

In the interactive predictive machine translation paradigm, we want to produce a suffix which is the best according to the user’s prefix and the source sentence. So in the statistical approach, we should maximize the probability of the suffix given the source sentence and the confirmed prefix, as follows:

\[ \hat{t}_s = \arg \max_{t_i} \Pr(t_i \mid s, t_p) \]

Where \( t_i \) is the suffix, \( t_p \) is the user’s confirmed prefix, and \( s \) is the source sentence. We can write the above equation as follows:

⁴ http://www.casmacat.eu/index.php
⁵ http://www.matecat.com/
\[ \hat{t}_s = \arg \max_{t_s} \Pr(t_p, t_s \mid s) = \arg \max_{t_s} \Pr(t_p, t_s) \cdot \Pr(s \mid t_p, t_s) \]

Noting that \( t_p t_s = t \), this equation would be very similar to the following, which is the statistical machine translation’s equation.

\[ \hat{t} = \arg \max_t \Pr(t) \cdot \Pr(s \mid t) \]

The main difference between these two is that in the interactive MT the search process is restricted to those target sentences that contains \( t_p \) as prefix. This implies that we can use the same MT models if the search procedures are adequately modified (Och et al., 2003).

One of the most common ways, that we used here, for searching the best suffix in the interactive MT is to use the search graph that is obtained once during the translation process, and complete each user’s confirmed prefix by searching the best path which has a prefix which matches to the user’s prefix, and offering the rest of it to the user. The most important advantage of using the pre-built search graph in each interaction is that it leads to an efficient response time for each interaction. However, there’s not always a path in the search graph whose prefix matches exactly with the user’s prefix, so some mechanisms should be chosen the search for a path which has the most similarity with the user’s prefix. A common approach is to use the edit distance measure which will be discussed in the next section.

4. Searching Algorithms for the Interactive Prediction

In this section, we first introduce the searching strategy based on the edit distance, which is presented in (Koehn, 2009), and is used as our base search algorithm. Then we try to improve this algorithm in two ways; first by adding the edit distance error as a weighted model to the translation scores computed for each path by the decoder, and using this score for selecting the best path, as in (Vakil and Khadivi 2012). And in the second method we add a jump operation to the computation of the edit distance, to consider the reorderings which are not in the search graph but might match with the user’s prefix.

4.1. Searching Based on Edit Distance

According to (Koehn, 2009), for giving a completed translation offer to the user, we should find a hypothesis (or node) in the graph whose generated partial translation has the minimum edit distance with the confirmed prefix; This approach is called edit distance-based search. The edit distance between two strings is defined as the minimum number of edits needed to transform one string into the other, while the allowable edit operations are insertion, deletion, or substitution (Levenshtein, 1966). Although these operations could be done at character or word level, the word level distance makes more sense for our purpose, as performing one or more character level operations on a word might change the whole meaning of that word, and the number of operations done is not matter.

In this method the prediction is the optimal completion path that matches the user’s prefix with minimal edit distance, and if there were more than one such paths, the one with highest translation score will be chosen. This approach is based on the assumption that a hypothesis with minimum edit distance with the confirmed prefix has the continuation which is more likely to be consistent with the remained part of the user’s desired translation.

This search method has shortcomings, as the search graph may not contain all of the reordering possibilities of words due to the pruning that is applied during the generation of the search graph. Since the edit distance only includes the insertion, deletion and substitution op-
erations, it could not be able to consider the difference in reorderings between the user prefix and any of the translation hypotheses in the search graph. Thus, it’s only able to find those hypotheses which have a similar reordering of words to the user’s prefix. In order to solve this problem we discuss two approaches in the next sections.

4.2. Searching Based on Weighted Edit Distance

This approach which is presented in (Vakil and Khadivi, 2012), is based on the fact that the paths in the search graph that have higher translation scores according to the translation and language models, are more probable to be better translations in terms of both semantics and syntax. However, these better translation hypotheses might not sometimes have lowest edit distance with the user’s prefix due to the different reordering of words they have. Thus, using the edit distance alone for selecting the best path, regardless of its translation score, would select another path in the graph that might be an unfavourable translation.

To clarify this, consider the example shown in Table 1. We have the user’s desired translation and two translation hypotheses samples, extracted from the search graph with their translation scores, for a specific sentence. Assume that the prefix confirmed up to now by the user is «Newton is one of the greatest scientists w», and the system should now offer a continuation for it.

Table 1. Example 1; the user's desired translation and system’s translation hypotheses

| User's desired translation: Newton is one of the greatest scientists who discovered gravity | Score: -5.25 |
| Translation hypothesis 1: one of the greatest scientists is Newton who discovered gravity | |
| Translation hypothesis 2: Newton gravity one of the greatest discovered which is the greatest | Score: -10.23 |

For offering the rest of the translation by the method based on the edit distance, the translation hypothesis whose prefix has less edit distance with the confirmed prefix, should be chosen.

The least edit distance for each of the hypotheses is shown in Table 2. The first hypothesis needs 2 insert operations and 2 delete operations, and the second hypothesis needs just 2 substitutions to match the user’s prefix. So the second hypothesis which is not a good translation at all would be chosen, and the completion offered to the user would be «which is the greatest», which is obviously not desired. Instead, if we consider the translation scores along with the edit distances the first hypothesis could be chosen that has a much better offer for the continuation.

Table 2. The least edit distance between each hypothesis and user's prefix in example 1

| # of edits | Newton is one of the greatest scientists who discovered gravity | 4 |
| 2 | Newton one of the greatest discovered which is the greatest |

As the previous example shows, using only the edit distance to find the best path may lead to a hypothesis which has low quality according to the translation and language models. To overcome this problem, in this approach the total score for each path that a prefix of it matches with the user’s prefix, would be a weighted summation of the edit distance with the
translation scores obtained by the statistical models, and in the search process we look for a hypothesis with best total score.

4.3. Edit Distance with Jump Operation

The weighted edit distance approach somehow overcomes the problem of lack of some reorderings in the search graph indirectly in some cases. In order to targeting this problem and resolving it directly, in this paper we propose another method that gives more freedom to the edit distance based searching by adding another operation, called jump operation.

To better explain the aim of adding jump operation and the effect that it could have on the system’s offers, consider the example shown in Table 3. In this example, alike the previous one we have the user’s desired translation and two system’s translation hypotheses, except that here both hypotheses are good enough translations.

Table 3. Example 2; the user’s desired translation and system’s translation hypotheses

<table>
<thead>
<tr>
<th>desired translation:</th>
<th>Newton who discovered gravity is one of the greatest scientists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translation hypothesis 1:</td>
<td>one of the greatest scientists is Newton who discovered gravity</td>
</tr>
<tr>
<td>Translation hypothesis 2:</td>
<td>Newton is one of the greatest scientists who discovered gravity</td>
</tr>
</tbody>
</table>

Assume that the user’s confirmed prefix is “Newton who discovered », and the system should now give an offer. For this purpose the minimum edit distance between the confirmed prefix and any prefix of each of the translation hypotheses should be computed, which is shown in Table 4.

Table 4. The least edit distance between each hypothesis and user's prefix in example 2

<table>
<thead>
<tr>
<th>Edits</th>
<th>hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>one of the greatest scientists is Newton who discovered gravity</td>
</tr>
<tr>
<td>2</td>
<td>Newton is one of the greatest scientists who discovered gravity</td>
</tr>
</tbody>
</table>

According to Table 4, if we use only the edit distance, the second hypothesis will be chosen and the system’s offer would be “of the greatest scientists who discovered gravity”. Also if we use the weighted edit distance method, the first hypothesis might be selected due to its better translation score, and the system might offer “greatest scientists is Newton who discovered gravity”. But obviously none of these offers are desirable by the user.

This problem caused by the fact that we consider the alignment between the user’s prefix and the prefix of the hypothesis is monotone and with no reordering, while due to the weakness of translation and reordering models and also the pruning done during the generation of the search graph, many of the reordering possibilities might not be produced or might get low scores and be pruned. So in some cases, the part that the user translates at the beginning is at the end or at the middle of the system’s translations. In these cases, if we could shift the parts of the hypothesis that match with the user’s prefix to the beginning of the sentence and arranged the remained parts in a good order (as more than one segments might be remained, that needs concatenating to make a single segment), we could propose a better offer. For instance, in the hypothesis 2 of the above example, if we could shift “who discovered
gravity“ after the word “Newton” at the beginning of it, we could get the user’s desired translation.

For this purpose, we define jump operation in the edit distance. With this operation we can jump from some parts of each translation hypothesis and after matching the user’s prefix completely, we collect the parts that we jumped over with the last remained part and arrange them in a good order to generate the system’s offer.

As using the jump operation without any limitations enlarges the search space exponentially, we just allow one jump with variable length over multiple edges in each path in the search graph. This is equivalent to allowing just one jump over some contiguous phrases in each system’s translations. Also the jump operation could add an error to the edit distance value. Different amounts of errors considered for the jump operation, which will be discussed in the experiments, but the best was adding one unit of error for each jump with any length.

In the above example we can match “Newton” in user’s prefix and hypothesis 2, then by jumping from “is one of the greatest scientists”, we match the part “who discovered” with the rest of the user’s prefix, and the part “gravity” will remain at the end of the hypothesis. So, with just one jump operation we could match the user’s prefix and the hypothesis.

The subject which is left is how to arrange the jumped over and remained parts to generate the final offer. As we could have at most one jumped over part and one ending part, we consider the two permutations of attaching these two parts; i.e. first the jumped over part comes and then the end part, or first the end part comes and then the jumped over part. Then for selecting between these two states we use language model, and the one whose language model score according to the user’s prefix is more will be chosen.

For the above example, the two sentences “Newton who discovered is one of the greatest scientists gravity” and “Newton who discovered gravity is one of the greatest scientists” will be scored with the language model, and as the second one is more well-formed the final offer to the user would be “gravity is one of the greatest scientists”.

The pseudocode of the search algorithm based on weighted edit distance with jump operation is shown in Figure 1. The base of this implementation is like the edit distance approach presented in (Koehn, 2009). This algorithm associates backpointers with each hypothesis (or node) in the graph that point back to the lowest error path with which it can be reached. Each backpointer saves the number of edit distance errors, the translation score and the position in the user’s prefix that the edit distance is computed up to there till the backpointer’s associated hypothesis. As each hypothesis may match the confirmed prefix at different positions, there are multiple backpointers for each hypothesis.

For adding the jump operation a variable named jumpState added to backpointers, which shows the state of the jump operation till that backpointer. If no jump had been occurred until this point of the path, this variable is zero, if in the previous step of the backpointer’s path a jump happened, this variable should be 1 (means the jump could continue in the next step or terminate here), and if a jump had been done and finished before this backpointer in its path, the jumpState value would be 2.

The algorithm starts from the first node (empty hypothesis) and iterates over the nodes in the search graph in the topological order. When examining each node’s backpointers, all forward transitions (edges of the search graph) are examined using a string edit distance between the remaining prefix and the words in the transition target phrase (line 10). This may consume the remaining prefix, and possibly lead to a new best path whose total score is better than the best total score found so far (lines 11-18). Otherwise, new backpointers for the forward states are created (lines 19-21).
Figure 1. The Pseudocode for searching based on weighted edit distance with jump operation

Input: user prefix (u), search graph (g), weight of edit distance error (α)
Output: best completion offer

1. \( \text{bestTotalScore} = \infty \), \( \text{bestPath} = \{ \} \)
2. Add backpointer (\( \text{Score} = 0.0 \), \( \text{Error} = 0 \), \( \text{prefixMatched} = 0 \), \( \text{jumpState} = 0 \)) to start state
3. for all state \( s \in G \) in topologically increasing order do
   4.   for all backpointer \( b \) of state \( s \) do
      5.     if \( \text{bestTotalScore} \leq b.\text{Score} + s.\text{forwardScore} + \alpha \cdot b.\text{Error} \) then
          6.             for all transition \( t \) from state \( s \) do
              7.               if \( b.\text{jumpState} \neq 0 \) \( \| \) \( b.\text{jumpState} = 1 \) then
                  8.                 ProcessJump(\( S \), \( b \), \( t \) )
                9.             end if
          10.            Compute edit distance matrix for end part of \( u \) after \( b.\text{prefixMatched} \) and \( t.\text{phrase} \)
          11.           for all matches \( m \) in matrix that consumed all of \( u \) do
              12.               new score \( c = b.\text{Score} + t.\text{Score} + t.\text{toState}.\text{forwardScore} \)
              13.               new error \( e = b.\text{Error} + m.\text{Error} \)
              14.               if \( c + \alpha \cdot e > \text{bestTotalScore} \) then
                  15.                 set this as \( \text{bestPath} \)
                  16.                 \( \text{bestTotalScore} = c + \alpha \cdot e \)
              17.           end if
            18.       end for
          19.       for all matches \( m \) in matrix that consumed all of \( t.\text{phrase} \) do
              20.               ProcessMatch(\( S \), \( b \), \( t \), \( m \) )
            21.       end for
       10.     end if
      11.   end for
  3. end for
26. return the best ordering for jumped part and the end part in \( \text{bestPath} \) using LM

Also, for each forward transition of a specific backpointer, the jumpState of that backpointer will be checked and if it is 0 or 1 it means that a jump could be done over that forward transition. Therefore, this operation performs and the new backpointer for this case is also created and added to the forward state’s backpointers (lines 7-9).

The ProcessJump function, shown in Figure 2, will make the new backpointer for performing the jump operation at each state. The error which is considered for this case is one if it is the beginning of a jump, and zero otherwise; i.e. for each jump with any length, 1 unit of error is considered (line 5).

Also, the ProcessMatch function will make the new backpointers for each case of matching a part of user’s prefix with the translation phrase produced at the processing transition, at each stage. The jumpState for the new backpointer will remain zero if the previous backpointer’s jumpState was zero (means that the jump operation could be performed later in the path), and it would be 2 otherwise (which means the jump operation was performed before, its finished and could no longer be used in the rest of this path) (line 6).
5. Experiments

In this section, we discuss the experiments done to evaluate the searching algorithms presented in the previous section for interactive prediction.

5.1. Evaluation Metrics

The common way for evaluating the performance of the interactive predictive machine translation systems automatically is to estimate the effort of a human translator to produce correct translations using that system. For this purpose, the translating process of the user is simulated using a set of reference translations as the desired translations in the user’s mind.

In this simulation, for each given source sentence in the test set, the translation is produced by the interactive MT system at the first step. Then it’s compared with the reference
given for that source sentence, and the character-level longest common prefix of them will be computed. Afterwards, the first non-matching character is replaced by the corresponding reference character and then the system offers a new completion for the given prefix. This process is iterated until a full match with the reference is obtained. Each time a part of the new system’s offer is accepted by the user and it expands the prefix, a mouse action is done, and each time the new character is added to the prefix from the reference a keystroke is considered to be done by the user.

The measure used here for evaluation is KSMR, which is the overall number of interactions, i.e. keystrokes and mouse-actions, done to translate the whole test set divided by the number of running characters in the reference translations.

### 5.2. Translation System Description

In this work, we use a Farsi to English phrase-based translation system to evaluate our proposed method. We use Giza++ (Och and Ney, 2003), to train the translation models and Moses Toolkit (Koehn et al., 2007) as the MT decoder, as well as the SRILM tools to build the language model. The interactive system is implemented as a server using Moses decoder, which gets a new source sentence or the user’s confirmed prefix each time and gives the continuation offer to the client, while the client simulates the user’s interactions.

The statistics of the corpora used for training the translation system (Jabbari et al., 2012) and the test set is shown in Table 5.

#### Table 5. The statistics of the training and testing corpora

<table>
<thead>
<tr>
<th>Corpora</th>
<th># of Sentences</th>
<th># of Running Words</th>
<th># of Lexicons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Farsi</td>
<td>2164408</td>
<td>37356049</td>
<td>394697</td>
</tr>
<tr>
<td>English</td>
<td></td>
<td>36770830</td>
<td>314372</td>
</tr>
<tr>
<td>Test</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Farsi</td>
<td>200</td>
<td>4524</td>
<td>314372</td>
</tr>
<tr>
<td>English</td>
<td></td>
<td>4431</td>
<td></td>
</tr>
</tbody>
</table>

### 5.3. Results

In our experiments, at first we evaluate each of the search methods discussed in section 4, to compare their performance, and to study the effect of each improvements done to the baseline. For the weighted edit distance method, the weight of the edit distance error is tuned on a development set and is finally set to -0.5. Also for the jump operation error, three different cases are considered; number of words we jumped over, number of phrases we jumped over and 1. The results of the KSMR and the average response time for each interaction are given in Table 6 for each of the methods.
Table 6. Evaluation results of different search methods

<table>
<thead>
<tr>
<th>Searching Method</th>
<th>KSMR (%)</th>
<th>Average Response Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit Distance Method</td>
<td>31.78</td>
<td>0.21</td>
</tr>
<tr>
<td>Adding Jump to Edit Distance Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jump Error = # of Words</td>
<td>31.62</td>
<td>0.93</td>
</tr>
<tr>
<td>Jump Error = # of Phrases</td>
<td>31.23</td>
<td>1.09</td>
</tr>
<tr>
<td>Jump Error = 1</td>
<td>30.43</td>
<td>1.67</td>
</tr>
<tr>
<td>Weighted Edit Distance Method</td>
<td>31.36</td>
<td>0.27</td>
</tr>
<tr>
<td>Adding Jump to Weighted Edit Distance Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jump Error = # of Words</td>
<td>30.31</td>
<td>1.1</td>
</tr>
<tr>
<td>Jump Error = # of Phrases</td>
<td>30.2</td>
<td>1.42</td>
</tr>
<tr>
<td>Jump Error = 1</td>
<td>30.15</td>
<td>1.65</td>
</tr>
</tbody>
</table>

As the results show, using the weighted sum of edit distance error and translation scores makes 0.42% improvements in KSMR and no significant growth in response time, compared to the base edit distance method.

Also, adding the jump operation to the edit distance method shows that it could improve its KSMR up to 1.35%, when the jump error is set to one regardless of the jump length. The results of the other settings for jump error are not as good as considering it to be one, which shows assuming it to be the number of words or phrases that we jumped over is a high error, which prevents some long jumps that might be useful.

Furthermore, when we combine weighted edit distance method with jump operation, we get 1.21% improvements in KSMR, in comparison to the weighted method itself, and 1.63% improvements compared to the base edit distance method. This shows that these two improving methods do not have much overlap in the cases that they improve the baseline system.

Despite the improvements in KSMR, the response time while we add the jump operation increases significantly. This increase can be expected as the number of backpointers created in the search process grows with adding jump operation, and the search space becomes much greater. For solving this issue we limit the jump operation more, by putting a threshold for the maximum amount of phrases (edges in the graph) that we could jump over. In the next experiment, the effect of this threshold is studied, and the results are shown in Table 7.

As the results in Table 7 shows, if we assume that the response time below 1 second is acceptable, then by selecting the threshold equal to 5, we can gain about 0.82% improvement in KSMR from the weighted edit distance method without jump and 1.24% improvement compared to the base edit distance method.
### Table 7. Results of the effect of limiting the jump length

<table>
<thead>
<tr>
<th>Searching Method</th>
<th>Maximum Jump Length</th>
<th>KSMR (%)</th>
<th>Average Response Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Jump to Weighted Edit Distance Method</td>
<td>1</td>
<td>31.16</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>30.92</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>30.71</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>30.64</td>
<td>0.93</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>30.54</td>
<td>0.96</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>30.41</td>
<td>1.04</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>30.27</td>
<td>1.13</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>30.23</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>∞</td>
<td>30.15</td>
<td>1.65</td>
</tr>
</tbody>
</table>

### 6. Conclusions and Future Work

In this paper we aimed to improve a common search method used for the interactive prediction in the computer assisted translation systems, which searches for a path in the search graph, whose prefix has least edit distance with the user’s confirmed prefix.

As the reordering of words which is desired by the user might not be in the search graph, the simple edit distance method has the problem of not considering the differences in reorderings between the system’s translation and the user’s prefix. In order to solve this problem, we propose adding a jump operation to the edit distance, which allows jumping over a contiguous sequence of phrases in a system’s translation while matching the user’s prefix with that translation, and then add the part that it jumped over to the continuation offer of the system.

The results show that adding jump to the edit distance improved the KSMR measure by 1.35%. Also, we show that this method does not have much overlap with another method which has been proposed for this purpose and used the weighted sum of edit distance and translation scores to select the best offer. Combining these two methods improved the base edit distance method by 1.63% in KSMR.

However, adding jump operation increases the average response time of the system in each interaction. This is because of the significant increase in the number of backpointers generated in the search process. We limit the length of the jumps to reduce the response time. Instead, the mechanisms could be presented to prune the inappropriate backpointers, and prevent them from further processing, so that we could increase the response speed with less reduction in KSMR.
References


Improving Semantic SMT via Soft Semantic Role Label Constraints on ITG Alignments

Meriem BELOUCIF  
mbeloucif@cs.ust.hk
Markus SAERS  
masaers@cs.ust.hk
Dekai WU  
dehai@cs.ust.hk
Human Language Technology Center  
Hong Kong University of Science and Technology, Hong Kong

Abstract

We show that applying semantic role label constraints to bracketing ITG alignment to train MT systems improves the quality of MT output in comparison to the conventional BITG and GIZA alignments. Moreover, we show that applying soft constraints to SRL-constrained BITG alignment leads to a better translation system compared to using hard constraints which appear too harsh to produce meaningful biparses. We leverage previous work demonstrating that BITG alignments are able to fully cover cross-lingual semantic frame alternations, by using semantic role labeling to further narrow BITG constraints, in a soft fashion that avoids losing relevant portions of the search space. SRL-based evaluation metrics like MEANT have shown that tuning towards preserving the shallow semantic structure across translations, robustly improves translation performance. Our approach brings the same intuition into the training phase. We show that our new alignment outperforms both conventional Moses and BITG alignment baselines in terms of the adequacy-oriented MEANT scores, while still producing comparable results in terms of edit distance metrics.

1 Introduction

The quality of machine translation output relies heavily on word alignment. However, the most widespread approach to word alignment is the ad hoc method of training IBM models (Brown et al., 1990) in both directions and combining their results using various heuristics. Word alignments based on inversion transduction grammars or ITGs (Wu, 1997), on the other hand, provide a more structured model leading to efficient and optimal bidirectional alignments.

In this paper we introduce an improved word aligner based on applying soft semantic role label constraints to ITG alignment. We show that both translation adequacy and fluency can be improved by replacing the conventional GIZA++ based alignment (Och and Ney, 2000) with more semantically motivated alignments obtained through training...
ITGs (Saers and Wu, 2009) under soft SRL constraints. The new approach is motivated by Addanki et al. (2012) who demonstrated empirically that the semantic role reorderings found in cross-lingual SRL frames is essentially 100% covered by ITG constraints, which suggests that it should be possible to use ITG constraints as a starting point under which to align semantic frames as we do in this paper.

Our approach is further motivated by the fact that including semantic role labeling in the SMT pipeline in a different way has already been shown to increase translation quality. The semantic frame based evaluation metric MEANT, which was shown to correlate better with human adequacy judgment than conventional surface based evaluation metrics (Lo et al., 2012), can be used as an objective function for tuning SMT. Tuning to MEANT, which attempts to optimize the degree to which a sentence’s semantic frames can be preserved across translation, was shown to improve translation quality across many metrics (Lo et al., 2013b; Beloucif et al., 2014). We show in this paper that including soft constraints based on semantic role labeling into the alignment training step yields both higher adequacy-oriented MEANT and, while still producing comparable scores on surface based and edit distance metrics.

2 Related work

2.1 Alignment

For most recent automatic machine translation systems, learning a good word alignment is paramount for producing meaningful translation. Unfortunately, conventional alignment algorithms such as IBM models (Brown et al., 1990) and the HMM-alignment model (Vogel et al., 1996) are flat and directed, meaning that (a) they allow unstructured movement of words leading to weak word alignment, (b) translations in one direction are considered in isolation, and (c) two separate alignments are needed to form a single bidirectional alignment. The harmonization of two directed alignments is typically done heuristically, which means that there is no model that considers the final bidirectional alignment that the translation system is trained on to be optimal. Transduction grammars, on the other hand, do provide a model that (a) is inherently structurally compositional, and (b) can provide optimal bidirectional alignments. Although this structured optimality comes at a higher cost in terms of time complexity, it allows for preexisting structured information to be incorporated into the model, and for models to be compared in a meaningful way.

There are different classes of transduction grammars, ranging from finite-state transduction grammar, via linear transduction grammar (Saers et al., 2010) and inversion transduction grammar (Wu, 1997; Saers and Wu, 2009; Saers et al., 2009), to syntax-directed transduction grammar (Lewis and Stearns, 1968; Aho and Ullman, 1972) and many ways to formulate the model over them: Wu (1995); Zhang and Gildea (2005); Chiang (2007); Cherry and Lin (2007); Blunsom et al. (2009); Haghighi et al.
In this paper, we introduce a semantically biased version of inversion transduction grammars (Wu, 1997) that is biased towards constituents that conform to monolingual semantic parses on the input and/or output languages, and compare their performance against (a) ITGs without such a bias, and (b) the conventional heuristics.

2.2 Semantic role labeling in MT

Our alignment method is fully compatible with the principle that a good translation is one where a human can successfully understand the main meaning of the output sentence as captured by the basic event structure: “who did what to whom, when, where and why” (Pradhan et al., 2004; Lo and Wu, 2011, 2012; Lo et al., 2012). The MEANT family of metrics are semantic evaluation MT evaluation metrics that correlate with human adequacy judgements more closely than most commonly used surface based metrics (Lo and Wu, 2011, 2012; Lo et al., 2012; Lo and Wu, 2013b; Macháček and Bojar, 2013). MEANT compares the MT output sentence against provided reference translations, and produce a score measuring the degree of similarity between their semantic frame structures. Our new approach is encouraged by the fact that many previous studies have empirically shown that integrating semantic role labeling into the training pipeline by tuning against MEANT improves the translation adequacy (Lo et al., 2013a; Lo and Wu, 2013a; Lo et al., 2013b; Beloucif et al., 2014). We show here, that soft incorporation of SRL constraints much earlier in the pipeline, at the word alignment stage of SMT training, can further improve translation adequacy.

2.3 Inversion transduction grammars

A transduction represents a set of bi-sentences that define the relation between an input language $L_0$ and an output language $L_1$. Accordingly, a transduction grammar generates a transduction or a set of bi-sentences, translates between sentences in $L_0$ and sentences in $L_1$, and accepts the sentence pairs in the transduction. Inversion transductions are a subset of syntax-directed transductions which are generated and parsed by inversion transduction grammars or ITGs (Wu, 1997). An ITG can always be written in 2-normal form and is represented by a tuple $\langle N, V_0, V_1, R, S \rangle$ where $N$ is a set of non-terminals, $V_0$ and $V_1$ are the vocabularies of $L_0$ and $L_1$ respectively, $R$ is a set of transduction rules and $S \in N$ is the start symbol. In the 2-normal form, each inversion transduction must be on one of the following forms:

\[
S \rightarrow A \\
A \rightarrow [BC] \\
A \rightarrow \langle BC \rangle \\
A \rightarrow e/\epsilon \\
A \rightarrow \epsilon/\emptyset
\]
Input: 这个在日本还没有出售。
Gloss: this one in Japan yet haven't sell.
Reference: They do not sell this in Japan, yet.
GIZA++: this in Japan yet on sale.
ITG: this in Japan has not on sale.
ITG with hard SRL constraints: this is not sold in Japan yet.
ITG with soft SRL constraints: this is not sold in Japan yet.

Figure 1: MT output of the three different systems for a given Chinese sentence

A → e/f

ITGs allow straight and inverted rules such that straight transduction rules use square brackets and take the form \( A \rightarrow [BC] \) and inverted rules use inverted brackets and take the form \( A \rightarrow ⟨BC⟩ \). Straight transduction rules generate transductions with the same order in \( L_0 \) and \( L_1 \) which means that, in the parse tree, the children instantiated by straight rules are read in the same order. Inverted transduction rules on the other hand, generate transductions with inverted order in \( L_0 \) and \( L_1 \), so the children instantiated by inverted rules are read left-to-right in \( L_0 \) and right-to-left in \( L_1 \). In this paper we show that an ITG-based SMT system is able to perform better on semantic metrics when biased towards respecting semantic role labeling.
please
tell
me
the
fastest
way
to
go
to
San
Francisco

please
tell
me
the
fastest
way
to
go
to
San
Francisco

Figure 3: An alignment of a bisentence produced by ITG alignment

Figure 4: An alignment of a bisentence produced by ITG alignment using automatic shallow semantic parsing constraints. The input is parsed by a Chinese automatic shallow semantic parser.

3 SRL-constrained ITGs

The model we propose in this paper introduces soft constraints to bias a bracketing inversion transduction grammar (BITG) towards preferring bilingual constituents that conform to automatically generated monolingual semantic role labels on both sides. Because of the structural differences between monolingual SRLs and the bilingual BITG parses, we implement this as a penalty for BITG constituents violating the semantic role labels.
The semantic roles and their fillers in a sentence sometimes span multiple syntactic units, or in technical terms: the semantic trees are (a) not necessarily consistent with the syntactic trees, and (b) not necessarily projective. Since BITG trees are defined to be projective, applying even a single monolingual SRL parse as a hard constraint would rule out all possible BITG trees, and all possible alignments for that sentence pair, since no BITG parse can conform to a non-projective constraint. Even when the monolingual SRL trees are projective in both languages, there is a risk of overly constraining the search, as the only way for the BITG parser to satisfy incompatible SRL constraints is to sacrifice the lexical correspondences; the only way to conform to (a) the input language SRL, (b) the output language SRL, and (c) the ITG constraint may be to delete a constituent in the input language and insert it in the output language, even when there is a good translation between them, because translating them would violate at least one of the two constraints. The ITG constraint is what allows us to do this processing in polynomial time, so it is non-negotiable. As the lexical relation is what defines the word alignment, which is what we are interested in, we opt to soften the SRL constraints. In practice, the automatic SRL parses are fairly noisy, an engineering reason to soften them, but even with perfect SRL parsers, soft constraints are theoretically necessary.

The soft constraints takes the form of a fixed penalty that is paid whenever the BITG parser wants to introduce a bi-constituent that crosses a semantic constituent (the string a predicate or one of its role fillers span). No penalty is paid for bi-constituents completely covering a semantic constituent or by bi-constituents that are completely covered by semantic constituents. To allow for some degrees of freedom, we allows for two separate penalties, one for crossing an input language semantic constituent, $\lambda_1$, and one for crossing an output language semantic constituent, $\lambda_0$. These hyper parameters need to be set manually.

4 Experiment Setup

4.1 Data

For this paper, we tested our systems on Chinese to English translations. We used IWSLT 2007 data set for this experiment. The training set contains 39,953 sentences. The dev and test set contain 1512 sentences and test 489 sentences, respectively. Both the English and Chinese corpora were normalized for punctuation, tokenized and true-cased. We also used our own Chinese named entity recognition, and dedicated proper name translator.
4.2 Word alignment

We compare the performance of our SRL soft-constrained model to the SRL hard-constrained system and to the conventional unconstrained ITG baseline. We perform a grid search over soft-constraints hyper parameters to find the optimal settings. We then compared the performance of our proposed alignments to the conventional GIZA++ baseline with grow-diag-final-and to harmonize the two alignment directions.

Our ITG baseline is a token-based BITG system. We initialize it with uniform structural probabilities, setting aside half of the probability mass for lexical rules. This probability mass is distributed among the lexical rules according to co-occurrence counts from the training data, assuming each sentence to contain one empty token to account for singletons. These initial probabilities are refined with 10 iterations of expectation maximization where the expectation step is calculated using beam pruned parsing (Saers et al., 2009) with a beam width of 100. On the last iteration, we extract the alignments imposed by the Viterbi parses as the word alignments outputted by the system.

The new SRL-constrained ITG approach adds the crossing penalty based on automatic SRL parses discussed in Section 3 to the ITG baseline discussed above. The shallow semantic parses of the training data were produced using ASSERT (Pradhan et al., 2004) and C-ASSERT (Wu et al., 2006) for English and Chinese respectively.

We show how applying soft SRL constrained ITG alignment outperforms alignment both (a) without constraints and (b) with hard SRL constraints. The hyper parameter $\lambda_i = 0$ represents the hard constraints, $\lambda_i = 1$ represents the case with no constraints and $\lambda_i$ between 0 and 1 are the soft constraints. We run some prior experiments and observed that applying hard SRL constraints did not lead to any alignment at all: the constraints were too harsh and did not permit any biparses. Soft SRL constrained ITGs, on the other hand, outperformed the unbiased BITG model in term of both adequacy-oriented MEANT scores. We noticed that $\lambda_0 = 1$ and $\lambda_1 = 0.5$ correspond to the best combination. The SRL constraints were only used during training of the probabilities of the ITG, and not when extracting the Viterbi parses and the corresponding word alignments.

4.3 SMT pipeline

To test the different alignments described in this paper, we used the standard Moses toolkit (Koehn et al., 2007), with a 6-gram language model learned with the SRI language model toolkit (Stolcke, 2002), to train our baselines. We tested our approach using Moses hierarchical models. For tuning, we used ZMERT (Zaidan, 2009), a standard implementation of minimum error rate training or MERT (Och, 2003), we run each tuning task ten times for each system, then we decoded with both DEV and TEST set, we then chose the results according to what performed the best on the know develop-
ment data. We also present the highest score achieved by each system among all runs (an upper bound on the score that can be achieved). We compared an edit-distance based metrics, BLEU (Papineni et al., 2002) and TER (Snover et al., 2006), and a semantic evaluation metric MEANT (Lo et al., 2012) as the tuning objective.

5 Results

We compared the the performance of our soft SRL-constrained ITG alignment to (a) the GIZA++ baseline and (b) the unbiased BITG, for both BLEU, TER and MEANT tuned systems. We evaluated our MT output using the semantic metric MEANT (Lo et al., 2012). Tables 1, 2 and 3 show the improvment in terms of MEANT scores for the SRL ITG aligned system in comparison to conventional ITG alignment and GIZA++ alignment for BLEU, TER and MEANT tuned systems respectively. The MEANT score for ITG based systems is considerably higher than the MEANT score for GIZA++ aligned model. We also observe that MEANT score for ITG with SRL constraints is better than the conventional ITG model. We believe that a better SRL-parser would yield a better system still. Tables 4, 5 and 6 give an upper bound on the results for the ten runs, we also see here that new semantically biased ITG model outperforms the baseline and ITG based alignment.

In addition, we evaluated the performance of our model using surface-based metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), CDER (Leusch et al., 2006), WER (Nießen et al., 2000), and TER (Snover et al., 2006), and observed that both the unbiased ITG and semantically biased ITG based systems yield comparable results that are high in comparison to conventional GIZA++ alignment.

Figure 1 shows an interesting example extracted from the test data. The Chinese input sentence has been pre-segmented into eight word-like units, and a word-for-word gloss reads approximately “this one in Japan yet haven’t sell.” The translator who produce the reference translation took some liberties and introduced the actor they, who no doubt was present in the context, but is not needed when the sentence is considered in isolation. The conventional GIZA++ system fails completely, not only is the translation bad English, but the meaning it conveys is the polar opposite of the original sentence due to a dropped negation. Although these kind of errors are disastrous for the consumer of the translations, they are common with surface based systems, and not likely to be addressed with surface based tuning objectives. The unbiased ITG system produces a sentence that is understandable but far from good, comparable to something a second-language learner would write early on. The produced sentence does, however, convey the correct meaning, although some nuance is missing due to the dropped yet. The ITG system with hard SRL constraints outputs nothing, because the constraints prevented it from learning from most of the training examples. The ITG system with soft SRL constraints produces a sentence that conveys the correct meaning in good English. Considered in isolation, the produced sentence is even better than the man-made
Table 1: The optimal results given the known development set for the BLEU tuned systems

<table>
<thead>
<tr>
<th>System</th>
<th>MEANT</th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
<th>CDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giza alignment</td>
<td>49.42</td>
<td>25.07</td>
<td>0.451</td>
<td>59.95</td>
<td>60.96</td>
<td>54.91</td>
<td>59.32</td>
</tr>
<tr>
<td>ITG alignment</td>
<td>50.02</td>
<td>24.13</td>
<td>0.460</td>
<td><strong>8.49</strong></td>
<td>59.55</td>
<td>53.98</td>
<td><strong>58.01</strong></td>
</tr>
<tr>
<td>SRL ITG alignment</td>
<td><strong>50.92</strong></td>
<td>23.67</td>
<td><strong>0.460</strong></td>
<td>58.78</td>
<td>59.79</td>
<td><strong>53.61</strong></td>
<td>58.25</td>
</tr>
</tbody>
</table>

Table 2: The optimal results given the known development set for the TER tuned systems

<table>
<thead>
<tr>
<th>System</th>
<th>MEANT</th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
<th>CDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giza alignment</td>
<td>48.70</td>
<td><strong>23.02</strong></td>
<td>0.407</td>
<td>59.93</td>
<td>60.83</td>
<td>55.76</td>
<td>59.55</td>
</tr>
<tr>
<td>ITG alignment</td>
<td>49.76</td>
<td>21.72</td>
<td>0.431</td>
<td><strong>57.83</strong></td>
<td>58.76</td>
<td>53.90</td>
<td><strong>57.38</strong></td>
</tr>
<tr>
<td>SRL ITG alignment</td>
<td><strong>50.02</strong></td>
<td>21.92</td>
<td><strong>0.433</strong></td>
<td>58.78</td>
<td>59.55</td>
<td><strong>53.34</strong></td>
<td>58.01</td>
</tr>
</tbody>
</table>

Table 3: The optimal results given the known development set for the MEANT tuned systems

<table>
<thead>
<tr>
<th>System</th>
<th>MEANT</th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
<th>CDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giza alignment</td>
<td>48.88</td>
<td>21.18</td>
<td>0.455</td>
<td>61.49</td>
<td>62.90</td>
<td>55.76</td>
<td>60.91</td>
</tr>
<tr>
<td>ITG alignment</td>
<td>49.12</td>
<td><strong>23.47</strong></td>
<td><strong>0.465</strong></td>
<td>60.30</td>
<td>61.20</td>
<td>55.36</td>
<td>59.53</td>
</tr>
<tr>
<td>SRL ITG alignment</td>
<td><strong>50.66</strong></td>
<td>22.83</td>
<td>0.446</td>
<td><strong>59.48</strong></td>
<td><strong>60.32</strong></td>
<td><strong>54.30</strong></td>
<td><strong>58.84</strong></td>
</tr>
</tbody>
</table>

reference translation, as it is closer to the original.

Figures 2-4 shows a sentence pair from the training data, and how the different systems end up aligning the words. The pre-segmented Chinese words correspond to please, tell, I go, San Francisco, most, fast, of, way, and full stop. The harmonized GIZA++ alignments (Figure 2) are good, except that to go to San Francisco and the fastest are grouped into atomic units, which means that this is not an example of go, San Francisco, or fastest being translated. The unconstrained ITG alignment (Figure 3) makes two mistakes: it insists on aligning 我去’I go’ with me instead of go, and it aligns the superlative marker 最 with Francisco. The SRL constraints (Figure 4) are able to fix the latter, but not the former.

6 Conclusion

In this paper we showed that incorporating SRL constraints into the training of bracketing ITGs for early stage word alignment in SMT training leads to improved semantic
Table 4: The upper bound score among the eleven runs for BLEU tuned systems for each pipeline.

<table>
<thead>
<tr>
<th>System</th>
<th>MEANT</th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
<th>CDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giza alignment</td>
<td>50.91</td>
<td>25.07</td>
<td>0.453</td>
<td>59.53</td>
<td>60.42</td>
<td>54.43</td>
<td>59.02</td>
</tr>
<tr>
<td>ITG alignment</td>
<td>50.75</td>
<td>24.41</td>
<td>0.469</td>
<td><strong>58.47</strong></td>
<td>59.55</td>
<td>53.90</td>
<td><strong>57.80</strong></td>
</tr>
<tr>
<td>SRL ITG alignment</td>
<td><strong>51.28</strong></td>
<td>24.67</td>
<td><strong>0.556</strong></td>
<td>58.78</td>
<td>59.69</td>
<td><strong>53.61</strong></td>
<td>58.25</td>
</tr>
</tbody>
</table>

Table 5: The upper bound score among the eleven runs for TER tuned systems for each pipeline.

<table>
<thead>
<tr>
<th>System</th>
<th>MEANT</th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
<th>CDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giza alignment</td>
<td>49.94</td>
<td>23.02</td>
<td>0.408</td>
<td>59.40</td>
<td>60.52</td>
<td>55.58</td>
<td>59.14</td>
</tr>
<tr>
<td>ITG alignment</td>
<td>50.75</td>
<td>22.11</td>
<td>0.434</td>
<td><strong>57.16</strong></td>
<td><strong>58.57</strong></td>
<td>53.55</td>
<td><strong>57.30</strong></td>
</tr>
<tr>
<td>SRL ITG alignment</td>
<td><strong>50.92</strong></td>
<td>24.70</td>
<td><strong>0.441</strong></td>
<td>57.93</td>
<td>58.86</td>
<td><strong>53.34</strong></td>
<td>57.43</td>
</tr>
</tbody>
</table>

Table 6: The upper bound score among the eleven runs for MEANT tuned systems for each pipeline.

<table>
<thead>
<tr>
<th>System</th>
<th>MEANT</th>
<th>BLEU</th>
<th>METEOR</th>
<th>TER</th>
<th>WER</th>
<th>PER</th>
<th>CDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giza alignment</td>
<td>49.15</td>
<td>21.25</td>
<td>0.455</td>
<td>61.49</td>
<td>62.90</td>
<td>55.76</td>
<td>60.75</td>
</tr>
<tr>
<td>ITG alignment</td>
<td>49.94</td>
<td><strong>23.94</strong></td>
<td>0.466</td>
<td>60.30</td>
<td>61.20</td>
<td>55.36</td>
<td>59.53</td>
</tr>
<tr>
<td>SRL ITG alignment</td>
<td><strong>50.66</strong></td>
<td>22.83</td>
<td>0.457</td>
<td><strong>59.48</strong></td>
<td><strong>60.32</strong></td>
<td><strong>54.30</strong></td>
<td><strong>58.81</strong></td>
</tr>
</tbody>
</table>

translation adequacy. Moreover, we showed that applying soft SRL constraints in one of the languages produces better performance on the semantically oriented MEANT metric, in comparison to not applying any constraints. As automatically produced semantic parses for both languages are incompatible much of the time, we showed that the increased flexibility of soft constraints helps improve the word alignment quality. Finally, we observed that applying SRL constraints to BITG alignment using soft constraints not only improves MEANT scores but also retains the performance using surface oriented metrics like metrics like BLEU, METEOR, TER, WER, PER and CDER.

7 Acknowledgment

This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under BOLT contract no. HR0011-12-C-0016, and GALE contract nos. HR0011-06-C-0022 and HR0011-06-C-0023; by the European Union under the FP7 grant agreement no. 287658; and by the Hong Kong Research Grants Council (RGC) research grants GRF620811, GRF621008, and GRF612806. Any opinions,
findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, the EU, or RGC. We are grateful to Pascale Fung, Yongsheng Yang and Zhaojun Wu for sharing the maximum entropy Chinese segmenter and C-ASSERT, the Chinese semantic parser, with us.

References

Karteek Addanki, Chi-kiu Lo, Markus Saers, and Dekai Wu. LTG vs. ITG coverage of cross-lingual verb frame alternations. In EAMT-2012, Trento, Italy, May 2012.


Chi-kiu Lo and Dekai Wu. Can informal genres be better translated by tuning on automatic semantic metrics? In *MT Summit XIV*, 2013.


Vectorial representations of words have grown remarkably popular in natural language processing and machine translation. The recent surge in deep learning-inspired methods for producing distributed representations has been widely noted even outside these fields. Existing representations are typically trained on large monolingual corpora using context-based prediction models. In this paper, we propose extending pre-existing word representations by exploiting Wiktionary. This process results in a substantial extension of the original word vector representations, yielding a large multilingual dictionary of word embeddings. We believe that this resource can enable numerous monolingual and cross-lingual applications, as evidenced in a series of monolingual and cross-lingual semantic experiments that we have conducted.

1 Introduction

In this work, we focus on the task of creating vector representations of multilingual words (as well as lexicalized phrases). Previous work in this area has relied on multilingual corpora to train bilingual word vectors. We investigate to what extent external large-scale resources can be used to create much more multilingual word representation data. In particular, we rely on Wiktionary, a sister project of Wikipedia that for many years now has been creating a large, collaboratively edited online dictionary. Due to its rich multilingual data, now with over 4 million entries in over 1,000 languages, Wiktionary has been used extensively in natural language processing, e.g. for part-of-speech tagging (Li et al., 2012) and named entity recognition (Richman and Schone, 2008), for cross-language image search (Etzioni et al., 2007) and text classification (Nastase and Strapparava, 2013), and for producing language registries (de Melo, 2015) and etymological databases (de Melo, 2014). Wiktionary has also made it possible to translate lexical knowledge bases such as WordNet to hundreds of languages (de Melo and Weikum, 2009) and to translate thesauri (Borin et al., 2014). Finally, it has been used as an extra ingredient in regular machine translation systems (Göhring, 2014).

Relying on Wiktionary instead of on other training data has playfully been called Wikily supervision (Li et al., 2012). Our work constitutes a form of Wiktionary-based supervision.
for multilingual word representation learning. More specifically, our method starts with existing word representations such as the widely available ones trained on large English corpora (Mikolov et al., 2013a; Pennington et al., 2014). It then uses Wiktionary to decide how to place new words into the same vector space.

2 Method

For obtaining the new word representations, we adopt the following framework. We assume, we are given vectors $\mathbf{u}_w$ for words $w \in V_0$, where $V_0$ is some initial vocabulary of words. Such vectors may come from any of the popular methods for training word vectors. We later use the well-known vectors from the word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) projects. Our goal is to create new vectors $\mathbf{v}_w$ for all words $w$ in a substantially larger vocabulary $V$, which typically will contain words from many different languages.

Note that the words $w$ are tagged with language codes and are distinguished accordingly. For example, the Czech word *tuna* refers to a ton (the weight unit), and the Spanish word *tuna* means prickly pear/nopal. Neither of these bear any relationship with the fish meaning of the English word *tuna*. Thus, we treat words with different language tags as distinct entities with separate vectors. This, of course, does not preclude connections in the data from encouraging a high degree of proximity between different vectors. For example, the method will encourage the English word *sushi* to have similar vectors to those of the French and Breton *sushi*, which have the same form and meaning.

The vectors $\mathbf{v}_w$ should reflect the semantics of the words so as to be useful in downstream applications. While in the past, word vectors were often chosen such that individual dimensions have some interpretable meaning, current state-of-the-art vector space word embeddings do not have this property. Instead, we allow for words to be assigned arbitrary vectors as long as vector similarities and distances reflect corresponding word similarities and distances.

In order to achieve this, we draw on Wiktionary in order to obtain a large set $W$ of semantic triples taking the form

$$(w_1, r, w_2)$$

where $w_1, w_2$ are words, and $r$ is a relation that holds between them. The most frequent relation that we obtain from Wiktionary data is the translation relation. Other examples include synonymy and derivational relationships. Based on the triples in $W$, we then define the following objective:

$$\sum_{w_1} \sum_{w_2} f_W(w_1, w_2) \mathbf{v}_{w_1} \cdot \mathbf{v}_{w_2}$$

subject to

$$\|\mathbf{v}_w\|_2 \leq 1 \quad \forall w,$$

where $f_W(w_1, w_2)$ should quantify the connection strength (and polarity) between words. Thus, words are encouraged to have similarities that correspond to their relatedness, measured in terms of their dot products.

In practice, we maximize this objective function iteratively using stochastic gradient ascent. Initially, we set

$$\mathbf{v}_w = \begin{cases} 
\mathbf{u}_w & w \in V_0 \\
0 & \text{otherwise}
\end{cases} \quad (1)$$

We then repeatedly make local updates for individual triples in order to optimize the vectors in the direction of the objective. We use two different learning rates $\alpha_1, \alpha_2$ with $\alpha_1 \geq \alpha_2$ for this. The first one, $\alpha_1$, is the greater of the two and is the learning rate used for new words, whilst the second, $\alpha_2$, is the learning rate used for words that were already in $V_0$, i.e., the vocabulary
of the input word vectors. Since the original words have already been optimized in some prior learning process, this severely tempers the extent to which they may be negatively affected by noise, especially towards the beginning, when the vectors for the new words have not yet stabilized.

In our experiments, we simply use

$$f_W(w_1, w_2) = |\{t \in W \mid \exists r : t = (w_1, r, w_2)\}|$$

to quantify relation strengths. While this function is non-negative, the fact that we start off with existing high-quality word vectors, that we constrain L2 norms to not grow indefinitely, and that we choose slow learning rates allow us to end up with high-quality word vectors.

### 3 Wiktionary Parsing

The word relationship triples in $W$ are taken from Wiktionary. Unfortunately, Wiktionary’s data is created using a rather informal semi-structured wiki markup form that is difficult to parse and not very standardized at all. For example, Figure 1 shows just a small part of the page for the word *pulse*. We rely on a custom information extraction system to produce a conversion of Wiktionary to structured data, as required for $W$. This is a rule-based system that partitions the raw wiki markup into different parts looking for sections and other subdivisions. It extracts translations both from the translation sections and from the gloss text, as these are a rich resource as well. In the gloss text, we sometimes have short translations, and sometimes we may also find inflectional or derivational links, as, for instance, in Figure 2.

Table 1 provides details about the extracted data, obtained by applying our parser on an XML dump of the English edition of Wiktionary (2013-12-17 version). Note that the links counts refer to the total numbers of directed links after adding inverses and removing any duplicates. We see that Wiktionary provides several million translation links as well as significant numbers of other relationships, including inflectional and derivational ones.
Figure 2: Simpler Wiktionary example showing the page for the French word *déjeûna*, which is listed as an inflected form of *déjeûner* (to have lunch).

<table>
<thead>
<tr>
<th>Item</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Translational equivalence links</td>
<td>3,598,807</td>
</tr>
<tr>
<td>Derivational/Inflectional links</td>
<td>2,455,781</td>
</tr>
<tr>
<td>Related term links</td>
<td>580,631</td>
</tr>
<tr>
<td>Synonymy links</td>
<td>490,130</td>
</tr>
<tr>
<td>Orthographic/other variant links</td>
<td>17,357</td>
</tr>
<tr>
<td>Unique words</td>
<td>3,968,843</td>
</tr>
</tbody>
</table>

Table 1: Wiktionary input statistics, where link counts refer to directed links after adding inverses and removing duplicates.

4 Experiments

4.1 Training

For the original input vectors, we rely on two well-known sources. The first are the pretrained word2vec vectors (Mikolov et al., 2013a) released by Google\(^1\). This dataset provides vector representations for words and multi-word phrases trained on a Google News dataset consisting of about 100B word tokens using word2vec. The vocabulary size is 3,000,000. However, out of these 3,000,000, actually 2,070,978 terms contain a space, most of which are multi-word expressions or named entities. Thus, the number of genuine words is much smaller.

As a second vector dataset, we experiment with the pre-trained vectors from the GloVe project (Pennington et al., 2014), which they obtained by applying their algorithm to data from a CommonCrawl corpus consisting of 840B word tokens. The vocabulary size is 2,195,960, out of which none contain a space. While the corpus is larger, it should be noted that CommonCrawl contains a lot of rather noisy Web data.

We train our model using a starting learning rate of \(\alpha_1 = 0.1\) for new words and \(\alpha_2 = 0.001\) for original words. The vectors stabilize fairly quickly, so we run the algorithm for only 10 epochs.

4.2 Coverage

As a result of this training process using Wiktionary data, the original word2vec representations are modified from covering 3 million tokens in just a single language to covering nearly 6

\(^1\)https://code.google.com/p/word2vec/
million words in over 500 languages. Similarly, the GloVe vectors are extended from 2,195,960 word vectors in one language to around 5 million vectors, again in over 500 languages. In Table 2, we list the languages with the greatest coverage on the extended word2vec dataset.

Remarkably, even the coverage of English increases quite substantially, by over 200,000 entries. Although this number might appear small in comparison with the original vocabulary size of 3,000,000, there is a marked difference in quality between the two. Apart from the roughly 2 million multi-word expressions among these 3 million vocabulary items, the original data also contains vast amounts of tokens that are not genuine lexical items but simply various sorts of names, codes, misspellings, file names, and so on (e.g. Krakowiak, SBSA, www.flu.gov, reccomend, WILLOW). In contrast, the added vocabulary items are mostly genuine word forms, contributed by Wiktionary’s editors.

Table 3 summarizes the total number of languages covered by the vectors trained on Wiktionary. A lot of rare minority languages are covered to some extent. While the vocabulary size for them tends to be small, the coverage often focuses on the most important words, such as those found in Swadesh lists and of interest in linguistic and anthropological studies. 38 languages are covered with a vocabulary size of at least 10,000. For these languages, the coverage should suffice for certain NLP tasks, including cross-lingual ones. This is what we shall investigate next.

4.3 Semantic Relatedness

Semantic relatedness studies have a long history in computational lexical semantics. Given a set of word pairs and corresponding scores quantifying how strongly human assessors deem the two respective words in a word pair semantically related, the goal is automatically produce similar assessment scores. The evaluation is normally carried out in terms of correlation coefficients.
### Table 3: Number of languages.

<table>
<thead>
<tr>
<th></th>
<th>Count (word2vec)</th>
<th>Count (GloVe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of languages with ≥ 50000 words</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>No. of languages with ≥ 10000 words</td>
<td>38</td>
<td>38</td>
</tr>
<tr>
<td>No. of languages with ≥ 5000 words</td>
<td>62</td>
<td>62</td>
</tr>
<tr>
<td>No. of languages with ≥ 1000 words</td>
<td>123</td>
<td>123</td>
</tr>
<tr>
<td>No. of languages with ≥ 100 words</td>
<td>267</td>
<td>266</td>
</tr>
<tr>
<td>No. of languages with ≥ 10 words</td>
<td>360</td>
<td>360</td>
</tr>
</tbody>
</table>

### Table 4: German semantic relatedness results, evaluated in terms of Spearman’s rank correlation coefficient and coverage.

<table>
<thead>
<tr>
<th></th>
<th>Chandar A P et al. (2014) En-De Vectors</th>
<th>Ours (word2vec)</th>
<th>Ours (GloVe)</th>
<th>Chandar A P et al. (2014) En-De Vectors</th>
<th>Faruqui et al. (2015)</th>
<th>Ours (word2vec)</th>
<th>Ours (GloVe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UKP30</td>
<td></td>
<td>0.212 @ 34.5%</td>
<td>0.752 @ 96.6%</td>
<td></td>
<td>0.096 @ 26.2%</td>
<td>0.603 @ N/A</td>
<td>0.717 @ 96.9%</td>
</tr>
<tr>
<td>GUR65</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>−0.319 @ 26.2%</td>
<td>0.603 @ N/A</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>Faruqui et al. (2015)</td>
<td>0.603 @ N/A</td>
<td>0.717 @ 96.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GUR350</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>0.558 @ 51.7%</td>
<td>0.605 @ 68.3%</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>Faruqui et al. (2015)</td>
<td>0.605 @ 68.3%</td>
<td>0.768 @ 96.9%</td>
</tr>
<tr>
<td>ZG222</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>0.111 @ 38.3%</td>
<td>0.161 @ 54.1%</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>Faruqui et al. (2015)</td>
<td>0.161 @ 54.1%</td>
<td>0.306 @ 54.1%</td>
</tr>
</tbody>
</table>

### Table 5: Spanish semantic relatedness results, evaluated in terms of Spearman’s rank correlation coefficient and coverage.

<table>
<thead>
<tr>
<th></th>
<th>Chandar A P et al. (2014) En-Es Vectors</th>
<th>Ours (word2vec)</th>
<th>Ours (GloVe)</th>
<th>Chandar A P et al. (2014) En-Es Vectors</th>
<th>Faruqui et al. (2015)</th>
<th>Ours (word2vec)</th>
<th>Ours (GloVe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RG65</td>
<td>Chandar A P et al. (2014) En-Es Vectors</td>
<td>0.629 @ 55.4%</td>
<td>0.805 @ 100.0%</td>
<td>Chandar A P et al. (2014) En-Es Vectors</td>
<td>Faruqui et al. (2015)</td>
<td>0.805 @ 100.0%</td>
<td>0.844 @ 100.0%</td>
</tr>
<tr>
<td>MC30</td>
<td>Chandar A P et al. (2014) En-Es Vectors</td>
<td>0.430 @ 60.0%</td>
<td>0.591 @ N/A</td>
<td>Chandar A P et al. (2014) En-Es Vectors</td>
<td>Faruqui et al. (2015)</td>
<td>0.591 @ N/A</td>
<td>0.830 @ 76.7%</td>
</tr>
<tr>
<td>WS353</td>
<td>Chandar A P et al. (2014) En-Es Vectors</td>
<td>0.256 @ 65.1%</td>
<td>0.538 @ 65.6%</td>
<td>Chandar A P et al. (2014) En-Es Vectors</td>
<td>Faruqui et al. (2015)</td>
<td>0.538 @ 65.6%</td>
<td>0.596 @ 65.6%</td>
</tr>
</tbody>
</table>
Table 6: French semantic relatedness results, evaluated in terms of Spearman’s rank correlation coefficient and coverage.

<table>
<thead>
<tr>
<th>Language Pair</th>
<th>Method</th>
<th>Spearman’s Rank Correlation</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Fr RG65</td>
<td>Chandar et al. (2014) En-Fr Vectors</td>
<td>0.586 @ 49.2%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ours (word2vec)</td>
<td>0.822 @ 96.9%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ours (GloVe)</td>
<td>0.827 @ 96.9%</td>
<td></td>
</tr>
<tr>
<td>En-Fr RG65</td>
<td>Chandar et al. (2014) En-Fr Vectors</td>
<td>0.606 @ N/A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ours (word2vec)</td>
<td>0.864 @ 100.0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ours (GloVe)</td>
<td>0.863 @ 100.0%</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Cross-lingual semantic relatedness results

These quantify to what degree the word pairs turn out to be in a similar order when sorting with respect to the two kinds of relatedness scores – ground-truth human-provided ones vs. automatically generated ones.

Although semantic relatedness assessment is not an end-user task in itself, it is an important building block in numerous NLP systems. For instance, measures of semantic relatedness can be used in search query expansion, text classification, and schema and ontology matching, among many others.

Following Pennington et al. (2014), we use cosine similarity over normalized vectors. The word vectors we generate are case-sensitive, distinguishing Reading, which often refers to the city, from reading, which often refers to the process of reading. However, some of the datasets do not preserve case and so we also consider any possible capitalized version of the input word. Whenever at least one word has multiple candidate vectors, we take the maximum similarity over all pairs.

We evaluate this method using Spearman’s rank correlation coefficients over all covered
word pairs, while reporting the respective coverage percentage. In computing Spearman’s $\rho$, we follow the recommended procedure of using average ranks for tied positions.

While most semantic relatedness datasets focus on English, there are a few non-English ones as well, which we can use to evaluate our system. Unfortunately, their number is rather small, so we are limited in the number of languages that we can readily consider in this sort of evaluation. We use several publicly released datasets that were often based on pre-existing English datasets\(^2\). In Table 4 we provide evaluation results on German-language datasets, while Tables 5 and 6 provide similar results on Spanish and French datasets. For comparison, we list all published results known to us that are also based on vectors as well as results on all other non-English word vectors we could obtain and evaluate directly. In all cases, we see that our vectors fare significantly better than the competitors.

4.4 Cross-Lingual Semantic Relatedness

Semantic relatedness can also be evaluated across languages. We adopt the same methodology as earlier but rely on the Spanish-English evaluation data from Hassan and Mihalcea (2009), which we can use to compare our vectors with those of Chandar A P et al. (2014). Further, we consider the new cross-lingual semantic relatedness evaluation data released by Camacho-Collados et al. (2015), which is based on the RG65 dataset. Our results on these cross-lingual datasets are listed in Table 7. Again, our Wiktionary-based representations compare favorably with other available results.

4.5 Word Choice Problems

Word choice problems consist of a target word and a selection of possible words or phrases describing it. Consider the following three examples.

<table>
<thead>
<tr>
<th>gourmet</th>
<th>dale</th>
<th>brace</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) enjoys cooking</td>
<td>a) plain</td>
<td>a) to scream</td>
</tr>
<tr>
<td>b) has indigestion</td>
<td>b) retreat</td>
<td>b) prepare for danger</td>
</tr>
<tr>
<td>c) has an expert appreciation of food</td>
<td>c) shelter</td>
<td>c) hold your breath</td>
</tr>
<tr>
<td>d) is hungry</td>
<td>d) valley</td>
<td>d) close your eyes</td>
</tr>
</tbody>
</table>

Here, the correct answers are c) for gourmet, d) for dale, and b) for brace. For English, we rely on a well-known dataset used by Mohammad et al. (2007). We also use a large German-language collection of similar quiz questions\(^3\). The latter consists of 984 problem instances collected from 2001 to 2005 editions of the German version of Reader’s Digest Magazine, where they appear as “Word Power” problems.

We compute cosine similarities between the target word and the candidate answers. Some answers are individual words or expressions already covered in our data, in which case this is simple. If a candidate answer, however, consists of multiple words that are not covered in our data as a multi-word expression, we simply use the maximum cosine similarity between any of the words in the answer phrase and the target word.

We assess the accuracy as the sum of scores over all problem instances divided by the number of problem instances. Following the convention from previous work (Mohammad et al., 2007), the score is 1 if the correct answer is ranked highest among the candidates, 0 if it is not.

\(^2\)For more information on these datasets, please refer to https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-relatedness-datasets/ as well as Hassan and Mihalcea (2009) and Camacho-Collados et al. (2015).

\(^3\)https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-word-choice-problems/
### Table 8: Accuracy results on English and German word choice problems

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Vectors</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>27.42%</td>
</tr>
<tr>
<td></td>
<td>Original word2vec input vectors</td>
<td>65.31%</td>
</tr>
<tr>
<td></td>
<td>Ours (word2vec)</td>
<td>76.49%</td>
</tr>
<tr>
<td></td>
<td>Original GloVe input vectors</td>
<td>68.54%</td>
</tr>
<tr>
<td></td>
<td>Ours (GloVe)</td>
<td>74.42%</td>
</tr>
<tr>
<td>German</td>
<td>Chandar A P et al. (2014) En-De Vectors</td>
<td>27.35%</td>
</tr>
<tr>
<td></td>
<td>Ours (word2vec)</td>
<td>40.91%</td>
</tr>
<tr>
<td></td>
<td>Ours (GloVe)</td>
<td>40.85%</td>
</tr>
</tbody>
</table>

ranked highest, and \( \frac{1}{n} \) if our method’s top ranked answers form a tie of \( n \) answers with the same similarity score.

The results are provided in Table 8. Although our method for handling phrases is very simplistic, we obtain reasonably good results. Somewhat surprisingly, we quite significantly improve over the original input vectors for English. The contribution could come from the English lexical information in Wiktionary as well as from the cross-lingual relationships extracted from Wiktionary.

For German, the results are not as good as for English, which, however, is mainly due to the morphological complexity of the phrases in German. Better results could easily be obtained by improving the linguistic analysis of candidate answers, for instance by performing lemmatization, stop word removal or interpretation, and compound splitting, which, of course, is particularly helpful for German with its notoriously long compound nouns. After that, one could then use our vectors to obtain more reliable similarity scores.

#### 4.6 Word and Entity Analogies

Mikolov et al. (2013c) showed that distributed word vectors trained on large corpora using prediction approaches may exhibit intriguing semantic and linguistic regularities, even if these are not in any way directly part of their training objective. For instance, in their results, the vectors for *king* and *queen* stand roughly in the same relationship to each other as the vectors for *uncle* and *aunt*, or *man* and *woman*. This works to the extent that simple vector arithmetic often produces a vector whose nearest known word vector is the correct answer.

In order to create a non-English analogy dataset, we took the semantic analogies dataset of Mikolov et al. (2013c) and filtered out the parts focusing on geography, as these are to a large extent language-independent names like *Portland* or *Alaska*. This left us with the family and male/female related analogies. From these, we randomly selected 50 examples and created the corresponding French-language analogies. When multiple different translations appeared reasonable, we first restricted the choice based on the register (*maman* for *mom* but *mère* for *mother*) and then used the most popular form in the few cases where more than one option remained, e.g. *belle-mère* for *stepmother* rather than *marâtre*, which typically has the connotation of implying an evil stepmother.

For each analogy entry, the first two words demonstrate the analogy, and for the second pair of words, only the first is given as input to the system. The goal is predict the second one. We follow Mikolov et al. (2013c) in computing the target vector using simple vector arithmetic. We then find words near that target vector by choosing the nearest neighbors in terms of the Euclidean distance, considering only French words so as to obtain an answer in the correct target language. On our dataset of 50 French analogies, we obtain the results shown in
Table 9. Note that the French dataset is somewhat more challenging than the English original, because some translations are polysemous and no longer retain the sense distinctions of the English originals. For instance, both *girl* and *daughter* correspond to *fille* in French. While the approach by Chandar A P et al. (2014) does extremely poorly, our vectors achieve reasonable results. In those cases where they return the wrong answer, the correct one is often among the top 3.

5 Background and Related Work

Distributed representations in neural networks go back to at least Rumelhart et al. (1986), who described, for their well-known family tree case study, how weights distributed across different input units can be used to describe people. Importantly, their representation allowed two different people from separate families to share most weights if their other attributes were similar.

The lineage of the distinct idea of *distributional semantics* can be traced to use theories of linguistic meaning, which, roughly speaking, hold that language use in context determines the meaning of a word. This view fits well with the idea of empiricist corpus linguistics and the computational goals of discerning aspects of meaning using data-driven methods. Thus, distributional methods have received considerable attention in natural language processing (Schütze, 1993). Over time, it became apparent that one of the challenges with many distributional methods is the sparsity of observed word co-occurrences in a corpus in comparison with the overall distribution of likely word co-occurrences. Since many distributional approaches use numerical vectors to represent the contexts, this sparsity often is manifested in the form of sparse vectors with many zeros. Smoothing techniques and algorithms such as Latent Semantic Analysis (Deerwester et al., 1990) were proposed to alleviate some of these problems.

More recently, distributed and distributional methods have grown together in the form of neural network-inspired architectures that learn distributed representations from large corpora by accounting for word co-occurrences (Collobert et al., 2011; Turian et al., 2010). The resulting representations are still vectorial and based on corpus co-occurrences, but much lower-dimensional than in traditional distributional approaches and thus significantly less sparse. The massive attention on deep learning in recent years, paired with fast training methods as in the word2vec method by Mikolov et al. (2013a), which actually forgoes deep learning, has propelled these methods to the forefront of NLP, to the point that they are known well beyond the core natural language processing community.

Subsequently, a number of improvements to the learning algorithms have been proposed. Our objective function is related to those of other models that aim to exploit similarities between words. Chen et al. (2015) extend the word2vec CBOW objective function in order to pay special attention to contexts that reveal more explicit semantic relationships, rather than treating all contexts as equal. The semantic relationships are obtained using information extraction methods, e.g. from lists and definitions. Yu and Dredze (2014) and Faruqui et al. (2015) propose to optimize monolingual word embeddings so as to match information from pre-existing lexical resources. Hill et al. (2015) used dictionary glosses from several resources (including Wiktionary) in order to train neural networks to produce vectors for multi-word phrases.

<table>
<thead>
<tr>
<th>Vectors</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chandar A P et al. (2014) En-Fr Vectors</td>
<td>2.0%</td>
</tr>
<tr>
<td>Ours (word2vec)</td>
<td>30.0%</td>
</tr>
<tr>
<td>Ours (GloVe)</td>
<td>35.0%</td>
</tr>
</tbody>
</table>
While most word representation learning research has been monolingual aiming at English, recently there has been some interest in multilingual aspects of it.

Some works take pre-existing vectors for different languages and connect them. Mikolov et al. (2013b) develop a method to learn projections between two monolingual word embedding spaces. Lazaridou et al. (2015) investigate means to improve such projections. Faruqui and Dyer (2014) propose using canonical correlation analysis (CCA), while Lu et al. (2015) suggest using Deep CCA instead. Our method, in contrast, does not assume that we have already created non-English word vectors. We only rely on English word vectors, which are readily available from numerous sources.

A number of studies have focused on using multilingual corpora, often parallel corpora, to produce bilingual word vector spaces (Kalchbrenner and Blunsom, 2013; Kočiský et al., 2014). Utt and Padó (2014) investigate using syntax for bilingual vector space models. Hermann and Blunsom (2014) create bilingual word representations without word alignment. Hill et al. (2014) showed that word embeddings obtained from translations better reflect the ontological status of words than regular neural embeddings. One advantage of corpus-based approaches is the potential to have a substantial coverage, given sufficiently large corpora. However, as the amount of available parallel text is somewhat limited, in practice, this advantage may only apply to methods that do not require parallel corpora. Our work is complementary to this line of research on corpus-driven approaches. We exploit the availability of high-quality word vectors for English trained on very large Web-scale data, leading to word vector spaces that reflect word analogies well. We further draw on the availability of multilingual lexical resources such as Wiktionary, covering hundreds of languages, including lesser-resourced ones, for which corpora may be difficult to obtain.

More generally, our work differs from previous work by going beyond bilingual vector spaces in order to place millions of word forms from different languages into a single shared vector space.

6 Conclusion

We have presented the first study to produce large amounts of word vectors from Wiktionary in many languages. Unlike previous work on bilingual word embedding spaces, our approach produces a single significantly multilingual word vector space rather than just bilingual ones. Our experiments show that our vectors reflect semantic properties and that they are useful both in monolingual and in cross-lingual settings.

In the future, we would like to investigate the potential of these multilingual vectors for machine translation of text. While deep recurrent neural network architectures have recently achieved state-of-the-art results in several machine translation settings, they still suffer from significant problems with out-of-vocabulary items (Sutskever et al., 2014; Luong et al., 2015). Rather than only addressing these with custom ad hoc techniques as in the approach taken by Luong et al. (2015), it would be helpful to investigate to what extent we can incorporate them within the same vector space.

References


