An Improvement to the Predicate-Argument Structure Based Pre-ordering Approach for Statistical Machine Translation

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai Seika-cho, Soraku-gun Kyoto 619-0237 Japan
{wu.xianchao,sudoh.katsuhito,kevin.duh,tsukada.hajime,nagata.masaaki}@lab.ntt.co.jp

1 Introduction

Pre-ordering methods (Isozaki et al., 2010b; Wu et al., 2011) have achieved state-of-the-art translation accuracies for translating between languages with distinct word orders, such as from English to Japanese. For example, the Head-Final English (HFE) (Isozaki et al., 2010b) based approach achieved the first rank in the NTCIR-9 English-to-Japanese patent translation task (Goto et al., 2011). Compared with HFE, Predicate-Argument Structures (PASs), generated by a state-of-the-art head-driven phrase structure grammar (HPSG) (Pollard and Sag, 1994; Sag et al., 2003) parser Enju1 (Miyao and Tsujii, 2008), based pre-ordering method (Wu et al., 2011) is language independent and achieved comparable translation accuracies.

However, a shortage of current PAS-based pre-ordering method is that, the relative position between a predicate and its modifiee node is ignored. Of the 46 predicate types in the Enju HPSG trees, there are 10 types that contain modifiee nodes, such as aux_mod_arg12, verb_mod_arg1, prep_mod_arg12, etc. In this paper, we explicitly make use of the relative positions between predicates and their modifiee nodes during pre-ordering rule extraction. We found in our currently used training data, there are only 0.7% predicate types that contain modifiee nodes. Consequently, experiments on English-to-Japanese translation did not show a significant improvement on the translation accuracies. However, we still argue that our improved PAS-based pre-ordering approach is now complete and should be further investigated by being applied to translate English into other languages.

2 PAS Based Pre-ordering

In (Wu et al., 2011), we have proposed a pre-ordering approach based on the PASs of source sentences. Specially, we extracted fine-grained pre-ordering rules among a predicate word and its argument phrases. By referring to the word alignment2, the relative positions among the predicate and the argument nodes are first determined by sorting and then recorded in the pre-ordering rules. Later, through the usage of a sequence of pre-ordering rules, the word order of an original source sentence is (approximately) changed into the word order of the target sentence. Compared with previous pre-ordering approaches, PASs have the following merits for describing reordering phenomena:

- predicates, corresponding to the terminal words, express reordering patterns in a lexicalized level;
- arguments, corresponding to the non-terminal nodes/phrases, express reordering patterns in an abstract level;
- predicates and arguments provide a fine-grained classification of the reordering patterns since they include factored representations of syntactic features.

During pre-ordering rule extraction, we traverse the terminal nodes from left to right and collect their argument nodes in the source HPSG tree. We use minimum covering trees (MCTs) as defined in our earlier work (Wu et al., 2010) to express the left-hand-side of pre-ordering patterns. A MCT exactly

1http://www-tsujii.is.s.u-tokyo.ac.jp/enju/index.html
2The word alignments are gained by running GIZA++ (Och and Ney, 2003).
takes a predicate node and all its argument nodes as the leaf nodes. The root of a MCT is the shared ancestor node which is nearest to the leaf nodes of MCT. Examples of MCT can be found in (Wu et al., 2010). When the MCT of a predicate word is determined, we can easily sort the relative positions of the leaf nodes based on the pre-generated word alignments.

When applying the extracted pre-ordering rules, we also collect the MCTs from the given HPSG tree of the source sentence, and then perform the following three steps:

1. rule matching, i.e., seek available pre-ordering rules for a given MCT;

2. bottom-up rule applying, i.e., generate the n-best reordered source phrases based on the pre-ordering rules; and,

3. sentence collecting, here, for retraining word alignment, we only pack one reordered sentence ranked by the highest frequency pre-ordering rules.

After rule application, we retrain the word alignments by using the pre-ordered source sentences and the original target sentences.

3 PAS Types with Modifiee

Of the 46 predicate types used in the HPSG trees (Miyao and Tsujii, 2008), there are 10 types that contain modifiee nodes, as listed in Table 1. In the training data, these 10 types occur only 0.7% of all the 46 predicate types. There are several points in Table 1, which lead to our improved pre-ordering approach:

- argument can takes “unk”, i.e., the real argument is not shown in the input sentence. The first example sentence of verb_mod_arg123 stands for this case. Thus, we will skip this unknown argument during pre-ordering extraction and applying;

- there are overlapping among the argument phrases and the modifiee phrase. The second example sentence of adj_mod_arg1 stands for this case. In this case, we only use the MCTs that cover the predicate node and the non-terminal nodes which cover the larger scale phrases.

By taking the modifiee nodes into consideration, a PAS-based pre-ordering rule is defined to be a five-tuple: \(<pw, args, mod, srcOrder, trgOrder>\). Here, \(pw\) is the predicate word, \(args\) are the argument nodes of \(pw\), \(mod\) is the modifiee node of \(pw\), and \(srcOrder/trgOrder\) respectively store the relative positions among \(pw, args,\) and \(mod\) in the source/target language sides. It is trivial to modify the pre-ordering rule extracting and applying algorithm in (Wu et al., 2011) by adding \(mod\). For simplicity, we skip the detailed description here.

4 Experiments

We use the NTCIR-9 English-Japanese patent corpus\(^3\) as our experiment set. For direct comparison to our previous work (Wu et al., 2011), we again split the original development set averagely into two parts, named dev.a and dev.b. In our experiments, we first take dev.a as our development set for minimum-error rate tuning (Och, 2003) and then report the final translation accuracies on dev.b. We use the configuration of the official baseline system\(^4\):

- Moses\(^5\) (Koehn et al., 2007): revision = “3717” as the baseline decoder;
- GIZA++: giza-pp-v1.0.3\(^6\) (Och and Ney, 2003) for first training word alignment using the original English sentences for pre-ordering rule extraction, and then for retraining word alignments using the pre-ordered English sentences;
- SRILM\(^7\) (Stolcke, 2002): version 1.5.12 for training a 5-gram language model using the target sentences of the total training set;
- Additional scripts\(^8\): for preprocessing English sentences and cleaning up too long (# of words > 40) parallel sentences;

\(^3\)http://ntcir.nii.ac.jp/PatentMT/
\(^4\)http://ntcir.nii.ac.jp/PatentMT/baselineSystems
\(^5\)http://www.statmt.org/moses/
\(^6\)http://giza-pp.googlecode.com/files/giza-pp-v1.0.3.tar.gz
\(^7\)http://www.speech.sri.com/projects/srilm/
\(^8\)http://homepages.inf.ed.ac.uk/jschroe1/how-to/scripts.tgz
Example Sentences

<table>
<thead>
<tr>
<th>PAS Type</th>
<th>Example Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>adj_mod_arg1</td>
<td>In addition, the values of the clearances $c_1$ are maintained unchanged whether the product container 2 or the washing container 23 is selectively mounted.</td>
</tr>
<tr>
<td>adj_mod_arg12</td>
<td>The frame structure determining module $g_1$ thus remains unable to receive a frame structure flag and thus unextractable [in the “ extraction result ” (s2007)]m.</td>
</tr>
<tr>
<td>aux_mod_arg12</td>
<td>The shift lever 12 can be shifted in the directions indicated by the arrows $a$ and $b$ shown in fig. 3 about the retainer 14 by operating a shift knob 13 mounted on the upper end of the shift lever 12m.</td>
</tr>
<tr>
<td>comp_mod_arg1</td>
<td>Basically, as shown in the plan view of fig. 31, stability is secured by providing two sets of guide rollers 3 for clamping the guide 5 from both sides thereof to support the chassis $L_1$.</td>
</tr>
<tr>
<td>prep_mod_arg12</td>
<td>The noise factor (1NF) of an amplifier $g_2$ will now be considered $m$.</td>
</tr>
<tr>
<td>prep_mod_arg123</td>
<td>Additional components depending upon the type of vehicle. Furthermore, a photographic device (not shown) $g_1$, comprising a camera, illumination lamps and so forth, is installed on xy table 54 in this embodiment $m$.</td>
</tr>
<tr>
<td>verb_mod_arg1</td>
<td>Referring now to the accompanying drawing, a description $m$ will be given of the embodiments of the present invention $m$.</td>
</tr>
<tr>
<td>verb_mod_arg12</td>
<td>By using the above bolts 46 and 50, any requisite components $g_1$ can be fixed to desired positions on the body structure by an easy mounting operation $m$.</td>
</tr>
<tr>
<td>verb_mod_arg123</td>
<td>It is possible to obtain the noncontact propelling driving force $g_2$, making it possible to make the driving device compact $m$.</td>
</tr>
</tbody>
</table>

Table 1: The types of predicate-argument structures that contain modifiee nodes. For each type, we list two example sentences (except prep_mod_arg123 and verb_mod_arg1234 whose examples are not found in the training data). In the example sentences, the predicate words are shown in italic font (cycled by boxes) and their arguments are underlined and subscripted with an argument number. In addition, modifiee nodes are underlined and subscripted with ’m’.

- Japanese word segmentation: Mecab v0.98 with the dictionary of mecab-ipadic-2.7.0-20070801.tar.gz.9

The statistics of the filtered training set, dev.a, and dev.b are shown in Table 2. The success parsing rate ranges from 98.7% to 99.3% by using Enju2.3.1. The averaged parsing time for each English sentence ranges from 0.30 to 0.48 seconds.

Table 2: Statistics of the experiment sets. Here, suc. = success, sec. = second, sent. = sentence.

<table>
<thead>
<tr>
<th>Code a</th>
<th>Train</th>
<th>Dev.a</th>
<th>Dev.b</th>
</tr>
</thead>
<tbody>
<tr>
<td># of sentence</td>
<td>2,032,679</td>
<td>1,000</td>
<td>1,000</td>
</tr>
<tr>
<td># of English words</td>
<td>48,322,058</td>
<td>31,890</td>
<td>31,935</td>
</tr>
<tr>
<td>Enju suc. rate</td>
<td>99.3%</td>
<td>98.9%</td>
<td>98.7%</td>
</tr>
<tr>
<td>parse time (sec./sent.)</td>
<td>0.30</td>
<td>0.38</td>
<td>0.48</td>
</tr>
<tr>
<td># of Japanese words</td>
<td>53,865,629</td>
<td>37,066</td>
<td>35,921</td>
</tr>
</tbody>
</table>

9http://sourceforge.net/projects/mecab/files/mecab-ipadic-2.7.0-20070801.tar.gz
10http://sourceforge.net/projects/mecab/files/mecab-ipadic

The BLEU (Papineni et al., 2002) and RIBES11 scores of the original and improved pre-ordering approaches.

11Code available at http://www.kecl.ntt.co.jp/icl/lirg/ribes
RIBES is the software implementation of Normalized Kendall’s
Table 3: Translation accuracies of the original and improved PAS based pre-ordering approach. The results of the original PAS-based approach have been reported in our previous work (Wu et al., 2011). '*' stands for the improved approach.

<table>
<thead>
<tr>
<th>Source sent.</th>
<th>BLEU</th>
<th>RIBES</th>
<th>BLEU*</th>
<th>RIBES*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original sent.</td>
<td>0.2773</td>
<td>0.6619</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>PAS-a</td>
<td>0.3088</td>
<td>0.7406</td>
<td>0.3098</td>
<td>0.7346</td>
</tr>
<tr>
<td>PAS-b</td>
<td>0.3054</td>
<td>0.7334</td>
<td>0.3025</td>
<td>0.7284</td>
</tr>
<tr>
<td>PAS-c</td>
<td>0.3063</td>
<td>0.7336</td>
<td>0.3021</td>
<td>0.7255</td>
</tr>
<tr>
<td>PAS-d</td>
<td>0.3020</td>
<td>0.7265</td>
<td>0.3007</td>
<td>0.7195</td>
</tr>
</tbody>
</table>

By comparing the results, we found that the improved approach is comparable to the original pre-ordering approach as described in (Wu et al., 2011). Under PAS-a, the BLEU score is slightly better yet the RIBES score is slightly worse. Recall that there are only 0.7% predicate types contain modifie nodes, we argue this result is reasonable. However, since this number is corpus-dependent and our approach is language-independent, we still argue it is valuable to investigate our approach by using other bilingual corpora and translating other language pairs.

5 Conclusion

We have improved our previous PAS-based pre-ordering approach (Wu et al., 2011) by further considering the relative positions among predicate words and their modifie phrases. Specially, we explicitly made use of the relative positions (before and after translating) during pre-ordering rule extracting and applying. Unfortunately, the improved pre-ordering approach did not achieve significant improvements in terms of English-to-Japanese patent translation. We argue this result is due to the specified bilingual corpus. We further argue that our improved PAS-based pre-ordering approach is complete now and can be applied to translate English into other languages with distinct word orders, such as Korea, Hindi, and Urdu.

References


---

\[\tau\] as proposed by (Isozaki et al., 2010a) to automatically evaluate the translation between distant language pairs based on rank correlation coefficients and significantly penalizes word order mistakes.

12Please refer to Table 6 in (Wu et al., 2011) for the definitions of template a, b, c, and d.