Rule-based Syntactic Preprocessing for Syntax-based Machine Translation

Yuto Hatakoshi, Graham Neubig, Sakriani Sakti, Tomoki Toda, Satoshi Nakamura
Nara Institute of Science and Technology
Graduate School of Information Science
Takayama, Ikoma, Nara 630-0192, Japan
{hatakoshi.yuto.hq8,neubig,ssakti,tomoki,s-nakamura}@is.naist.jp

Abstract

Several preprocessing techniques using syntactic information and linguistically motivated rules have been proposed to improve the quality of phrase-based machine translation (PBMT) output. On the other hand, there has been little work on similar techniques in the context of other translation formalisms such as syntax-based SMT. In this paper, we examine whether the sort of rule-based syntactic preprocessing approaches that have proved beneficial for PBMT can contribute to syntax-based SMT. Specifically, we tailor a highly successful preprocessing method for English-Japanese PBMT to syntax-based SMT, and find that while the gains achievable are smaller than those for PBMT, significant improvements in accuracy can be realized.

1 Introduction

In the widely-studied framework of phrase-based machine translation (PBMT) (Koehn et al., 2003), translation probabilities between phrases consisting of multiple words are calculated, and translated phrases are rearranged by the reordering model in the appropriate target language order. While PBMT provides a light-weight framework to learn translation models and achieves high translation quality in many language pairs, it does not directly incorporate morphological or syntactic information. Thus, many preprocessing methods for PBMT using these types of information have been proposed. Methods include preprocessing to obtain accurate word alignments by the division of the prefix of verbs (Nießen and Ney, 2000), preprocessing to reduce the errors in verb conjugation and noun case agreement (Avramidis and Koehn, 2008), and many others. The effectiveness of the syntactic preprocessing for PBMT has been supported by these and various related works.

In particular, much attention has been paid to preordering (Xia and McCord, 2004; Collins et al., 2005), a class of preprocessing methods for PBMT. PBMT has well-known problems with language pairs that have very different word order, due to the fact that the reordering model has difficulty estimating the probability of long distance reorderings. Therefore, preordering methods attempt to improve the translation quality of PBMT by rearranging source language sentences into an order closer to that of the target language. It's often the case that preordering methods are based on rule-based approaches, and these methods have achieved great success in ameliorating the word ordering problems faced by PBMT (Collins et al., 2005; Xu et al., 2009; Isozaki et al., 2010b).

One particularly successful example of rule-based syntactic preprocessing is Head Finalization (Isozaki et al., 2010b), a method of syntactic preprocessing for English to Japanese translation that has significantly improved translation quality of English-Japanese PBMT using simple rules based on the syntactic structure of the two languages. The most central part of the method, as indicated by its name, is a reordering rule that moves the English head word to the end of the corresponding syntactic constituents to match the head-final syntactic structure of Japanese sentences. Head Finalization also contains some additional preprocessing steps such as determiner elimination, particle insertion and singularization to generate a sentence that is closer to Japanese grammatical structure.

In addition to PBMT, there has also recently been interest in syntax-based SMT (Yamada and Knight, 2001; Liu et al., 2006), which translates using syntactic information. However, few attempts have been made at syntactic preprocessing for syntax-based SMT, as the syntactic information given by the parser is already incorporated directly in the translation model. Notable excep-

Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 34–42,
October 25, 2014, Doha, Qatar.

tions include methods to perform tree transformations improving correspondence between the sentence structure and word alignment (Burkett and Klein, 2012), methods for binarizing parse trees to match word alignments (Zhang et al., 2006), and methods for adjusting label sets to be more appropriate for syntax-based SMT (Hanneman and Lavie, 2011; Tamura et al., 2013). It should be noted that these methods of syntactic preprocessing for syntax-based SMT are all based on automatically learned rules, and there has been little investigation of the manually-created linguistically-motivated rules that have proved useful in preprocessing for PBMT.

In this paper, we examine whether rule-based syntactic preprocessing methods designed for PBMT can contribute anything to syntax-based machine translation. Specifically, we examine whether the reordering and lexical processing of Head Finalization contributes to the improvement of syntax-based machine translation as it did for PBMT. Additionally, we examine whether it is possible to incorporate the intuitions behind the Head Finalization reordering rules as soft constraints by incorporating them as a decoder feature. As a result of our experiments, we demonstrate that rule-based lexical processing can contribute to improvement of translation quality of syntax-based machine translation.

2 Head Finalization

Head Finalization is a syntactic preprocessing method for English to Japanese PBMT, reducing grammatical errors through reordering and lexical processing. Isozaki et al. (2010b) have reported that translation quality of English-Japanese PBMT is significantly improved using a translation model learned by English sentences preprocessed by Head Finalization and Japanese sentences. In fact, this method achieved the highest results in the large scale NTCIR 2011 evaluation (Sudoh et al., 2011), the first time a statistical machine translation (SMT) surpassed rule-based systems for this very difficult language pair, demonstrating the utility of these simple syntactic transformations from the point of view of PBMT.

2.1 Reordering

The reordering process of Head Finalization uses a simple rule based on the features of Japanese grammar. To convert English sentence into Japanese word order, the English sentence is first parsed using a syntactic parser, and then head words are moved to the end of the corresponding syntactic constituents in each non-terminal node of the English syntax tree. This helps replicate the ordering of words in Japanese grammar, where syntactic head words come after non-head (dependent) words.

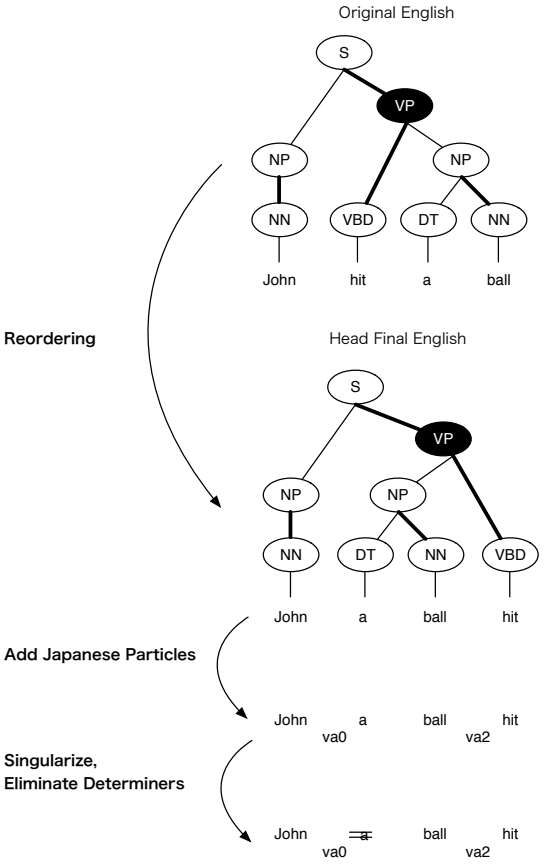

Figure 1: Head Finalization

Figure 1 shows an example of the application of Head Finalization to an English sentence. The head node of the English syntax tree is connected to the parent node by a bold line. When this node is the first child node, we move it behind the dependent node in order to convert the English sentence into head final order. In this case, moving the head node VBD of black node VP to the end of this node, we can obtain the sentence "John a ball hit" which is in a word order similar to Japanese.

2.2 Lexical Processing

In addition to reordering, Head Finalization conducts the following three steps that do not affect word order. These steps do not change the word

ordering, but still result in an improvement of translation quality, and it can be assumed that the effect of this variety of syntactic preprocessing is not only applicable to PBMT but also other translation methods that do not share PBMT's problems of reordering such as syntax-based SMT. The three steps included are as follows:

1. Pseudo-particle insertion
2. Determiner ("a", "an", "the") elimination
3. Singularization

The motivation for the first step is that in contrast to English, which has relatively rigid word order and marks grammatical cases of many noun phrases according to their position relative to the verb, Japanese marks the topic, subject, and object using case marking particles. As Japanese particles are not found in English, Head Finalization inserts "pseudo-particles" to prevent a mistranslation or lack of particles in the translation process. In the pseudo-particle insertion process (1), we insert the following three types of pseudo-particles equivalent to Japanese case markers "wa" (topic), "ga" (subject) or "wo" (object).

- va0: Subject particle of the main verb
- va1: Subject particle of other verbs
- va2: Object particle of any verb

In the example of Figure 1, we insert the topic particle va0 behind of "John", which is a subject of a verb "hit" and object particle va2 at the back of object "ball."

Another source of divergence between the two languages stems from the fact that Japanese does not contain determiners or makes distinctions between singular and plural by inflection of nouns. Thus, to generate a sentence that is closer to Japanese, Head Finalization eliminates determiners (2) and singularizes plural nouns (3) in addition to the pseudo-particle insertion.

In Figure 1, we can see that applying these three processes to the source English sentence results in the sentence "John va0 (*wa*) ball va2 (*wo*) hit" which closely resembles the structure of the Japanese translation "*jon wa bo-ru wo utta*."

3 Syntax-based Statistical Machine Translation

Syntax-based SMT is a method for statistical translation using syntactic information of the sentence (Yamada and Knight, 2001; Liu et al., 2006). By using translation patterns following the structure of linguistic syntax trees, syntax-based translations often makes it possible to achieve more grammatical translations and reorderings compared with PBMT. In this section, we describe tree-to-string (T2S) machine translation based on synchronous tree substitution grammars (STSG) (Graehl et al., 2008), the variety of syntax-based SMT that we use in our experiments.

T2S captures the syntactic relationship between two languages by using the syntactic structure of parsing results of the source sentence. Each translation pattern is expressed as a source sentence subtree using rules including variables. The following example of a translation pattern include two noun phrases NP_0 and NP_1, which are translated and inserted into the target placeholders X_0 and X_1 respectively. The decoder generates the translated sentence in consideration of the probability of translation pattern itself and translations of the subtrees of NP_0 and NP_1.

$\mathrm{S}((\mathrm{NP}_0)\ (\mathrm{VP}(\mathrm{VBD\ hit})\ (\mathrm{NP}_1)))$
$\rightarrow \mathrm{X}_0$ *wa* X_1 *wo utta*

T2S has several advantages over PBMT. First, because the space of translation candidates is reduced using the source sentence subtree, it is often possible to generate translations that are more accurate, particularly with regards to long-distance reordering, as long as the source parse is correct. Second, the time to generate translation results is also reduced because the search space is smaller than PBMT. On the other hand, because T2S generates translation results using the result of automatic parsing, translation quality highly depends on the accuracy of the parser.

4 Applying Syntactic Preprocessing to Syntax-based Machine Translation

In this section, we describe our proposed method to apply Head Finalization to T2S translation. Specifically, we examine two methods for incorporating the Head Finalization rules into syntax-based SMT: through applying them as preprocessing step to the trees used in T2S translation, and

through adding reordering information as a feature of the translation patterns.

4.1 Syntactic Preprocessing for T2S

We applied the two types of processing shown in Table 1 as preprocessing for T2S. This is similar to preprocessing for PBMT with the exception that preprocessing for PBMT results in a transformed string, and preprocessing for T2S results in a transformed tree. In the following sections, we elaborate on methods for applying these preprocessing steps to T2S and some effects expected therefrom.

Table 1: Syntactic preprocessing applied to T2S

Preprocessing	Description
Reordering	Reordering based on Japanese typical head-final grammatical structure
Lexical Processing	Pseudo-particle insertion, determiner elimination, singularization

4.1.1 Reordering for T2S

In the case of PBMT, reordering is used to change the source sentence word order to be closer to that of the target, reducing the burden on the relatively weak PBMT reordering models. On the other hand, because translation patterns of T2S are expressed by using source sentence subtrees, the effect of reordering problems are relatively small, and the majority of reordering rules specified by hand can be automatically learned in a well-trained T2S model. Therefore, preordering is not expected to cause large gains, unlike in the case of PBMT.

However, it can also be thought that preordering can still have a positive influence on the translation model training process, particularly by increasing alignment accuracy. For example, training methods for word alignment such as the IBM or HMM models (Och and Ney, 2003) are affected by word order, and word alignment may be improved by moving word order closer between the two languages. As alignment accuracy plays a important role in T2S translation (Neubig and Duh, 2014), it is reasonable to hypothesize that reordering may also have a positive effect on T2S. In terms of the actual incorporation with the T2S system, we simply follow the process in Figure 1, but output the reordered tree instead of only the reordered terminal nodes as is done for PBMT.

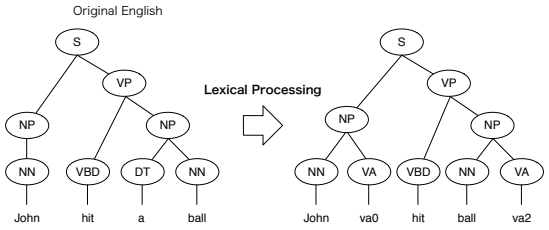

Figure 2: A method of applying Lexical Processing

4.1.2 Lexical Processing for T2S

In comparison to reordering, Lexical Processing may be expected to have a larger effect on T2S, as it will both have the potential to increase alignment accuracy, and remove the burden of learning rules to perform simple systematic changes that can be written by hand. Figure 2 shows an example of the application of Lexical Processing to transform not strings, but trees.

In the pseudo-particle insertion component, three pseudo particles "va0," "va1," and "va2" (as shown in Section 2.2) are added in the source English syntax tree as terminal nodes with the non-terminal node "VA". As illustrated in Figure 2, particles are inserted as children at the end of the corresponding NP node. For example, in the figure the topic particle "va0" is inserted after "John," subject of the verb "hit," and the object particle "va2" is inserted at the end of the NP for "ball," the object.

In the determiner elimination process, terminal nodes "a," "an," and "the" are eliminated along with non-terminal node DT. Determiner "a" and its corresponding non-terminal DT are eliminated in the Figure 2 example.

Singularization, like in the processing for PBMT, simply changes plural noun terminals to their base form.

4.2 Reordering Information as Soft Constraints

As described in section 4.1.1, T2S work well on language pairs that have very different word order, but is sensitive to alignment accuracy. On the other hand, we know that in most cases Japanese word order tends to be head final, and thus any rules that do not obey head final order may be the result of bad alignments. On the other hand, there are some cases where head final word order is not applicable (such as sentences that contain the determiner

"no," or situations where non-literal translations are necessary) and a hard constraint to obey head-final word order could be detrimental.

In order to incorporate this intuition, we add a feature (HF-feature) to translation patterns that conform to the reordering rules of Head Finalization. This gives the decoder ability to discern translation patterns that follow the canonical reordering patterns in English-Japanese translation, and has the potential to improve translation quality in the T2S translation model.

We use the log-linear approach (Och, 2003) to add the Head Finalization feature (HF-feature). As in the standard log-linear model, a source sentence $\boldsymbol{f}$ is translated into a target language sentence $\boldsymbol{e}$, by searching for the sentence maximizing the score:

$$\hat{\boldsymbol{e}} = \arg\max_{\boldsymbol{e}} \boldsymbol{w}^{\mathrm{T}} \cdot \boldsymbol{h}(\boldsymbol{f}, \boldsymbol{e}). \quad (1)$$

where $\boldsymbol{h}(\boldsymbol{f}, \boldsymbol{e})$ is a feature function vector. $\boldsymbol{w}$ is a weight vector that scales the contribution from each feature. Each feature can take any real value which is useful to improve translation quality, such as the log of the n-gram language model probability to represent fluency, or lexical/phrase translation probability to capture the word or phrase-wise correspondence. Thus, if we can incorporate the information about reordering expressed by the Head Finalization reordering rule as a features in this model, we can learn weights to inform the decoder that it should generally follow this canonical ordering.

Figure 3 shows a procedure of Head Finalization feature (HF-feature) addition. To add the HF-feature to translation patterns, we examine the translation rules, along with the alignments between target and source terminals and non-terminals. First, we apply the Reordering to the source side of the translation pattern subtree according to the canonical head-final reordering rule. Second, we examine whether the word order of the reordered translation pattern matches with that of the target translation pattern for which the word alignment is non-crossing, indicating that the target string is also in head-final word order. Finally, we set a binary feature ($h_{\mathrm{HF}}(\boldsymbol{f}, \boldsymbol{e}) = 1$) if the target word order obeys the head final order. This feature is only applied to translation patterns for which the number of target side words is greater than or equal to two.

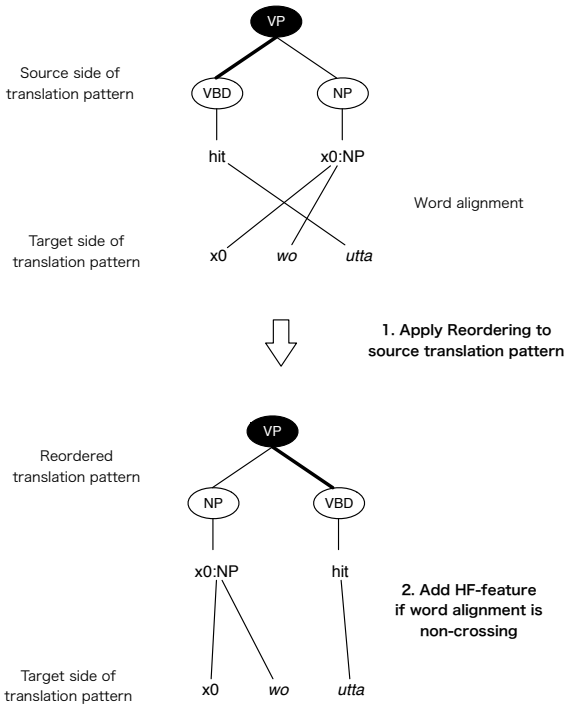

Figure 3: Procedure of HF-feature addition

Table 2: The details of NTCIR7

Dataset	Lang	Words	Sentences	Average length
train	En	99.0M	3.08M	32.13
	Ja	117M	3.08M	37.99
dev	En	28.6k	0.82k	34.83
	Ja	33.5k	0.82k	40.77
test	En	44.3k	1.38k	32.11
	Ja	52.4k	1.38k	37.99

5 Experiment

In our experiment, we examined how much each of the preprocessing steps (Reordering, Lexical Processing) contribute to improve the translation quality of PBMT and T2S. We also examined the improvement in translation quality of T2S by the introduction of the Head Finalization feature.

5.1 Experimental Environment

For our English to Japanese translation experiments, we used NTCIR7 PATENT-MT's Patent corpus (Fujii et al., 2008). Table 2 shows the details of training data (train), development data (dev), and test data (test).

As the PBMT and T2S engines, we used the Moses (Koehn et al., 2007) and Travatar (Neubig, 2013) translation toolkits with the default settings.

Enju (Miyao and Tsujii, 2002) is used to parse English sentences and KyTea (Neubig et al., 2011) is used as a Japanese tokenizer. We generated word alignments using GIZA++ (Och and Ney, 2003) and trained a Kneser-Ney smoothed 5-gram LM using SRILM (Stolcke et al., 2011). Minimum Error Rate Training (MERT) (Och, 2003) is used for tuning to optimize BLEU. MERT is replicated three times to provide performance stability on test set evaluation (Clark et al., 2011).

We used BLEU (Papineni et al., 2002) and RIBES (Isozaki et al., 2010a) as evaluation measures of translation quality. RIBES is an evaluation method that focuses on word reordering information, and is known to have high correlation with human judgement for language pairs that have very different word order such as English-Japanese.

5.2 Result

Table 3 shows translation quality for each combination of HF-feature, Reordering, and Lexical Processing. Scores in boldface indicate no significant difference in comparison with the condition that has highest translation quality using the bootstrap resampling method (Koehn, 2004) ($p < 0.05$).

For PBMT, we can see that reordering plays an extremely important role, with the highest BLEU and RIBES scores being achieved when using Reordering preprocessing (line 3, 4). Lexical Processing also provided a slight performance gain for PBMT. When we applied Lexical Processing to PBMT, BLEU and RIBES scores were improved (line 1 vs 2), although this gain was not significant when Reordering was performed as well.

Overall T2S without any preprocessing achieved better translation quality than all conditions of PBMT (line 1 of T2S vs line 1-4 of PBMT). In addition, BLEU and RIBES score of T2S were clearly improved by Lexical Processing (line 2, 4, 6, 8 vs line 1, 3, 5, 7), and these scores are the highest of all conditions. On the other hand, Reordering and HF-Feature addition had no positive effect, and actually tended to slightly hurt translation accuracy.

5.3 Analysis of Preprocessing

With regards to PBMT, as previous works on preordering have already indicated, BLEU and RIBES scores were significantly improved by Reordering. In addition, Lexical Processing also contributed to improve translation quality of PBMT slightly. We also investigated the influence that each element of Lexical Processing (pseudo-particle insertion, determiner elimination, singularization) had on translation quality, and found that the gains were mainly provided by particle insertion, with little effect from determiner elimination or singularization.

Table 5: Optimized weight of HF-feature in each condition

HF-feature	Reordering	Word Processing	Weight of HF-feature
+	-	-	-0.00707078
+	-	+	0.00524676
+	+	-	0.156724
+	+	+	-0.121326

Although Reordering was effective for PBMT, it did not provide any benefit for T2S. This indicates that T2S can already conduct long distance word reordering relatively correctly, and word alignment quality was not improved as much as expected by closing the gap in word order between the two languages. This was verified by a subjective evaluation of the data, finding very few major reordering issues in the sentences translated by T2S.

On the other hand, Lexical Processing functioned effectively for not only PBMT but also T2S. When added to the baseline, lexical processing on its own resulted in a gain of 0.57 BLEU, and 0.99 RIBES points, a significant improvement, with similar gains being seen in other settings as well.

Table 4 demonstrates a typical example of the improvement of the translation result due to Lexical Processing. It can be seen that translation performance of particles (indicated by underlined words) was improved. The underlined particle is in the direct object position of the verb that corresponds to "comprises" in English, and thus should be given the object particle "を *wo*" as in the reference and the system using Lexical Processing. On the other hand, in the baseline system the genitive "と *to*" is generated instead due to misaligned particles being inserted in an incorrect position in the translation rules.

5.4 Analysis of Feature Addition

Our experimental results indicated that translation quality is not improved by HF-feature addition (line 1-4 vs line 5-8). We conjecture that the reason why HF-feature did not contribute to an im-

Table 3: Translation quality by combination of HF-feature, Reordering, and Lexical Processing. Bold indicates results that are not statistically significantly different from the best result (39.60 BLEU in line 4 and 79.47 RIBES in line 2).

ID				PBMT		T2S	
	HF-feature	Reordering	Lexical Processing	BLEU	RIBES	BLEU	RIBES
1	-	-	-	32.11	69.06	38.94	78.48
2	-	-	+	33.16	70.19	**39.51**	**79.47**
3	-	+	-	37.62	77.56	38.44	78.48
4	-	+	+	37.77	77.71	**39.60**	**79.26**
5	+	-	-	—	—	38.74	78.33
6	+	-	+	—	—	**39.29**	**79.23**
7	+	+	-	—	—	38.48	78.44
8	+	+	+	—	—	**39.38**	**79.21**

Table 4: Improvement of translation results due to Lexical Processing

Source	another connector 96 , which is matable with this cable connector 90 , comprises a plurality of male contacts 98 aligned in a row in an electrically insulative housing 97 as shown in the figure .
Reference	このケーブルコネクタ９０と嵌合接続される相手コネクタ９６は、図示のように、絶縁ハウジング９７内に雄コンタクト９８を整列保持して構成される。
- Lexical Processing	このケーブルコネクタ９０は相手コネクタ９６は、図に示すように、電気絶縁性のハウジング９７に一列に並ぶ複数の雄型コンタクト９８とから構成されている。
+ Lexical Processing	このケーブルコネクタ９０と相手コネクタ９６は、図に示すように、電気絶縁性のハウジング９７に一列に並ぶ複数の雄型コンタクト９８を有して構成される。

provement in translation quality is that the reordering quality achieved by T2S translation was already sufficiently high, and the initial feature led to confusion in MERT optimization.

Table 5 shows the optimized weight of the HF feature in each condition. From this table, we can see that in two of the conditions positive weights are learned, and in two of the conditions negative weights are learned. This indicates that there is no consistent pattern of learning weights that correspond to our intuition that head-final rules should receive higher preference.

It is possible that other optimization methods, or a more sophisticated way of inserting these features into the translation rules could help alleviate these problems.

6 Conclusion

In this paper, we analyzed the effect of applying syntactic preprocessing methods to syntax-based SMT. Additionally, we have adapted reordering rules as a decoder feature. The results showed that lexical processing, specifically insertion of pseudo-particles, contributed to improving translation quality, and it was effective as preprocessing for T2S.

It should be noted that this paper, while demonstrating that the simple rule-based syntactic processing methods that have been useful for PBMT can also contribute to T2S in English-Japanese translation, more work is required to ensure that this will generalize to other settings. A next step in our inquiry is the generalization of these results to other proposed preprocessing techniques and other language pairs. In addition, we would like to try two ways described below. First, it is likely that other tree transformations, for example changing the internal structure of the tree by moving children to different nodes, would help in cases where it is common to translate into highly divergent syntactic structures between the source and target languages. Second, we plan to investigate other ways of incorporating the preprocessing rules as a soft constraints, such as using n-best lists or forests to enode many possible sentence interpretations.

References

Eleftherios Avramidis and Philipp Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. In *Annual Meeting of the*

Association for Computational Linguistics (ACL), pages 763–770.

David Burkett and Dan Klein. 2012. Transforming trees to improve syntactic convergence. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 863–872.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 176–181.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 531–540.

Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro, Terumasa Ehara, Hiroshi Echizenya, and Sayori Shimohata. 2008. Overview of the patent translation task at the NTCIR-7 workshop. In *Proceedings of the 7th NTCIR Workshop Meeting*, pages 389–400.

Jonathan Graehl, Kevin Knight, and Jonathan May. 2008. Training tree transducers. *Computational Linguistics*, pages 391–427.

Greg Hanneman and Alon Lavie. 2011. Automatic category label coarsening for syntax-based machine translation. In *Workshop on Syntax and Structure in Statistical Translation*, pages 98–106.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 944–952.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for SOV languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 244–251.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *North American Chapter of the Association for Computational Linguistics*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 177–180.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 609–616.

Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proceedings of the second international conference on Human Language Technology Research*, pages 292–297.

Graham Neubig and Kevin Duh. 2014. On the elements of an accurate tree-to-string machine translation system. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 143–149.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 529–533.

Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. *Annual Meeting of the Association for Computational Linguistics (ACL)*, page 91.

Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*, pages 1081–1085.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, pages 19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, page 5.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata, Xianchao Wu, Takuya Matsuzaki, and Jun ' ichi Tsujii. 2011. NTT-UT statistical machine translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR*, pages 585–592.

Akihiro Tamura, Taro Watanabe, Eiichiro Sumita, Hiroya Takamura, and Manabu Okumura. 2013. Part-of-speech induction in dependency trees for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 841–851.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *International Conference on Computational Linguistics (COLING)*, page 508.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve smt for subject-object-verb languages. In *North American Chapter of the Association for Computational Linguistics*, pages 245–253.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 523–530.

Hao Zhang, Liang Huang, Daniel Gildea, and Kevin Knight. 2006. Synchronous binarization for machine translation. In *North American Chapter of the Association for Computational Linguistics*, pages 256–263.