Weblio Pre-reordering Statistical Machine Translation System

Zhongyuan Zhu
Weblio Inc.
chugen.shu@weblio.co.jp

Abstract

This paper describes details of the Weblio Pre-reordering Statistical Machine Translation (SMT) System, participated in the English-Japanese translation task of 1st Workshop on Asian Translation (WAT2014). In this system, we applied the pre-reordering method described in (Zhu et al., 2014), and extended the model to obtain N-best pre-reordering results. We also utilized N-best parse trees simultaneously to explore the potential improvement for pre-reordering system with forest input.

1 Introduction

In this paper, we describe the details of Weblio Pre-reordering Statistical Machine Translation (SMT) System, experiments and some issues we faced. For this SMT system, we applied the pre-reordering method proposed in (Zhu et al., 2014). In particular, this method automatically learns pre-reordering model from word alignments and parse trees. Statistical language model is integrated in the pre-reordering model in order to reorder each node layer in parse trees. In the 1st Workshop on Asian Translation (WAT2014) (Nakazawa et al., 2014), we mainly applied this method in English-Japanese translation subtask. The parse tree we used is head-restructured CFG parse tree for English, which is also proposed in (Zhu et al., 2014). After the pre-reordering phase, we trained a conventional Phrase-based model to do the final translation.

To make some improvements, we enabled the pre-reordering system to output N-best reordering results. Also, we feed the whole translation pipe line with N-best parse trees generated by Egret parser. As a result, multiple translation hypotheses can be collected for one input sentence. Finally, we select the best hypothesis according to a balanced score.

In our experiments, the system utilizes N-best pre-reordering results shows the ability to obtain more accurate translation result. After incorporating N-best parse trees, improvements on the automatic evaluation scores are also observed.

In section 2 and 3, we briefly describe the method used for tree parsing and pre-reordering. In the remaining sections, we give some details of the experiments of our system.

2 Head-restructured CFG parse tree

In order to reorder SVO (subject-verb-object) order into SOV (subject-object-verb) order, correctly reordering long-distance words those play important roles in a sentence is crucial. Thus, the reordering model is required to capture the reordering patterns for those words. Obviously, using dependency tree should be a quick solution for this problem. As in a dependency tree, all closely related words of a specific head word come underneath that head word. All we need to do is to find the best order for the branches under those related words. In particular, if the head word is the root of the whole sentence (usually a verb), then it's reasonable to think each dependent word leads to a branch that contains a important part of the whole sentence.

However, using dependency tree naively does not work well in practice. First of all, not all components in a sentence need to be reorder. In the case of English-Japanese translation, noun phrases are usually in the identical order of English. Secondly, some local grammar structures tend to keep their unique order. For example, the combination of an adjective word with a noun usually has the same order in Japanese. But a noun follows the preposition "of" in English will appear before it in the order of Japanese. A model merely based on dependency parse trees will be sparse and hard

Proceedings of the 1st Workshop on Asian Translation (WAT2014), pages 33–38,
Tokyo, Japan, 4th October 2014.

to deal with unknown words correctly. POS (part-of-speech) tags and also structural information are still necessary.

The approach in (Zhu et al., 2014) addresses this problem by injecting sentence-level dependencies into CFG parse trees. Local grammatical structures are still kept unchanged in the parse tree. This new parse tree is called "Head-restructured CFG parse tree" in the original paper. In this paper, we use "HRCFG tree" in short to represent it. An example of HRCFG tree is shown in Figure 1.

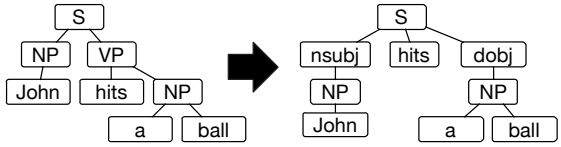

Figure 1: An example of HRCFG tree converted from CFG parse tree (A direct parent nodes of terminal nodes are now include)

In Figure 1, A normal CFG parse tree is shown in the left, and the corresponding HRCFG tree in the right. In this example, tree components explicitly shows subject, object and verb parts in the sentence. This structure with explicit annotations makes the reordering model easier to capture long-distance reordering patterns.

3 Reordering model integrated with language model

The reordering model we used in our MT systems follows the same fashion of the model in (Zhu et al., 2014). A language model is integrated to identify the best order of a node layer according to the order of target language. Although the using of language model still involves spare problem and fails to give the correct probability in some cases, it makes the implementation fairly simple.

With a bilingual training data and automatically learned word alignments given by GIZA++ (Och and Ney, 2004), we find the best order for each node layer in all parse trees, make them fit the order of aligned parts in the target sentence. Specifically, for tree nodes $\mathbf{n} = (n_1, n_2, ..., n_k)$, terminal nodes beneath n_i is defined as $\mathbf{t}^i$. Let $\mathbf{w}^i$ represent a set of words in target side which are aligned with any terminal node in $\mathbf{t}^i$. I this step, for each node layer $\mathbf{n}$, we redetermine the order of $\mathbf{n}$ according to the average position of aligned words $\mathbf{w}^i$ for each node n_i.

Then we exports a sequence of reordered non-terminal tags. For some kinds of node, non-terminal tags in the sequences are replaced by head word. After we trained a language model on them, the language model can be used to estimate the likelihood that a tag sequence follows the order of target language.

We show an example of this reordering process in Figure 2.

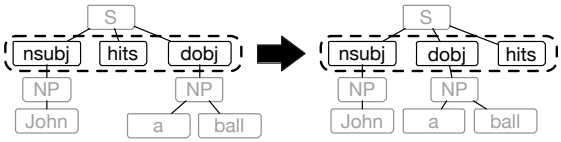

Figure 2: An example of HRCFG tree reordering

To reorder the node layer underneath the "S" node in Figure 2, we list all possible orders for the tag sequence "nsubj hits dobj" (6 possibilities in all). Then we use the language model we trained on reordered tag sequences to estimate the probability for each possible order. Finally, it is expected that the sequence "nsubj dobj hits" gets the highest language model score, as it is most closer to the order in Japanese.

The reordering operation in this fashion is applied to all node layers in the HRCFG tree. We export all terminal words in the reordered parse trees as new training data in the source side. Like Head-finalization (Isozaki et al., 2010), we also incorporate seed words in the results. So the final reordering result of the sentence shown in Figure 1 will be "John va_nsubjpass a ball va_dobj hits".

3.1 N-best reordering

In the reordering model we described above, the best order for the whole sentence is actually comprised by all *1*-best orders of every node layers in the parse tree. Although the language model usually works perfectly to give the best reordering result. In some cases, the best reordering result is unclear until the translation phase.

We give an example here, for the sentence "The rocket is launched by NASA", two plausible reordering results are shown in Table 1.

The first reordering result in Table 1 is preferred by the reordering model as "nsubjpass auxpass launched prep_by" is usually reordered into "nsubjpass prep_by launched auxpass". Unfortunately,

Table 1: Two reordering results of the sentence "The rocket is launched by NASA"

#	reordering result and corresponding translation
1	The rocket va_nsubjpass NASA by launched is ロケットは NASA によって発射された
2	NASA by The rocket va_nsubjpass launched is NASA はロケットを発射した

in this case, the translation follows the second reordering result is more natural in Japanese.

For the cases like what we show in Table 1, it's hard to find out a best order before translation. Considering N-best reordering results is necessary in order to obtain the best translation result. In our MT system, we implement this feature simply by collecting N-best reordering results for all node layers, and finally rank the reordering results by accumulated language model score.

4 Experiments

4.1 Experimental settings

For our baseline system, we use *1*-best parse trees for training and test. Stanford tokenizer and Berkeley parser (Petrov et al., 2006) are selected in the pre-processing phase in order to produce CFG parse trees. Then we obtain dependency parse trees by applying Stanford rules (Klein and Manning, 2002) to CFG parse trees. HRCFG trees are built upon these two kinds of parse trees. For the Japanese text, we use Kytea (Neubig, Nakata, and Mori, 2011) to tokenize it.

Due to the limitation of computational resource, we are only able to train our reordering model on 1.5M bilingual text (with relatively high scores in ASPEC parallel corpus) for English-to-Japanese translation task. We used this trained reordering model to reorder all training data in the source side.

We use conventional Phrase-based model implemented in Moses toolkit to finish remaining SMT pipe line. Distortion limit is set to 6 in all our experiments.

For system translates forest inputs, we use Egret parser to generate N-best packed forests. We unpack each forest and parse each individual tree to HRCFG tree. For all candidate of parse trees, we reorder them and merge same reordering results. Then for all reordering results we obtained, we translate them and record translation scores given by Moses. Finally, a best translation result is selected out by the sum of translation score and reordering score.

4.2 Experiment results

We carried out several experiments combining the use of N-best parse trees and N-best reordering results. A list of automatic evaluation scores for different system settings are listed in Table 2.

Table 2: Experiment results for different system settings

#	System	BLEU(%)	RIBES
1	*1*-best parse + *1*-best reorder	34.46	0.7817
2	N-best parse + *1*-best reorder	34.80	0.7851
3	*1*-best parse + N-best reorder	34.90	0.7857
4	N-best parse + N-best reorder	**35.10**	**0.7887**

In particular, for systems marked with "N-best parse", 30 parse trees with highest parsing scores are used. For systems marked with "N-best reorder", 10 reordering results with highest reordering scores are accepted for each parse tree. That is, for System 4, a maximum of 300 reordering results are generated for one sentence.

In WAT2014, we submitted System 3 and System 4 to human evaluation. Note that in Table 2, our in-house automatic evaluation scores are slightly different from that on the score board of WAT2014 due to different automatic evaluation pipe line we used. Official evaluation scores are listed in Table 3. Where "BASELINE" refers to Phrase-based SMT system (Koehn, Och, and Marcu, 2003) as the official baseline for human evaluation.

Table 3: Official evaluation scores in WAT2014 (kytea used for post-processing)

# System	BLEU(%)	RIBES	HUMAN
3	34.87	0.7869	**+43.250**
4	**35.04**	**0.7900**	+36.000
BASELINE	**29.80**	**0.6919**	+0.000

Our experiment results shown in Table 2 show that incorporating N-best parse tree and reordering results gained improvements for both BLEU and RIBES metrics.

In the official human evaluation, although System 4 achieved better results in automatic evaluations. Human evaluation score of it degraded compared to System 3, which only considers *1*-best parse tree.

In Figure 3 and 4, we show the growth of BLEU and RIBES when increasing candidate number considered for N-best parse trees and reordering results. Both BLEU and RIBES scores are tending to converge after we increased the N-best parse tree candidates to 30 for System 2. For System 3, the automatic evaluation scores are still increasing after 10 reordering results are considered.

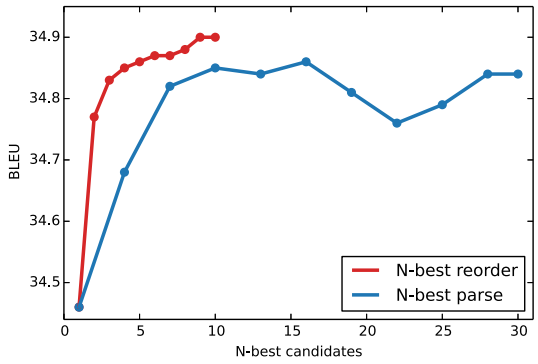

Figure 3: Growth of BLEU with increasing N-best candidates

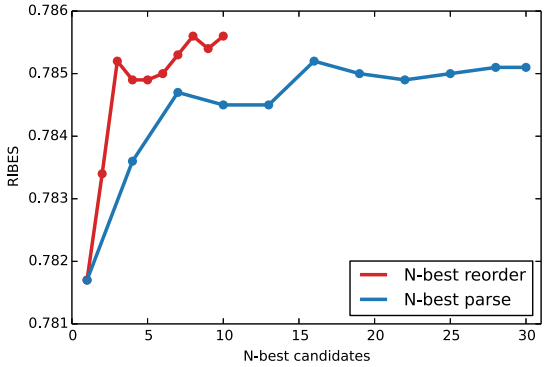

Figure 4: Growth of RIBES with increasing N-best candidates

4.3 Evaluation for pre-reordering

In this section, we evaluate the performance of pre-reordering. Follows the method described in (Isozaki et al., 2010), we estimate Kendall's τ from word alignments. A comparison of Kendall's τ distribution upon first 1.5M sentences of ASPEC corpus is shown in Figure 5.

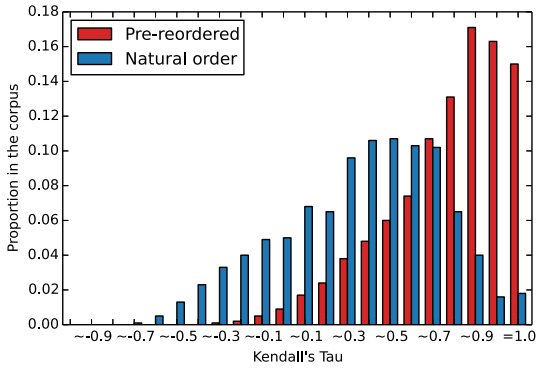

Figure 5: A comparison of Kendall's τ distribution

Average Kendall's τ of natural order and adjusted order is 0.30 and 0.71 respectively. Note that in (Isozaki et al., 2010), the algorithm for estimating Kendall's τ does not take the words with multiple alignments into account. Hence, the graph of Kendall's τ only gives a rough idea of the performance of pre-reordering. In particular, the algorithm skipped 20.30% aligned words for corpus in natural order and 14.06% aligned words for pre-reordered corpus. However, the distribution of Kendall's τ in Figure 5 gives a intuitive picture of the improvements of word order. Sentences which are fully identical in word order increased from 1.8% to 15% after pre-reordering (labeled with "=1.0" in Figure 5).

5 Error analysis

5.1 Issues of pre-reordering

Although our pre-reordering SMT system is able to produce relatively better translation results compared to baseline SMT systems. In many cases, the translation results still suffer from the defect of the reordering model. As the reordering model described in Section 3 is actually a language model built on sequences mixed with non-terminal tags and words. Involving words in the model makes the reordering more flexible, but also makes the model sparse. For some rare or unknown words, the reordering model usually fails to reorder sentences correctly.

In Table 4, we show 2 reordering samples. In Sample 1, the sentence is correctly reordered. The word "were" in English side should be placed in the end of the reordered sentence, which is expected to be translated to "された" in Japanese. In Sample 2, we replace the verb "confirmed" in Sample 1 to "observed", then the reordering model

Table 4: Samples of reordering result

#	Samples of reordering result
1	**the changes were confirmed** → the changes va_nsubjpass confirmed were
2	**the changes were observed** → the changes va_nsubjpass were observed

fails to place the word "were" into the rightmost position.

The errors like what we show in Table 4 are actually widespread in the reordering results for the ASPEC test corpus. Although in the decoding phase, the lexical distortion model of Phrase-based SMT model can partially mitigate some local errors, some critical errors still can be observed from the final translations.

5.2 Issues for Context-aware Machine Translation

In this section, we describe some efforts for utilizing context information during the translation.

We made an attempt to tackle the phrase selection problem for English-Japanese translation. In Japanese, many English words have multiple translations. Especially Japanese words in the form of Katakana usually also have corresponding expressions comprised of Chinese characters. For instance, the phrase "remote control" can be translated to both "ENKAKUSEIGYO"(遠隔制御) and "RIMOKON"(リモコン). We show the distribution of these two translations across different domains in Figure 6.

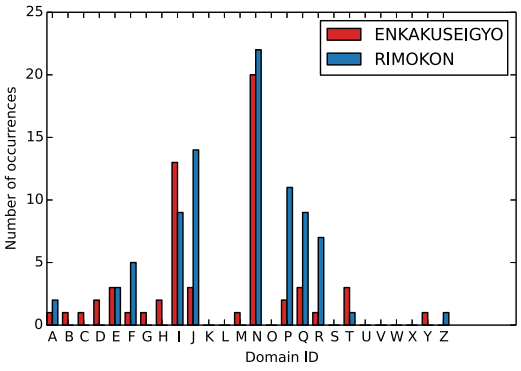

Figure 6: Translation distribution for "remote control" across several categories

From Figure 6, it's reasonable to think the phrase "RIMOKON" is more preferred in domain J, P, Q and R. While in domain N, the two phrases appear almost same times.

A simple solution is to make language model more domain-specific. We carried out experiments that simply interpolate general language model and in-domain language model. The experiment results for first three domains are shown in Figure 7.

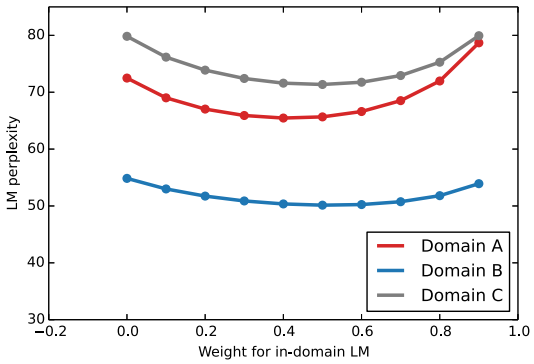

Figure 7: LM perplexity on domain-specific test data using interpolated language models

In Figure 7, we show the language model perplexity achieved on domain-specific test data using different settings. Different interpolation weights for the in-domain language model are tried. We can see the interpolated language model generally achieves best perplexities when the weight for in-domain language model is set to 0.5. Applying these interpolated language models for translation tasks in corresponding domains should help improving the quality of translation.

6 Conclusion

In this paper, we described the reordering model we applied in Weblio Pre-reordering SMT system, and also some efforts to utilize N-best parse trees and N-best reordering results. According to our in-house experiment results, the automatic evaluation scores are generally improved when multiple candidates of parse tree and reordering result are considered. However, in the human evaluation, incorporating N-best parse trees did not gain improvements.

As we demonstrated in Section 5.1, the reordering model is still unstable, and fails to work correctly even for some simple cases. Further improvement is required to enable the reordering

model to deal with general cases correctly. Then, in Section 5.2, we show interpolating general and in-domain language models can be a quick solution to improve translation quality when domain information is given as context.

For future research, we still plan to explore the performance limit of pre-reordering models. With a complex reordering model considers multiple factors of the language, it's still plausible for this approach to grow in performance. Also, as the pre-reordering model used in this paper is independent of specific language pair, more experiments can be conducted on different language pairs.

References

Isozaki, Hideki et al. (2010). "Head finalization: A simple reordering rule for sov languages". In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, pp. 244–251.

Klein, Dan and Christopher D Manning (2002). "Fast exact inference with a factored model for natural language parsing". In: *Advances in neural information processing systems*, pp. 3–10.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu (2003). "Statistical phrase-based translation". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 48–54.

Nakazawa, Toshiaki et al. (2014). "Overview of the 1st Workshop on Asian Translation". In: *Proceedings of the 1st Workshop on Asian Translation (WAT2014)*.

Neubig, Graham, Yosuke Nakata, and Shinsuke Mori (2011). "Pointwise prediction for robust, adaptable Japanese morphological analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, pp. 529–533.

Och, Franz Josef and Hermann Ney (2004). "The alignment template approach to statistical machine translation". In: *Computational linguistics* 30.4, pp. 417–449.

Petrov, Slav et al. (2006). "Learning Accurate, Compact, and Interpretable Tree Annotation". In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*. ACL-44. Sydney, Australia: Association for Computational Linguistics, pp. 433–440.

Zhu, Zhongyuan et al. (2014). "A preordering method using head-restructured CFG parse tree for SMT". In: *Proceedings of the 20th Annual Meeting of the Association for Natural Language Processing*, pp. 594–597.