NICT at WAT 2015

Chenchen Ding, Masao Utiyama, Eiichiro Sumita
Multilingual Translation Laboratory
National Institute of Information and Communications Technology
3-5 Hikaridai, Seikacho, Sorakugun, Kyoto, 619-0289, Japan
{chenchen.ding, mutiyama, eiichiro.sumita}@nict.go.jp

Abstract

Translation systems of our NICT team at *the 2nd Workshop on Asian Translation* (WAT 2015) are described in this paper. We participated in two translation tasks: Japanese-to-English (JE) and Korean-to-Japanese (KJ). A baseline phrased-based (PB) statistical machine translation (SMT) system in Moses was used. On JE translation, two pre-reordering approaches were applied: a simple *reverse preordering* and a dependency-based approach. On KJ translation, the processing was purely conducted on character-level. Evaluation results show that even simple approaches can improve JE and KJ PB SMT significantly. These techniques can be easily applied in practice because of the simplicity.

1 Introduction

Statistical machine translation (SMT) techniques have been well developed and widely applied in practice. Linguistic knowledge-free SMT frameworks, such as phrase-based (PB) SMT (Koehn et al., 2003) and hierarchical phrase-based SMT (HIERO) (Chiang, 2007), handle many translation tasks efficiently as long as sufficient training data prepared. Further, sophisticated syntactically-driven approaches (Neubig, 2013) give better performance than PB SMT and HIERO on difficult translation tasks (Neubig, 2014).

At *the 2nd Workshop on Asian Translation* (WAT 2015) (Nakazawa et al., 2015), our intention is to test the efficiency of several simple techniques for Japanese-to-English (JE) and Korean-to-Japanese (KJ) translation, specifically, pre-reordering approaches for JE translation and character-based processing for KJ translation. On JE translation, we found the simple *reverse preordering* approach proposed by Katz-Brown and Collins (2008) performed as well as a well-designed dependency-based approach, in improving a PB SMT baseline. Considering the simplicity of the *reverse preordering*, we think the approach should be used more widely for JE translation. On KJ translation, we found even a pure character-based approach outperformed the organizer's baseline a lot, due to the similarity of the two languages on their vocabularies and syntaxes. We give descriptions of the approaches in the following sections.

2 Pre-reordering for JE Translation

As Japanese and English have dramatically different word orders, the performance of *word reordering* affects translation results significantly. Among different lines of researches, *pre-reordering* has been widely applied in practice and still studied in recent researches (de Gispert et al., 2015; Hoshino et al., 2015).

For the JE translation task of WAT 2015, we test two pre-reordering approaches. The first one is the *reverse preordering* (REV-REO) proposed by Katz-Brown and Collins (2008) for the NTCIR-7 JE Patent MT translation task. Another one is a recently proposed dependency-based approach (DEP-REO) (Ding et al., 2015)[1] with well-designed rules. We select the two approaches because they are on two extremes, that REV-REO is an approach needs no syntactic analysis at all, while the DEP-REO makes a good use of the dependency structure of Japanese sentences. As both approaches have been described in detail in their original papers, We do not give repeated descriptions but just state several details in experiments.

For DEP-REO, the processes were completely identical to the experiments in Ding et al. (2015), where the tool chain of `MeCab`[2] and `CaboCha`[3]

[1] A non-refereed version in Japanese is Ding et al. (2014a).
[2] `http://taku910.github.io/mecab/`
[3] `http://taku910.github.io/cabocha/`

Proceedings of the 2nd Workshop on Asian Translation (WAT2015), pages 42–47,
Kyoto, Japan, 16th October 2015.

(Kudo and Matsumoto, 2002) based on IPA system for Japanese morphemes was used. For REV-REO, an important point is to **avoid the reordering across punctuations**[4]. In the experiments, we used four marks to compose the punctuation set: U+002C[5], U+FF0C[6], U+3001[7], and U+3002[8]. For the Japanese topic marker *wa*, which plays the key role of the approach, we did not judge it only by the surface form, but also referred to the specific tag *joshi, kakarijoshi*[9].

3 Character-based KJ Translation

As Korean and Japanese share so many similar features, we tried a purely character-based approach in WAT 2015. The process was identical to Ding et al. (2014b). Specifically, no morphological analysis or text normalization[10] were conducted except (unicode) characters were separated using spaces. The original space is replaced by a <sp> tag and the original tab is replaced by a <tab> tag[11]. The processes were applied consistently on training and test sets.

We found even the above-mentioned trivial process led to satisfactory performance on KJ translation. We further found a post-processing of bracket balancing (because the data contain many brackets) could give a slight improvement in performance. We will describe the process in the following Section 4.

4 Experiment and Evaluation

We used the PB SMT system in Moses[12] (Koehn et al., 2007) for JE and KJ translation tasks. Basically, we used identical settings as the organizer used in the baseline. However, there were several differences as follows.

- We used SRILM[13] (Stolcke, 2002) for language model training (interpolated modified Kneser-Ney discounting; 5-gram on English for JE translation and 9-gram on Japanese for KJ translation).
- We used MeCab (IPA) and CaboCha to process Japanese sentences in JE translation.
- We used no tools for Korean and Japanese morphological analysis in KJ translation, instead, the *max-phrase-length* were set to 9 in translation model training.

We selected the optimal distortion limit (DL) in PB SMT decoding by indoor experiments[14] and used the selected setting in the final submissions.

Table 1 shows the experimental results of DEP-REO and REV-REO on JE devtest set. The excellent performance of REV-REO is impressive. However, REV-REO needs a proper DL to reach its best performance, while DEP-REO has a more

	DL	BLEU	RIBES
BASELINE	20	16.95	.6356
+DEP-REO	0	17.77	.6512
	3	17.54	.6520
	6	**17.80**	**.6545**
	9	17.60	.6488
	12	17.79	.6497
	15	17.74	.6499
+REV-REO	0	15.16	.6556
	3	16.08	.6586
	6	17.58	.6707
	9	**18.02**	**.6751**
	12	17.78	.6733
	15	17.54	.6691

Table 1: Devtest set BLEU score and RIBES on JE translation.

	DL	BLEU	RIBES
+Lex.-Reo.	0	66.79	.9222
	3	66.64	.9221
	6	66.80	.9228
−Lex.-Reo.	0	66.74	.9221
+ Bracket Balanc.	0	**66.98**	**.9224**

Table 2: Devtest set BLEU score and RIBES on KJ translation (morpheme level, by MeCab).

[4] otherwise the reordering will become excessive.

[5] i.e., the ordinary comma.

[6] "fullwidth comma", the Chinese comma.

[7] "ideographic comma", the Japanese *tōten*.

[8] "ideographic full stop", the Japanese *kuten*.

[9] Because the DEP-REO is totally based on the IPA system, we also used the system for REV-REO. Actually 100% of the surface form *wa* were tagged as *joshi, kakarijoshi* by MeCab in our experiments.

[10] We only introduce the minimum rewriting to replace the "|", "[", "]" to full-width characters for Moses' decoder.

[11] The spaces mainly appeared on the Korean side due to its orthography. Those occasional spaces on the Japanese side were also replaced with tags.

[12] http://www.statmt.org/moses/

[13] http://www.speech.sri.com/projects/srilm/

[14] In KJ translation, we measured the results on morpheme-level by applying MeCab on outputs (after <sp> and <tab> tags recovered).

	Local Evaluation		Organizer Evaluation		
	BLEU	**RIBES**	**BLEU**	**RIBES**	**HUMAN**
BASELINE $_{\text{organizer, DL=20}}$	–	–	18.45	.6451	–
BASELINE $_{\text{indoor, DL=20}}$	18.09	.6435	18.09	.6397	–
BASELINE $_{\text{indoor}}$ + DEP-REO $_{\text{DL=6}}$	18.99	.6644	18.98	.6599	+16.000
BASELINE $_{\text{indoor}}$ + REV-REO $_{\text{DL=9}}$	18.96	.6870	18.96	.6845	+6.500

Table 3: Evaluation of our submission on JE translation compared with the organizer's PB SMT baseline.

	Local Evaluation		Organizer Evaluation		
	BLEU	**RIBES**	**BLEU**	**RIBES**	**HUMAN**
BASELINE $_{\text{organizer, DL=0, +Lex.-Reo.}}$	–	–	69.73	.9408	–
BASELINE $_{\text{indoor, DL=0, −Lex.-Reo.}}$	70.94	.9428	70.92	.9427	+8.250
+ Bracket Balancing	71.12	.9429	71.11	.9429	+10.500

Table 4: Evaluation of our submission on KJ translation compared with the organizer's PB SMT baseline.

stable performance across different DLs. The phenomenon is in agree to Ding et al. (2015).

Table 2 shows the experimental results on KJ translation results. We tested different DLs of 0, 3, and 6 with the lexicalized orientation reordering model (+Lex.-Reo.). The performance has only quite slight changes under different DLs. We also tested the monotone translation (DL $= 0$) without reordering model (−Lex.-Reo.). The change on performance is still insignificant. So a pure monotone translation is enough for KJ and a reordering model helps little. The phenomenon is in agree to Ding et al. (2014b).

We have observed there are many brackets in the data of KJ translation task. The translations of brackets are not consistent in training data and PB SMT cannot handle bracket pairs well in decoding. We used a simple post-processing for bracket balancing according to the following steps.

1. Getting $1,000$-best list for each output[15];
2. Selecting the m-th candidate, where m is $\min(\arg\min_n |\#\mathtt{L}_n - \#\mathtt{R}_n|)$; $\#\mathtt{L}_n$ and $\#\mathtt{R}_n$ are counts of "(" and ")" in the n-th candidate;
3. Inserting untranslated source-side ")" to the selected candidate after the translated parts of its preceding character[16], when
   (a) its paired "(" on source side is translated to a "(" on target side;
   (b) it has no paired "(" on source side but follows numbers / alphabets.

The described brackets balancing brought a gain about +0.2 BLEU scores on devtest set, which is larger than the effect of DL and reordering models. We consider specific post-processing will improve KJ translation more.

The evaluation results of our submission are listed in Table 3 and Table 4. Our local evaluation on automatic measures had slight but not significant differences compared with the organizer's in cases. On JE translation, our baseline was a little lower than the organizer's baseline, as the experimental settings were not totally identical to the organizer's ones, we think the difference is acceptable. Both REV-REO and DEP-REO improved the baseline (ours) approximately one point on BLEU score, but REV-REO gave a larger improvement on RIBES. On KJ translation, the listed scores are all based on the `MeCab`'s analysis. Our baseline, i.e., a character-based one, outperformed the organizer's baseline more than one BLEU score and the bracket balancing still gave a further improvement around +0.2 BLEU scores.

As to the human evaluations, our approaches still have stable improvement. On JE translation, the DEP-REO has a more obvious improvement than REV-REO, although the BLEU scores of the two approaches are nearly the same. We consider the using of specific syntactic information in DEP-REO brings benefits in human evaluation. On KJ translation, the automatic and human evaluations have consistent results, that our character-based

[15] We used the *distinct* options of Moses, so there were less than $1,000$ candidates.

[16] based on the alignment information given by `Moses`.

baseline performs better than organizer's baseline and post-processing gives further improvement.

5 Discussion

From the evaluation results, we have observed that simple (or, naïve) approaches can give satisfactory improvement for a PB SMT baseline. We show examples of REV-REO and DEP-REO in Fig. 1 and Fig. 2, respectively. JE and KJ translation examples are shown in Table 5 and Fig. 3, respectively.

On JE translation, in our opinion, the REV-REO approach should be used as a new baseline in future, due to its simplicity and efficiency. The REV-REO only needs morphological analysis, which is needed after all for a general SMT task. As the Japanese topic marker *wa* is available across different POS systems[17], the REV-REO is actually an approach with strong ability of generalization[18].

On KJ translation, we illustrated character-based processing led to good performance due to the similarity of the two languages. Actually, our approach is more like a transliteration process rather than a translation process. Although an SMT system gives satisfactory performance on KJ translation, we would like to state several issues for KJ SMT in practice.

- Although the syntaxes are similar between Korean and Japanese, there are differences in collocations of verbs and postpositions (case markers)[19]. Specific process or stronger models are needed for correct translation if such a collocation is over a long-range.

- Negation is purely realized by suffixes[20] in Japanese, but can be realized by both suffixes[21] and prefixes[22] in Korean. So, reordering is needed when a Korean negative prefix is translated into Japanese, unless we have a translation table covering all the negation forms of all the verbs. Specific process is also needed for this phenomenon.

- Specific named entity recognition / translation modules are needed for correct translation of proper nouns.

6 Conclusion

We have described the translation systems of NICT team for JE and KJ translation task at WAT 2015). Although the approaches we used are very simple, their efficiency has been proved by the evaluation. We expect these techniques to be more widely applied in the community of Asian NLP.

References

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2015. Fast and accurate preordering for SMT using neural networks. In *Proc. of NAACL-HLT*, pages 1012–1017.

Chenchen Ding, Keisuke Sakaushi, Hirona Touji, and Mikio Yamamoto. 2014a. Dependency tree-based pre-reordering rules for statistical Japanese-to-English machine translation. In *Proc. of ANLP*, pages 963–966. (In Japanese).

Chenchen Ding, Masao Utiyama, Mitsuo Yoshida, and Mikio Yamamoto. 2014b. Model learning form parallel documents: Korean-japanese smt. In *Proc. of ANLP*, pages 820–823. (In Japanese).

Chenchen Ding, Keisuke Sakanushi, Hirona Touji, and Mikio Yamamoto. 2015. Inter-, intra-, and extra-chunk pre-reordering for statistical Japanese-to-English machine translation. *ACM Transactions on Asian Language Information Processing*, x(x):x. (Accepted, to appear).

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, Katsuhiko Hayashi, and Masaaki Nagata. 2015. Discriminative preordering meets Kendall's Tau maximization. In *Proc. of ACL (Short Papers)*, pages 139–144.

Jason Katz-Brown and Michael Collins. 2008. Syntactic reordering in preprocessing for Japanese → English translation: MIT system description for NTCIR-7 patent translation task. In *Proc. of NTCIR*, pages 409–414.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of HTL-NAACL*, pages 48–54.

[17] Of course, the specific tag is different.

[18] We believe (although we have not done experiments) the REV-REO should work for Korean-to-English translation task as well because Korean has a topic marker *(n)eun* which is very similar to Japanese *wa*.

[19] Here are examples for some common verbs. Japanese *noru* and Korean *tada*, both have the meaning of *to ride*; *noru* requires a dative marker *ni* but *tada* requires an accusative marker *(r)eul* (the equivalent Japanese accusative marker is *wo*). Japanese *naru* and Korean *toeda*, both have the meaning of *to become*; *naru* requires a dative marker *ni* but *toeda* requires a nominative marker *i / ga* (the equivalent Japanese nominative marker is *ga*).

[20] Analyzed as auxiliary verbs, e.g., *nai*, *nu*, *mai*, etc.

[21] Analyzed as auxiliary verbs, e.g., *anta*, *anida*, etc.

[22] Analyzed as adverbs, e.g., *an* and *mot*.

[24] For convenience, we just use *kanji* here instead of *hanja*.

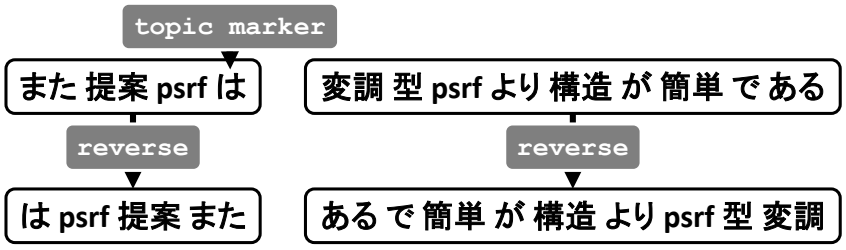

Figure 1: Example of REV-REO. The original Japanese sentence at the top is segmented after the topic marker and the morphemes within each segment are reversed.

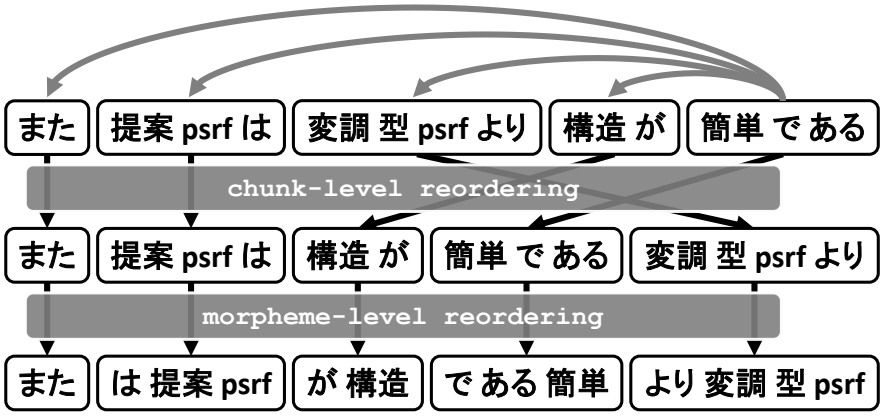

Figure 2: Example of DEP-REO. The original Japanese sentence at the top is reordered on both chunk- and morpheme-level based on its dependency structure.

BASELINE	*the proposed psrf psrf modulation type than the simple structure*
REV-REO	*the proposed psrf is simple structure than psrf modulation type*
DEP-REO	*the proposed psrf structure is simpler than psrf modulation type*
REFERENCE	*and , the proposed psrf has simpler structure than that of modulated psrf*

Table 5: JE translation examples. The inputs for BASELINE, REV-REO, and DEP-REO are the original Japanese sentence at the top of Fig. 1 (and Fig. 2), reordered Japanese sentence at the bottom of Fig. 1, and reordered sentence at the bottom of Fig. 2, respectively.

Figure 3: KJ translation example on a part of a Korean sentence. The gray blocks show the spaces used in Korean orthography. The characters[24] above hanguls show the Sino-Korean morphemes. The Japanese sentence at the bottom is the output by the character-level translation; the alignment between input and output is also shown. The output is nearly identical to the reference translation except an insignificantly redundant *tōten* (underlined).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*, pages 177–180.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc. of CoNLL*, pages 63–69.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi, and Eiichiro Sumita. 2015. Overview of the 2nd workshop on Asian translation. In *Proc. of WAT*.

Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. of ACL (Conference System Demonstrations)*, pages 91–96.

Graham Neubig. 2014. Forest-to-string SMT for Asian language translation: NAIST at WAT 2014. In *Proc. of WAT*, pages 20–25.

Andreas Stolcke. 2002. SRILM–an extensible language modeling toolkit. In *Proc. of ICSLP 2002*, pages 901–904.