The POSTECH Statistical Machine Translation Systems for NTCIR-7 Patent Translation Task

Jin-Ji Li[0], Hwi-Dong Na[0], Hankyong Kim[*], Chang-Hu Jin[*], and Jong-Hyeok Lee[0]

[0]*Department of Computer Science and Engineering,*

Electrical and Computer Engineering Division,

[*]*Graduate School for Information Technology,*

Pohang University of Science and Technology (POSTECH),

San 31 Hyoja Dong, Pohang, 790-784, R. of Korea

E-mail: {ljj, leona, arch, hchchh, jhlee}@postech.ac.kr

Abstract

This paper describes the POSTECH statistical machine translation (SMT) systems for the NTCIR-7 patent translation task. We entered two patent translation subtasks: Japanese-to-English (KLE-je), and English-to-Japanese translation (KLE-ej). The baseline systems are derived from a common phrase-based SMT framework. In addition, for Japanese-to-English translation, we adopted two kinds of methods. The first method is a word reordering model for preprocessing that reorders the words in the source sentence similarly to the order of target sentence before the decoding phase. The second is a cluster based model using syntactic information of Japanese sentences.

Keywords: phrase-based SMT, word reordering model, cluster based model.

1. Introduction

We participated in the patent translation task for the NTCIR-7 workshop. We prepared two MT systems for two subtasks: Japanese-to-English and English-to-Japanese. Our baseline systems are derived from a phrase-based SMT framework, Moses[1], an open source toolkit commonly used for experimentations in the community. For the English-to-Japanese subtask, we only applied the baseline SMT system. For the Japanese-to-English subtask, we applied two linguistically motivated techniques.

The first method is the word reordering model for preprocessing. Japanese and English belong to linguistically distant language families, and their word ordering in a sentence is very different. To account for this difference, we apply a word reordering model to Japanese sentences as a preprocessing step based on syntactic information, so that word order of source sentences will be more like that of target sentences.

The second method is the cluster based model using Japanese syntactic information. Usually sentences with similar syntactic structures yield similar distributions of n-grams reflecting their word order. Therefore, cluster specific language models can benefit SMT. We clustered the training corpus according to the linguistic patterns of the source language, and built SMT systems based on each cluster. Each cluster-based model is also built on the common phrase-based SMT framework.

In Section 3, we will describe the word reordering for preprocessing and the cluster-based model will be presented in Section 4.

2. Corpus profile and baseline system

Patent parallel corpus (PPC) was prepared for the NTCIR-7 patent translation task [12, 13]. The PPC contains sentence-aligned Japanese-English parallel patent data that can be used for training and developing MT systems. We only use PPC-1 (first released version) as our training and development corpus.

After cleaning the corpus with the script provided by Moses, we achieve 1,172,709 and 609 source sentences as the training and development corpus, respectively. We set the maximum length of each sentence at 40. For the formal run, the test corpus size is 1,381. A series of alphanumeric characters are grouped into one word in the Japanese corpus and recovered to their original forms after the translation. English words are lower-cased and recovered after the translation using the recaser script provided by Moses. Japanese sentences are tokenized and parsed by the CaboCha parser[2]. The detailed corpus statistics are shown in Table 1.

	Training corpus	
	Japanese	English
Number of words	30,761,076	28,683,697
Number of singletons	131,219	131,321
Average length	26.23	24.46
	Development corpus	
	Japanese	English

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

[2] http://chasen.org/~taku/software/cabocha/

Number of words	15,997	14,818
Number of singletons	2,697	2,817
Average length	26.27	24.33
	Test corpus	
	Japanese	English
Number of words	48,278	44,910
Number of singletons	4,088	4,273
Average length	34.96	32.52

Table 1. The detailed statistics of training, development, and test corpora.

The baseline systems are built using Moses with the default setting and evaluated by the NTCIR-7 scoring tools[3]. For the English-to-Japanese translation, we did not perform the re-tokenizing step after removing all white spaces of the translation result. The Bleu score is 24.48 and 30.06 for the Japanese-to-English and English-to-Japanese translation, respectively[4].

Sections 3 and 4 describe the techniques applied only in the Japanese-to-English direction.

3. Reordering model as preprocessing

There are many previous works proposing word reordering methods in the preprocessing phase to improve the performance of the phrase-based SMT system [1, 2, 3, 4, 5, 6 and 7]. These methods reorder the words of the source language before translation so they are similar to the word order of the target language. These methods can be a complement to a phrase-based SMT system which uses a relatively simple distortion model in the decoding phase.

3. 1. System overview

Japanese and English belong to different language families: SOV and SVO, respectively. To reorder the Japanese word order more effectively, we use a dependency tree which is marked with head-relative position information rather than a flat sequence of surface strings. First, we parse Japanese sentences. Then we reconstruct the Japanese dependency trees by applying a series of reordering rules to each node recursively. Finally, we recover the surface strings from the reconstructed dependency trees and run the Moses. The reordering rules are applied to the training, development, and test corpus before translation.

For example, a Japanese input sentence is as follows and the dependency tree is given in Figure 1.

Before reordering:
スキャナ/ー部/は　　原稿/載置台　　２８/および
スキャナ/ーユニット　２９/を　備え/ている

After applying the reordering rules, we get the reordered Japanese sentence from the reconstructed dependency tree. Because it has the head-relative position information, we can easily recover the surface string. The surface string after reordering is shown as follows and Figure 2 shows the related dependency tree.

After reordering:
スキャナ/ー部/は　　備え/ている　　原稿/載置台/
２８/および　　スキャナ/ーユニット　　２９/を

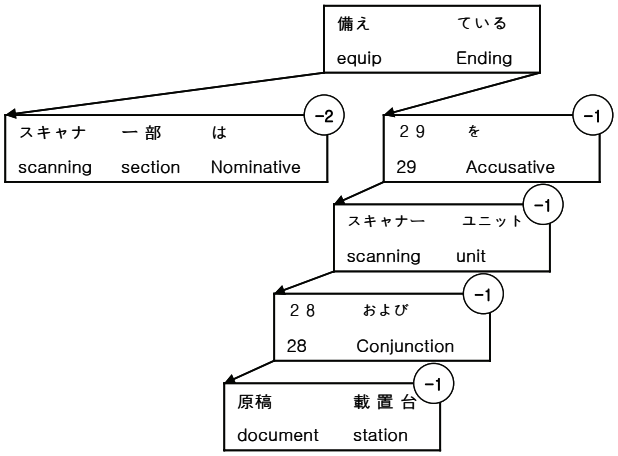

Figure 1. A dependency tree of a Japanese sentence with head-relative position information.

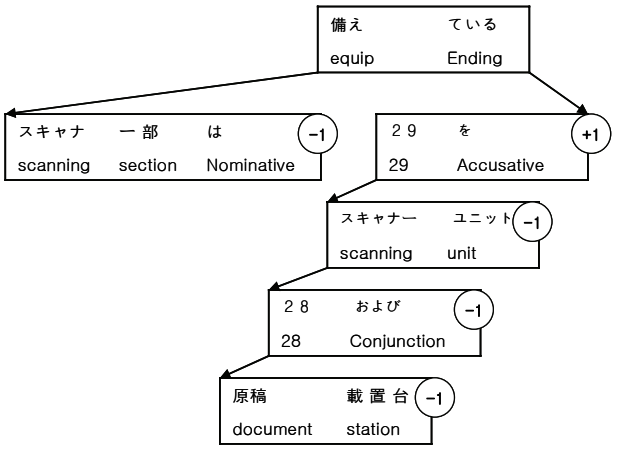

Figure 2. A dependency tree of a Japanese sentence after reordering.

In Figure 1, the predicate (備えている) governs the nominative (スキャナー部は) *bunsetsu* and the accusative(２９を). If the accusative *bunsetsu* moves from the left to the right side of the predicate as shown in Figure 2, then we can obtain a Japanese sentence following the English word order from the reconstructed dependency tree. The reordering rules consist of a head and several arguments which are the direct child nodes of the head. The child nodes such as adverbs which are not critical in determining word order are ignored. From the training corpus, we extracted reordering rules which satisfy the above condition and have a frequency of over 100,000. As a result, 29 rules are extracted and applied to the source sentences as shown in Table 2. The constituents of LHS and RHS in reordering rules are co-

[3] http://www.mibel.cs.tsukuba.ac.jp/~norimatsu/bleu_kit/

[4] This is not the official result. The official result is 24.49 and 26.93, respectively [14].

indexed and the parenthesized part indicates the head. The RHS column shows the reordering results after applying the reordering rules.

The average matching ratio of reordering rules is about 2 times per sentence. This is the first method.

LHS	RHS
を$_0$. (動詞-自立$_1$)	(1) 0
の$_0$. (を$_1$)	(1) 0
に$_0$. (動詞-自立$_1$)	(1) 0
の$_0$. (により$_1$)	(1) 0
の$_0$. (の$_1$)	(1) 0
の$_0$. (に$_1$)	(1) 0
の$_0$. (名詞-数$_1$)	(1) 0
の$_0$. (名詞-一般$_1$)	(1) 0
は$_0$. を$_1$. (動詞-自立$_2$)	0 (2) 1
は$_0$. に$_1$. (動詞-自立$_2$)	0 (2) 1
を$_0$. に$_1$. (動詞-自立$_2$)	(2) 0 1
に$_0$. を$_1$. (動詞-自立$_2$)	(2) 1 0
の$_0$. (で$_1$)	(1) 0
が$_0$. に$_1$. (動詞-自立$_2$)	0 (2) 1
で$_0$. (動詞-自立$_1$)	(1) 0
の$_0$. (から$_1$)	(1) 0
から$_0$. (動詞-自立$_1$)	(1) 0
に$_0$. (動詞-自立$_1$)	(1) 0
に$_0$. が$_1$. (動詞-自立$_2$)	1 (2) 0
について$_0$. (動詞-自立$_1$)	(1) 0
の$_0$. (について$_1$)	(1) 0
は$_0$. の$_1$. (ある$_2$)	0 (2) 1
の$_0$. (は$_1$)	(1) 0
の$_0$. (が$_1$)	(1) 0
により$_0$. が$_1$. (動詞-自立$_2$)	1 (2) 0
の$_0$. (には$_1$)	(1) 0
で$_0$. を$_1$. (動詞-自立$_2$)	(2) 0 1
が$_0$. を$_1$. (動詞-自立$_2$)	0 (2) 1
は$_0$. で$_1$. (動詞-自立$_2$)	0 (2) 1
において$_0$. が$_1$. を$_2$. (動詞-自立$_3$)	1 (3) 2 0
は$_0$. を$_1$. に$_2$. (動詞-自立$_3$)	0 (3) 1 2

Table 2. List of reordering rules of the first method.

The second method is similar to the first one, but we only consider a head, and one argument from the child nodes. It is a more generalized method than the first one. This kind of rule indicates whether the argument should be moved from the left to the right side of the head or not. Finally, we choose 14 rules for the second method as Table 3 shows.

Dependent	Head
を	動詞-自立
に	動詞-自立
で	動詞-自立
も	動詞-自立
に対して	動詞-自立
には	動詞-自立
において	動詞-自立
について	動詞-自立
における	動詞-自立
により	動詞-自立
によって	動詞-自立
の	*
から	動詞-自立
動詞-自立	動詞-自立

Table 3. List of reordering rules of the second method. The asterisk indicates any arbitrary head.

3. 2. Experimental results

The experimental results of word reordering models for preprocessing are presented in Table 4.

Method	Bleu
Baseline	24.48
First Method	24.21
Second Method	23.45

Table 4. The Bleu values when reordering models are applied.

Using the first method of reordering models, the performance is slightly lower than the baseline system. Through human error analysis we find that the proposed method can capture the long-distance reordering effectively in a simple sentence, but it does not work well when the source sentence is a complex sentence. The reason for this is that our reordering rules do not include the reordering rules between 動詞-自立 s because of high variations. The reordering rules between 動詞-自立 s are left as our future work.

The second method which is a generalized version of the first method obtained an even poorer result than the first method. The reason may be that the reordering rules of the second method are excessively applied to the source language.

4. Cluster based model

In our model, a cluster based model indicates a model with clustered language modeling. A translation model (TM) and a language model (LM) are two main components in SMT systems. Language modeling is a long-established issue in natural language processing (NLP). There are a number of papers about clustered language modeling in SMT systems that report the effectiveness of this technique [8, 9, 10 and 11].

Usually, sentences with similar syntactic structures yield similar distributions of n-grams reflecting their word order. Therefore, cluster specific language models can benefit SMT systems. Under the assumption that source sentences with similar syntactic structures can be translated into similar styles, our clustering method is based on linguistic patterns of the source language.

To decide the cluster types, we first parse the Japanese sentences using the CaboCha parser. We define

some subtree structures in the source dependency trees which ignore the adjuncts as cluster types. In the example in Figure 1, the syntactic pattern is 'は. を. (動詞-自立)'. This pattern describes a cluster of sentences whose dependency tree includes a predicate and a nominative and an accusative *bunsetsu* as child nodes.

From all syntactic patterns, we only select 27 cluster types whose frequency is greater than 10,000 in the training corpus. The list of linguistic patterns is presented in Table 5.

Cluster type	Freq.
を. (動詞-自立)	407,629
に. (動詞-自立)	246,188
が. (動詞-自立)	143,579
動詞-自立. (動詞-自立)	134,566
は. に. (動詞-自立)	81,434
は. を. (動詞-自立)	79,717
は. (ある)	63,646
を. に. (動詞-自立)	59,294
に. を. (動詞-自立)	53,438
と. (動詞-自立)	39,354
は. (動詞-自立)	36,164
が. に. (動詞-自立)	32,751
で. (動詞-自立)	32,576
は. が. (動詞-自立)	30,639
から. (動詞-自立)	26,782
に. が. (動詞-自立)	25,065
について. (動詞-自立)	24,044
は. の. (ある)	22,509
は. と. (動詞-自立)	16,106
により. が. (動詞-自立)	14,826
で. を. (動詞-自立)	14,274
が. を. (動詞-自立)	13,822
の. (動詞-自立)	13,201
は. で. (動詞-自立)	11,562
も. (動詞-自立)	11,007
を. と. (動詞-自立)	10,910
は. を. に. (動詞-自立)	10,310

Table 5. Cluster type list.

4. 1. System overview

The system overview is as follows.

1. Predict clusters according to cluster types
2. Translate using baseline SMT system
3. Optimize LM integration parameters
4. Re-translate using general + cluster-specific LM
5. Select best translation result

Given an input source sentence, we first predict the cluster according to the proposed linguistic patterns. In this phase we allow multiple matching. In other words, an input sentence can belong to several clusters. The best translation result will be chosen in step 5. The average matching ratio is about 1.73 in the development corpus.

Then, the input sentence will be translated using the baseline SMT system. Here, we call the LM in the baseline system the general LM.

To optimize the LM integration parameters we use the same method proposed by Yamamoto and Sumita [9], that the cluster specific LM is used as an additional feature in the log-linear combination. The integration of general and cluster specific LMs were tuned on the development corpus and the sum of these LM weights is equal to the weight in the baseline system.

After re-translating using the integrated version of general and cluster specific LMs, finally we select the best translation result using perplexity as a measure. The translation result with the lowest perplexity value will be selected.

4. 2. Experiment results

We conducted two kinds of experiments. First, we optimized the integration parameter of the general and cluster-specific LM. We also studied the effect of training corpus size in the cluster based model. The experimental results are given in Table 6 and 7. The results reported in this paper are different from the formal run results because we could not submit them on time.

General LM	Bleu
Baseline	24.48
(1 - λ) * General LM + λ * Cluster-specific LM	**Bleu**
λ = 0.1	24.67
λ = 0.2	24.54
λ = 0.3	24.52
λ = 0.4	24.44
λ = 0.5	24.28
λ = 0.6	24.25
λ = 0.7	23.99
λ = 0.8	23.76
λ = 0.9	23.59

Table 6. The optimized parameter when integrating general and cluster-specific LM.

Training corpus size	Baseline	Cluster-based
50k	21.48	22.14
100k	22.55	22.91
300k	23.46	23.74
All	24.48	24.67

Table 7. The Bleu values when the training corpus size is different.

For each corpus of a different size, we optimized the integration parameters respectively.

From Table 7, we find that the difference between the baseline and the cluster based system becomes smaller as the training corpus size grows. When the training

corpus size is 50k, the improvement is statistically significant. However, when using the original training corpus, the improvement is small and it is not statistically significant anymore.

The cluster-based model does not work as well as we expected. We analyzed the results from several aspects. First, we cautiously infer that the cluster-based model does not work better than the baseline SMT system when using a large scale corpus. The reason may be that the general LM already has enough information because of the large size of the training corpus. Secondly, the linguistic patterns which we used as cluster types are relatively simple; however, the sentences in the parallel corpus are very long. Hence, multiple matching occurs more easily, which means a sentence can belong to several clusters.

There are still several issues that we consider as our future work: How to determine the cluster types which are better for large scale training corpus? How to set the matching priority when there is multiple matching?

5. Conclusion

For the NTCIR-7 patent translation task, we focused on the Japanese-to-English direction and proposed a word reordering model and a cluster based model. Both approaches were built on the framework of a common phrase-based SMT system.

We investigated two kinds of word reordering models and a cluster based model which are based on syntactic information of the source language. The performance of the word reordering method is slightly lower than the baseline system. We expected the human evaluation result of the reordering model to reflect the effectiveness of our method. Unfortunately, we did not submit the results of our reordering model in time and could not verify the effectiveness through human evaluation.

For the cluster based model, we found an interesting result, that when the same method was applied to a small size corpus, it worked better than when applied to a large size one. To the best of our knowledge, our experimentation is the first to apply the cluster-based model to a large-sized corpus. Further study is required for verification.

6. Acknowledgement

This work was supported in part by MKE & IITA through the IT Leading R&D Support Project and also in part by the BK 21 Project in 2008.

References

[1] Michael Collins, Philip Koehn, and Ivona Kučerová, *Clause restructuring for statistical machine translation*, In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, 2005

[2] Fei Xia and Michael McCord, *Improving a statistical MT system with automatically learned rewrite patterns*. In Proceedings of the 20th international Conference on Computational Linguistics, 2004

[3] Chao Wang, Michael Collins, Philip Koehn, *Chinese Syntactic Reordering for Statistical Machine Translation* Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.737-745, 2007

[4] Deepa Gupta, Mauro Cettolo, and Marcello Federico. *POS-based reordering models for statistical machine translation*. In Proceedings of Machine Translation Summit XI, pp. 207-213, 2007

[5] Kay Rottmann and Stephan Vogel. *Word reordering in statistical machine translation with a POS-based distortion model*; Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation, 2007

[6] Yuqi Zhang, Richard Zens, and Hermann Ney: *Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation*. Workshop at NAACL-HLT 2007 "Syntax and structure in statistical translation", pp.1-8, 2007

[7] Yuqi Zhang, Richard Zens, Hermann Ney: *Improved chunk-level reordering for statistical machine translation*. In Proceeding of International Workshop on Spoken Language Translation, 2007

[8] Sasa Hasan, Hermann Ney. *Clustered Language Models based on Regular Expressions for SMT*, 2005, EAMT

[9] Hirofumi Yamamoto, Eiichiro Sumita. *Bilingual Cluster Based Models for Statistical Machine Translation*, 2007, EMNLP

[10] Matthias Eck, Stephan Vogel, Alex Waibel. *Language Model Adaptation for Statistical Machine Translation based on Information Retrieval*, 2004, LREC

[11] Bing Zhao, Matthias Eck, Stephan Vogel. *Language Model Adaptation for Statistical Machine Translation with Structured Query Models*, 2004, COLING

[12] Masao Utiyama and Hitoshi Isahara. A Japanese-English patent parallel corpus. MT Summit XI, 2007

[13] Masao Utiyama, Mikio Yamamoto, Atsushi Fujii, and Takehito Utsuro. Description of Patent Parallel Corpus for NTCIR-7 Patent Translation Task. http://if-lab.slis.tsukuba.ac.jp/fujii/ntc7patmt/ppc.pdf

[14] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, Takehito Utsuro. Overview of the Patent Translation Task at the NTCIR-7 Workshop. Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access, 2008.