NTT-NII Statistical Machine Translation for NTCIR-10 PatentMT

Katsuhito Sudoh Jun Suzuki Hajime Tsukada Masaaki Nagata
NTT Communication Science Laboratories
{sudoh.katsuhito, suzuki.jun, hajime.tsukada, masaaki.nagata}@lab.ntt.co.jp

Sho Hoshino
The Graduate University for Advanced Studies
hoshino@nii.ac.jp

Yusuke Miyao
National Institute of Informatics
yusuke@nii.ac.jp

ABSTRACT

This paper describes details of the NTT-NII system in NTCIR-10 PatentMT task. The system is an extension of the NTT-UT system in NTCIR-9 by: a new English dependency parser (for EJ task), a syntactic rule-based pre-ordering (for JE task), a syntax-based post-ordering (for JE task). Our system ranked 1st in EJ subtask both in automatic and subjective evaluation, and was the best SMT system in JE subtask.

Team Name

NTITI

Subtasks

Japanese-to-English, English-to-Japanese

Keywords

pre-ordering, post-ordering, system combination

1. INTRODUCTION

Statistical machine translation (SMT) is a promising way for machine translation in domains in which large-scale bilingual language resources are available. The NTCIR-10 PatentMT task [2] provides millions of parallel sentences for translation of patent documents. We NTT-NII group (ID: `NTITI`) participated in Japanese-to-English and English-to-Japanese subtasks with SMT systems.

Our SMT systems are based on our NTCIR-9 systems with GMBR system combination [12] but include our recent research progresses: syntax-based pre-ordering [4] and post-ordering [14] for the Japanese-to-English subtask, the use of English dependency parsing [15] for Head Finalization [5] for the English-to-Japanese subtask. Each individual SMT system was trained with supplied bilingual and monolingual resources and used Moses[1] as the decoder.

The remainder of this paper is organized as follows. Section 2 briefly reviews GMBR system combination [1]. Section 3 and 4 describes our system for Japanese-to-English and English-to-Japanese subtasks, respectively. Finally, section 5 concludes the paper.

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

2. GENERALIZED MINIMUM BAYES RISK SYSTEM COMBINATION

This section briefly present our GMBR system combination. Please refer to our paper [1] for details. Note that our system combination only picks one hypothesis from an N-best list and does not generate a new hypothesis by mixing partial hypotheses among the N-best.

2.1 Theory

Minimum Bayes Risk (MBR) is a decision rule to choose hypotheses that minimize the expected loss (i.e. *Bayes Risk*). In the task of SMT from a French sentence (f) to an English sentence (e), MBR decision rule on $\delta(f) \rightarrow e'$ with the loss function L over the possible space of sentence pairs ($p(e, f)$) is denoted as:

$$\operatorname*{argmin}_{\delta(f)} \sum_{e} L(\delta(f)|e)p(e|f) \tag{1}$$

In practice, we approximate this using N-best list $N(f)$ for the input f.

$$\operatorname*{argmin}_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e)p(e|f) \tag{2}$$

Although MBR works effectively for re-ranking single system hypotheses, it is challenging for system combination because the estimated $p(e|f)$ from different systems cannot be reliably compared. One practical solution is to use uniform $p(e|f)$ but this does not achieve Bayes Risk. GMBR corrects by parameterizing the loss function as a linear combination of sub-components using parameter $\boldsymbol{\theta}$:

$$L(e'|e;\boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k L_k(e'|e) \tag{3}$$

For example, suppose the desired loss function is "1.0−BLEU". Then the sub-components could be "1.0−precision(n-gram) ($1 \leq n \leq 4$)" and "brevity penalty".

Assuming uniform $p(e|f)$, the MBR decision rule can be denoted as:

$$\begin{aligned} &\operatorname*{argmin}_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) \frac{1}{|N(f)|} \\ &= \operatorname*{argmin}_{e' \in N(f)} \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e'|e) \end{aligned} \tag{4}$$

To ensure that the uniform hypotheses space gives the same decision as the original loss in the true space $p(e|f)$, we use a small development set to tune the parameter $\boldsymbol{\theta}$ as follows. For any two hypotheses e_1, e_2, and a reference translation e_r (possibly not in $N(f)$) we first compute the true loss: $L(e_1|e_r)$ and $L(e_2|e_r)$. If $L(e_1|e_r) < L(e_2|e_r)$, then we would want $\boldsymbol{\theta}$ such that:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) < \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \qquad (5)$$

so that GMBR would select the hypothesis achieving lower loss. Conversely if e_2 is a better hypothesis, then we want opposite relation:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) > \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \qquad (6)$$

Thus, we directly compute the true loss using a development set and ensure that our GMBR decision rule minimizes this loss.

2.2 Implementation

We implement GMBR for SMT system combination as follows.

First we run SMT decoders to obtain N-best lists for all sentences in the development set, and extract all pairs of hypotheses where a difference exists in the true loss. Then we optimize $\boldsymbol{\theta}$ in a formulation similar to a Ranking SVM [6]. The pair-wise nature of Eqs. 5 and 6 makes the problem amendable to solutions in "learning to rank" literature [3]. In this shared task, we used RIBES+BLEU as our objective functions, so that we want to choose the best translation hypotheses both in terms of local view (BLEU) and global view (RIBES). There is one regularization hyperparameter for the Ranking SVM, which we set by cross-validation.

The development set of each translation task consisted of 2,000 sentences; we divided it halves and used the first half for tuning SMT parameters by Minimum Error Rate Training (MERT) [10], and the other half for training the GMBR system combination.

3. JAPANESE-TO-ENGLISH SUBTASK

3.1 Run Configurations and Results

We submitted three runs for the Japanese-to-English subtask: one system combination result and two individual system results.

JECOMBINATION: System combination of three systems: JEPREORDER, JEPOSTORDER, and a baseline Moses phrase-based MT (JEPBMT). (run ID: `NTITI-je-1`)

JEPREORDER: Moses phrase-based MT with a syntax-based Japanese pre-ordering. (run ID: `NTITI-je-2`)

JEPOSTORDER: Moses monotone phrase-based MT followed by Moses-Chart syntax-based MT for post-ordering. (run ID: `NTITI-je-3`)

JEPREORDER and JEPBMT used a word 7-gram language model trained on supplied bilingual and monolingual (USPTO patent applications) corpora, and JEPOSTORDER used a word 6-gram language model trained only on the bilingual corpora[2]. The very large language model could not be trained by a simple application of SRILM `ngram-count` to the whole English corpora, so we first trained year-wise word 7-gram models and then linearly interpolated them with SRILM `ngram-merge`. Reordering limits for the individual systems were: 18 for JEPREORDER and JEPBMT, and unlimited (`max-chart-span=999`) for JEPOSTORDER.

Table 1: Intrinsic evaluation results in Japansese-to-English subtask.

System	BLEU	RIBES	Ave. Adequacy
System Combination			
JECOMBINATION	0.3255	0.7324	3.32
Individual Systems			
JEPREORDER	0.3079	0.6911	n/a
JEPOSTORDER	0.3129	0.7171	3.26
JEPBMT	0.3010	0.6796	n/a
Other teams' runs			
`EIWA-je-1`	0.3250	**0.7402**	3.53
`JAPIO-je-1`	0.2288	0.7214	**3.67**
`RWTH-je-1`	**0.3377**	0.7175	3.07
`RBMT1-1`	0.2035	0.7106	3.57
`BASELINE1-1`	0.2856	0.6972	2.81

Table 1 shows evaluation results of our Japanese-to-English systems. Our primary run by JECOMBINATION ranked second in BLEU and RIBES among all runs (including RBMT and hybrid systems), and showed the best adequacy score among all SMT runs (not including hybrid systems).

3.2 Pre-ordering

Conventional studies on pre-ordering in Japanese-to-English translation [8, 7, 12, 9] did not show so large improvements. We recently proposed a novel rule-based Japanese pre-ordering with chunk- and word-based reordering as follows (details in [4] in Japanese). First, we reorder Japanese chunks (*bunsetsu*) close to English word order based on case structures, given by a Japanese dependency and case structure analyzer KNP[3]. Basically we moved a predicate chunk between its subject and object chunks and other modifier chunks after their modifiers, to realize English-like word order, except coordinations identified by KNP. Then, we moved functional words in each chunk (e.g., case particles) in front of the chunk to emulate English prepositions.

We used a standard SMT procedure with Moses (phrase-based) in pre-ordered Japanese-to-English translation. Here we also applied lexicalized reordering for correcting *local* pre-ordering errors.

Table 2 shows translation examples of JEPREORDER compared with JEPBMT and the reference translations. The pre-ordered Japanese sentence in Example 2-1 is almost monotone with the reference and JEPREORDER shows fairly good translation result. Contrary, JEPBMT shows severe reordering errors. Our pre-ordering worked well when dependency and case structures are correct as in Example 2-1, but it is

[2]This was due to a wrong configuration in the formal run using an inappropriate `moses.ini`, which was tuned in our pilot test. So its post-evaluation results slightly differed from the official ones.

[3]http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KNP

largely affected by their errors. Example 2-2 shows an example with a miss of subject chunk in Japanese "トランジスタ Q2 , Q3 は ," (transistors Q2 and Q3). It should be placed before its predicate chunk "カスコード 接続 さ れて いる" but wrongly placed after the predicate as its modifier.

3.3 Post-ordering

Motivated by the success of Head Finalization in English-to-Japanese translation [5, 12], we proposed a *post-ordering* SMT framework in which Japanese is translated first into Head-Final English (HFE) and then reordered into English [13] by isolated processes. Our recent study has extended the HFE-to-English post-ordering by a syntax-based SMT [14].

The post-ordering-based system used two translation models: a Japanese-to-HFE phrase-based model and a HFE-to-English syntax-based model. HFE sentences for training the translation models and an HFE language model were derived from the English syntactic parse results on the training sentences. The Japanese-to-HFE translation was solved as a monotone phrase-based SMT problem with Moses, and the HFE-to-English translation (post-ordering) was solved as a target-syntax-based SMT problem with Moses-Chart with an unlimited reordering span.

Table 3 shows translation examples of JEPOSTORDER compared with JEPBMT and the reference translations. HFE sentence in Example 3-1 could be correctly reordered into English word order by the syntax-based post-ordering, as in JEPOSTORDER. JEPBMT failed to construct a meaningful sentence. JEPOSTORDER is an efficient approximation of Japanese-to-English syntax-based SMT and can capture English syntactic constraints for accurate reordering. On the other hand in Example 3-2, JEPOSTORDER failed to place a subject "it". This was mainly because the source Japanese sentence did not have an explicit subject word. Since the post-ordering process had a very limited word restoration ability – only inserting articles `a`, `an`, and `the`, JEPOSTORDER often faced this kind of problem.

3.4 Back-transliteration for Unknown Katakana Words

In the test set, we found not a few unknown words, most of which were written in Katakana. These Katakana words were transliterated English words so that we tried to back-transliterate them.

Our back-transliteration method was very simple; translate Katakana character sequences into roman alphabet sequences by Moses, with a character-based 5-gram language model and a character-based phrase table. We used a word translation table obtained in the training of JEPBMT to learn the character-based phrase table, and used the whole English sentences in supplied bitexts to learn the character-based language model. Since most of the word translation table entries were not transliteration pairs, we first removed word translation pairs whose Japanese-side includes non-Katakana characters and then we filtered out non-transliteration pairs by the same manner proposed by Sajjad *et al.* [11].

In the intrinsic evaluation test set, we found 122 unknown Katakana words[4] (70 unique words) and transliterated 90 words (50 unique words) successfully by this method.

[4]In these unknown words, two typographical errors were found: デルタル (`de-l-ta-l`) for "digital" and バイエルン (Bayern) for "Baier".

Table 4: Katakana-to-English transliteration examples. System outputs in *italic* are incorrect ones.

Source	System	Reference
デトネーション	detonation	detonation
ベンゾトリフルオライド	benzotrifluoride	benzotrifluoride
ディスクリートトラック ディスクタイプ	discrete track disk type	discrete track disk type
スピナードライバ	*spinner drive*	spinner driver
プリクラッシュ	*pre-crash*	pre-crush
アウターパーツ	*outer period*	outer parts

Table 4 shows transliteration examples. The first three examples are correct ones, and the latter three examples are incorrect ones. As also shown in the examples, most of unknown Katakana words were compound words; each element was appeared in bitexts but the Japanese tokenizer failed to tokenize such a long compound word correctly. Our transliteration method could transliterate many Katakana words correctly and helped to translate complex Japanese compound words in this subtask.

3.5 Post-Evaluation Results

We conducted additional experiments with different configurations after the official submission. In this post-evaluation experiment, we compared five different language models: baseline 5- and 6-gram models trained only using supplied bilingual corpora, and large 5-, 6-, and 7-gram models trained using supplied bilingual and monolingual corpora.

Results are shown in Table 5. Large language models worked to improve BLEU, but also decreased RIBES. It suggests that a simple application of such a large-scaled word n-gram language model does not help to improve reordering. This seems to be reasonable because a word n-gram language model provides local word order constraints.

4. ENGLISH-TO-JAPANESE SUBTASK

4.1 Run Configurations and Results

We submitted two runs for the English-to-Japanese subtask: one system combination result and one individual system result.

EJCOMBINATION: System combination of three systems: EJPREORDER, a baseline Moses-Chart syntax-augmented MT with English syntax (EJSAMT), and a baseline Moses phrase-based MT (EJPBMT). (run ID: `NTITI-ej-1`)

EJPREORDER: Moses phrase-based MT with a dependency-based English pre-ordering. (run ID: `NTITI-ej-2`)

All individual systems used a word 6-gram language model trained on supplied bilingual and monolingual (JPO patent applications) corpora, by the same procedure as the English word 7-gram model in the previous section. Reordering limits for the individual systems were: 6 for EJPREORDER, unlimited (`max-chart-span=999`) for EJSAMT, and 18 for EJPBMT.

Table 6 shows evaluation results of our English-to-Japanese systems. Our primary run by EJCOMBINATION was the best at BLEU, RIBES, and Acceptability, and the individual system EJPREORDER was the best at Adequacy.

Table 2: Translation examples of JEPREORDER in Japanese-to-English subtask.

Example 2-1	
Source	ディスプレイ 用 筐 体 13 の 輪郭 は 本体 筐 体 12 の 輪郭 に 重なる 。
Pre-ordered Japanese	は(topic marker) 輪郭(outline/contour) の(of) ディスプレイ(display) 用(for) 筐体(enclosure/case) 13 重なる(overlap) に(to) 輪郭(outline/contour) の(of) 本体(body) 筐体(enclosure/case) 12 。
JEPREORDER	The contour of the display enclosure 13 overlaps the contour of the body case 12 .
JEPBMT	A main body housing 12 with the contour of the contour of the display enclosure 13 .
Reference	The outline of the display enclosure 13 is homologized with the outline of the main body enclosure 12 .
Example 2-2	
Source	高 利得 経路 における トランジスタ Q 2 , Q 3 は , カス コード 接続 さ れ て いる 。
Pre-ordered Japanese	さ れて いる カスコード 接続(are cascode-connected) は(topic marker) トランジスタ(transistor) Q 2 , Q 3 おける に(in) 高(high) 利得(gain) 経路(path) 。
JEPREORDER	Are cascode - connected transistors Q 2 and Q 3 in the high gain path .
JEPBMT	High gain path of the cascode - connected transistors Q 2 and Q 3 .
Reference	The transistors Q 2 , Q 3 along the high gain path are cascade connected .

Table 3: Translation examples of JEPOSTORDER in Japanese-to-English subtask.

Example 3-1	
Source	ここ で , 表 1 及び 図 7 に 示す 各 記号 は , 次 の もの を 表し て いる 。
Head-Final English	here , Table 1 and FIG. 7 in shown symbol va0 following element va2 represents .
JEPOSTORDER	Here , the symbols shown in Table 1 and FIG. 7 represents the following elements .
JEPBMT	Here , the following Table 1 and shown in FIG. 7 of each symbol .
Reference	Here , symbols shown in Table 1 and FIG. 7 represent the following items .
Example 3-2	
Source	また , 実質 的 に は 70 ° より 小さい 偏向 角度 と する こと が 好ましい 。
Reference	substantially , a deflection angle smaller than 70 ° is preferable .
Head-Final English	Also , substantially va0 70 ° than smaller is deflection angle va2 set to va0 preferable is .
JEPOSTORDER	Also , substantially is preferable to set the deflection angle is smaller than 70 ° .
JEPBMT	Further , it is preferable that the deflection angle is substantially smaller than 70 ° .
Reference	substantially , a deflection angle smaller than 70 ° is preferable .

Table 5: Post-evaluation results with different language models in Japanese-to-English subtask.

Language Model	BLEU	RIBES
JEPREORDER		
baseline 5-gram	0.2973	0.6955
baseline 6-gram	0.2998	0.6971
large 5-gram	0.3027	0.6927
large 6-gram	0.3062	0.6924
large 7-gram	0.3079	0.6911
JEPOSTORDER		
baseline 5-gram	0.3126	0.7126
baseline 6-gram	0.3136	0.7127
*official result (baseline 6-gram)	0.3129	0.7171
large 5-gram	0.3146	0.7055
large 6-gram	0.3153	0.7060
large 7-gram	0.3145	0.7027
JEPBMT		
baseline 5-gram	0.2921	0.6715
baseline 6-gram	0.2927	0.6717
large 5-gram	0.2977	0.6762
large 6-gram	0.2970	0.6698
large 7-gram	0.3010	0.6796

Table 6: Intrinsic evaluation results in Japansese-to-English subtask.

System	BLEU	RIBES	Ave. Adequacy
System Combination			
EJCOMBINATION	**0.4289**	**0.7984**	3.81
Individual Systems			
EJPREORDER	0.4207	0.7939	**3.84**
EJSAMT	0.3793	0.7598	n/a
EJPBMT	0.3665	0.7156	n/a
Other teams' runs			
`EIWA-ej-1`	0.3693	0.7692	3.42
`TSUKU-ej-1`	0.3141	0.7556	2.79
`JAPIO-ej-1`	0.2736	0.7281	3.53
`BASELINE1-1`	0.3298	0.7231	2.69
`RBMT6-1`	0.2461	0.7229	3.47

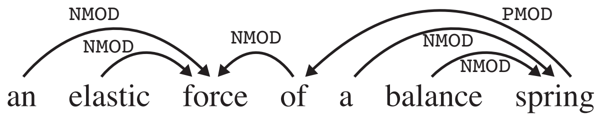

Figure 1: Example of English dependency sub-structure.

4.2 Pre-ordering

In the NTT-UT system for NTCIR-9 English-to-Japanese subtask, Head Finalization [5] worked very effectively to overcome rule-based systems in subjective evaluation. We extended it by our in-house English dependency parser trained using patent corpora.

The dependency parser was based on semi-supervised learning [15]. We developed a dependency corpus of 10,000 English parent sentences and used it with a Penn Treebank corpus to train the dependency parser. The parser was further adapted to the patent domain by semi-supervised learning using English patent sentences without dependency annotations.

Head Finalization was originally developed based on Enju[5] (English HPSG parser) and its pre-ordering reorders syntactic head nodes into right hand side in binary trees. However, when we use dependency parse results, moving heads into rightmost position in dependency structures provides very different pre-ordering results. We developed new, a bit more complex pre-ordering rules to emulate Head Finalization with dependency parse results. Basically the new rules move a head into rightmost position, but they also moves modifiers that modifies the head from left. Here, we use a exception rule to prohibit words from moving toward coordination conjunctions and punctuations as in Head Finalization [5].

For example, in a dependency sub-structure shown in Figure 1, we can get a pre-ordering result:

elastic balance string of force

by moving the head word "force" into the rightmost position and removing articles "a" and "an". Here, the noun modifier "elastic" should be adjacent to its head. So we moved such modifiers together with their head as:

balance string of elastic force

Table 7 shows translation examples. In Example 4-1, EJPREORDER successfully translated this complex structured sentence with two nested subordinate clauses. Baseline systems failed to solve this complex reordering. In Example 4-2, EJPREORDER failed to identify coordination of "the phase of the voltage" and "the phase of the current". It identified "the phase of the voltage" and "the phase of the current in the driving signal" as a coordination and wrongly reordered them. This is a common problem in a syntax-based pre-ordering; a syntax-based pre-ordering method works poor when the parsing fails.

4.3 Post-Evaluation Results

We conducted an additional experiment with different configurations after the official submission. In this post-evaluation

[5]http://www.nactem.ac.uk/tsujii/enju/

Table 7: Translation examples of JEPREORDER in Japanese-to-English subtask.

Example 4-1	
Source	this is the step in which a determination is made as to whether the count value X is less than the setting value Y / 2 .
Head-Final English	this va0 determination va1 count value X va1 setting value Y / 2 than less is whether to as made is which in step is .
EJPREORDER	この 判定 は , カウント 値 X が 設定 値 Y / 2 より 小さい か 否 か を 判定 する ステップ で ある 。
EJSAMT	これ は , ステップ で 判定 を 行う よう に する か , カウント 値 X が 小さい 場合 , 設定 値 Y / 2 。
EJPBMT	これ は , カウント 値 X 未満 で ある か 否 か を 判定 する ステップ で は , 設定 値 Y / 2 で ある 。
Reference	計数 値 X が 設定 値 Y / 2 未満 か 否 か を 判断 する ステップ で ある 。
Example 4-2	
Source	this change generates a phase difference between the phase of the voltage and the phase of the current in the driving signal .
Head-Final English	this change va0 voltage of phase and current of driving signal in phase between phase difference generates .
EJPREORDER	この 変化 は , 電圧 の 位相 と 電流 の 駆動 信号 の 位相 と の 間 に 位相 差 が 発生 する 。
EJSAMT	この 変化 は , 駆動 信号 の 電流 の 位相 と 電圧 の 位相 と の 位相 差 を 生成 する 。
EJPBMT	この 変化 は , 電流 の 位相 が 電圧 の 位相 と の 位相 差 と , 駆動 信号 で ある 。
Reference	この 変化 に 応じ て 駆動 信号 の 電圧 位相 および 電流 位相 の 位相 差 は 生み 出され る 。

Table 8: Post-evaluation results with different language models in English-to-Japanese subtask.

Language Model	BLEU	RIBES
EJPREORDER		
baseline 5-gram	0.4025	0.7877
baseline 6-gram	0.4057	0.7887
large 5-gram	0.4164	0.7927
large 6-gram	0.4207	0.7939
EJSAMT		
baseline 5-gram	0.3651	0.7534
baseline 6-gram	0.3663	0.7556
large 5-gram	0.3776	0.7586
large 6-gram	0.3793	0.7598
EJPBMT		
baseline 5-gram	0.3441	0.7056
baseline 6-gram	0.3485	0.7081
large 5-gram	0.3613	0.7110
large 6-gram	0.3665	0.7156

experiment, we compared four different language models: baseline 5- and 6-gram models trained only using supplied bilingual corpora, and large 5- and 6-gram models trained using supplied bilingual and monolingual corpora.

Results are shown in Table 8. The results clearly show the effectiveness of the large-scaled language models. These are very different from our Japanese-to-English results, in which the large-scaled language models did not work. This may be from the difference in translation accuracy ranges between E-to-J and J-to-E or smaller constraints of word order in Japanese.

We also tried to use word 7-gram language models but did not use them in the final experiments. A baseline 7-gram model was worse than the 6-gram model in our pilot test. A large 7-gram model could not be trained by our procedure (SRILM `ngram-merge` aborted due to memory shortage).

5. CONCLUSION

In this shared task, we extended our NTCIR-9 systems by improving individual systems. We achieved the best results in English-to-Japanese subtask among all runs by our pre-ordering with domain-adapted dependency parsing. Our combination system was the best among all SMT runs in Japanese-to-English subtask, but RBMT systems showed much better performance.

Acknowledgments

We would like to thank the PatentMT organizers for their great effort for the shared task.

6. REFERENCES

[1] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata. Generalized Bayes Risk System Combination for Machine Translation. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, 2011.

[2] I. Goto, K. P. Chow, B. Lu, E. Sumita, and B. K. Tsou. Overview of the Patent Machine Translation Task at the NTCIR-10 Workshop. In *Proceedings of the NTCIR-10 Workshop*, 2013.

[3] C. He, C. Wang, Y. X. Zhong, and R. F. Li. A survey on learning to rank. In *Proc. of International Conference on Machine Learning and Cybernetics*, 2008.

[4] S. Hoshino, Y. Miyao, K. Sudoh, and M. Nagata. Pre-ordering based on Predicate-Argument Structures for Japanese-to-English Statistical Machine Translation. In *Proceedings of the 19th Annual Meeting of the Association for Natural Language Processing*, 2013. (in Japanese).

[5] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh. HPSG-based Preprocessing for English-to-Japanese Translation. *ACM Transactions of Asian Language Information Processing*, 11(3), 2012.

[6] T. Joachims. Training linear SVMs in linear time. In *Proc. of KDD*, 2006.

[7] J. Katz-Brown and M. Collins. Syntactic Reordering in Preprocessing for Japanese-English Translation: MIT System Description for NTCIR-7 Patent Translation Task. In *Proceedings of the NTCIR-7 Workshop*, 2008.

[8] M. Komachi, Y. Matsumoto, and M. Nagata. Phrase Reordering for Statistical Machine Translation based on Predicate-Argument Structure. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2006)*, pages 77–82, 2006.

[9] S. Kondo, M. Komachi, Y. Matsumoto, K. Sudoh, K. Duh, and H. Tsukada. Learning of Linear Ordering Problems and its Application to J-E Patent Translation in NTCIR-9 PatentMT. In *Proceedings of the NTCIR-9 Workshop*, 2011.

[10] F. J. Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[11] H. Sajjad, A. Fraser, and H. Schmid. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–477, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[12] K. Sudoh, K. Duh, H. Tsukada, M. Nagata, X. Wu, T. Matsuzaki, and J. Tsujii. NTT-UT Statistical Machine Translation in NTCIR-9 PatentMT. In *Proceedings of the NTCIR-9 Workshop*, 2011.

[13] K. Sudoh, X. Wu, K. Duh, H. Tsukada, and M. Nagata. Post-ordering in Statistical Machine Translation. In *Proceedings of the 13th MT Summit*, 2011.

[14] K. Sudoh, X. Wu, K. Duh, H. Tsukada, and M. Nagata. Syntax-based Post-ordering for Efficient Japanese-to-English Translation. *ACM Transactions on Asian Language Information Processing*, accepted for publication.

[15] J. Suzuki, H. Isozaki, X. Carreras, and M. Collins. An empirical study of semi-supervised structured conditional models for dependency parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 551–560, Singapore, August 2009. Association for Computational Linguistics.