Journal of Information Processing Vol.20 No.4 871–882 (Oct. 2012)

[DOI: 10.2197/ipsjjip.20.871]

Regular Paper

An Empirical Comparison of Parsers in Constraining Reordering for E-J Patent Machine Translation

Isao Goto[1,a)] Masao Utiyama[1] Takashi Onishi[1] Eiichiro Sumita[1]

Received: November 14, 2011, Accepted: May 12, 2012

Abstract: Machine translation of patent documents is very important from a practical point of view. One of the key technologies for improving machine translation quality is the utilization of syntax. It is difficult to select the appropriate parser for English to Japanese patent machine translation because the effects of each parser on patent translation are not clear. This paper provides an empirical comparative evaluation of several state-of-the-art parsers for English, focusing on the effects on patent machine translation from English to Japanese. We add syntax to a method that constrains the reordering of noun phrases for phrase-based statistical machine translation. There are two methods for obtaining the noun phrases from input sentences: 1) an input sentence is directly parsed by a parser and 2) noun phrases from an input sentence are determined by a method using the parsing results of the context document that contains the input sentence. We measured how much each parser contributed to improving the translation quality for each of the two methods and how much a combination of parsers contributed to improving the translation quality for the second method. We conducted experiments using the NTCIR-8 patent translation task dataset. Most of the parsers improved translation quality. Combinations of parsers using the method based on context documents achieved the best translation quality.

Keywords: patent translation, parser, comparison, reordering constraint, English to Japanese

1. Introduction

In recent years, demands for patent machine translation have increased. With globalization comes an increase in the need for the international circulation of patent documents. It is, therefore, important to improve the quality of machine translation of patent sentences. Word ordering is the main issue in statistical machine translation of long patent sentences between language pairs with widely different word orders, such as English-Japanese. One of the key technologies for improving translation quality is the utilization of syntax to determine proper word order. The syntax of an input sentence is considered useful in determining the word order of a translated sentence. It is difficult to select the appropriate parser for patent translation. There are mainly two reasons:

- Parsing is a difficult task, and several methods have been proposed in recent years. There are probabilistic CFG-based parsers [3], [5], [13], [29], dependency parsers [20], [25], and an HPSG-based parser [22].
- The effects of each parser on patent translation are not clear in the commonly used evaluations of parsers. Most state-of-the-art parsers for English were trained with the Wall Street Journal (WSJ) from the Penn Treebank corpus. Such parsers were evaluated by measuring bracketing precision and recall of the output using the WSJ from the Penn Treebank corpus. From the evaluation, it is not clear how well these models work in other domains such as the patent domain.

To the best of our knowledge, no prior study has compared the effects of parsers on patent machine translation. One study examined the relation between parse accuracy and translation quality [31]. This showed the relationship between a parser's training data size and the translation quality. They did not compare parsers, nor did they use a patent corpus. Studies have also been done on the relationship between four parsers and translation quality [37], and on the use of the four parsers in combination and translation quality [34] for string-to-tree based statistical machine translation. These studies did not use a patent corpus, and only evaluated probabilistic CFG-based parsers. They used target side syntax and did not use source side syntax. There is a study that empirically compared parsers [21] based on a task-oriented evaluation. This study compared parsers based on the accuracy of identifying protein-protein interaction by using parser output as features for machine learning models. It did not evaluate parsers for patent machine translation.

In this paper, we compare the effects of several state-of-the-art parsers on patent machine translation. This research reveals how effective each parser is in patent machine translation.

There are statistical machine translation methods that use input sentence syntax: reordering constraint methods [4], [19], [27], [35], [36], tree-to-string methods [12], [16], and tree-to-tree methods [6], [17], [38]. In this study, we use a reordering constraint method which directly controls word order using the syntax of an input sentence for phrase-based statistical machine translation, one of the widely used statistical translation methods. The syntax structure is obtained using each parser to be com-

[1] National Institute of Information and Communications Technology, Keihanna Science City, Kyoto 619–0289, Japan

[a)] igoto@nict.go.jp

pared. We evaluate the effects of each parser on patent machine translation by evaluating patent machine translation quality.

Moreover, we apply a method that used parsed context documents containing the input sentence to determine the noun phrases in the input sentence [27]. Our results show how effective their method was with each parser and combination of parsers.

The main contributions of this paper are:

Novelty This paper is novel in that it gives data that provides new findings. Until now, there had been no patent machine translation research substantiating and comparing the effectiveness of parsers on patent machine translation. There had also not been research investigating the accuracy of parsers on patent sentences. As a result, up to now, the effectiveness of each of the well-known publicly available parsers on patent machine translation was unknown. Our research results are on this point of uncertainty.

Effectiveness Because our substantiative research used well-known, publicly available parsers and a machine translation system, people who need patent machine translation can easily utilize the findings from our results. As there is a significant practical need in industry for patent machine translation, knowledge contributing on patent machine translation can have a large impact on industry.

The rest of this paper is organized as follows: In Section 2, we show the six parsers that we compared. In Section 3, we explain the method of comparison. In Section 4, we discuss the experiment results from the NTCIR-8 patent translation task data. In Section 5, we conclude this paper.

2. Parsers

We focused on seven well-known publicly available parsers. The parsers are categorized by method into three groups: probabilistic CFG parser, dependency parser, and HPSG parser.

2.1 Probabilistic CFG Parser

Owing to Penn Treebank [18], there has been a lot of research into parsers based on probabilistic CFG that output phrase structures. **Figure 1** shows an example of a phrase structure. The ways to parameterize the probabilistic models vary. In this research, we used the following four parsers:

COLLINS Collins' parser [5]. The parser uses a lexicalized probabilistic CFG model. The tool includes three models: model 1, 2, and 3. Model 1 is the base model, model 2 adds a complement/adjunct distinction, and model 3 adds a wh-movement model on top of that. We used model 3. Since the tool did not include a POS tagger function, we used Tsuruoka's English POS tagger [32] to obtain part-of-speech.

Collins' parser outputs noun phrase (NP) structures that includes periods and commas following noun phrases. We tested not only the original parsing results but also the modified parsing results that periods and commas at the end of NP structures were excluded from the NP structures. We call the modified case **COLLINS (modify)**.

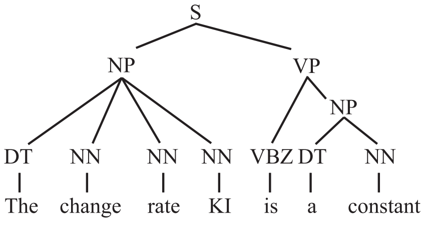

Fig. 1 Penn Treebank-style phrase structure.

CHARNIAK Charniak's parser [3]. The parser uses a lexicalized probabilistic CFG model. The model is based on the principle of maximum entropy.

STANFORD Stanford's parser [13]. The parser uses an unlexicalized probabilistic CFG model. We used version 1.6.5.

BERKELEY Berkeley's parser [29]. The parser uses an unlexicalized probabilistic CFG model using latent variables that refine each non-terminal node. We used release 1.1.

2.2 Dependency Parser

Owing to the CoNLL shared tasks [1], [26], research into dependency parsing have been active. Dependency structure is a tree structure in which a node is a word and an edge is the relation between a parent node and a child node. A child node modifies its parent node. **Figure 2** shows an example of a dependency tree structure. In this research, we used the following two parsers:

MST MacDonald and Pereira's parser [20]. Projective dependency parsing is based on Eisner's algorithm [9]. We used version 0.4.3b. The tool did not contain a model. We built a model using WSJ section 2 to 21 from Penn Treebank. We used the default settings (the first order model) for the parser. Since the tool did not include a POS tagger function, we used Tsuruoka's English POS tagger [32] to obtain part-of-speech.

MALT MaltParser [25]. This parser is an implementation of inductive dependency parsing with deterministic parsing algorithms. We used version 1.6.1. Since the tool did not include a POS tagger function, we used Tsuruoka's English POS tagger [32] to obtain part-of-speech. We call the case using a model published on the MaltParser Web page [*1] **MALT**. We used the model of SVMs with a polynomial kernel for the classification for MALT.

In addition, to enable a fair comparison of the algorithms between MaltParser and MST, we built a model using WSJ section 2 to 21 from Penn Treebank, which is the same data we used to build the MST model. We used the default settings for this train: namely Nivre's algorithm [24] and the SVM model was used for the parser. We call this case using

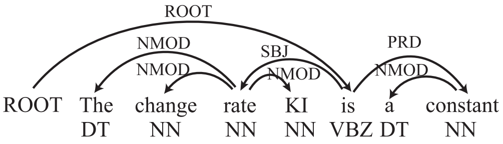

Fig. 2 Dependency tree structure.

[*1] http://maltparser.org/

the newly trained model **MALT (train)**.

2.3 HPSG Parser

There is a parser based on the HPSG [30] theory. HPSG-based parsers analyze not only phrase structure but also deeper structures, such as the arguments of a predicate, simultaneously. We used only the phrase structures of the parsing results. In this research, we used the following parser:

ENJU An HPSG parser [22]. It consists of an HPSG grammar extracted from the Penn Treebank, and a maximum entropy model trained with an HPSG Treebank derived from the Penn Treebank. We used version 2.3.1.

3. Comparison Methodology

We compared parsers based on the translation quality of patent sentences translated by a phrase-based statistical machine translation with reordering constraints using syntax of input sentences. We translated from English to Japanese, whose word orders are widely different. In translation between languages with widely different word orders, it is difficult to assign the proper word order, especially with long input sentences. Input sentence syntax is useful in deciding a word order for the translated sentence. We parsed the input sentence and constrained the word order using these parsed results. The translation quality was measured using the 4-gram BLEU [28] scores, the NIST [7] scores, and the WER [23] values. For BLEU and NIST, a larger value is better. For WER, a smaller value is better. There is a method that determines the noun phrases (NPs) in an input sentence by using the parsing results of the context document that contains the input sentence. We applied this method and compared parsers based on the translation quality. We also examined the effects of this method on each parser and combinations of parsers.

First, we show the issue of patent translation. Next, we explain the methods that deal with the issue by constraining reordering using syntax of input sentences. Finally, we explain the method that estimates noun phrases using context documents.

3.1 Patent Translation

In this research, we focused on the translation of patent sentences. Patent sentence translation is difficult and the main reason for this is that patent sentences are long. As shown in **Table 1**, patent sentences are longer than those in other domains. In general, longer sentences cause an explosion of reordering combinations and degrade translation quality.

When we translate between languages with similar word orders, we can prevent the loss of translation quality by using distortion limits that constrain word reordering in phrase-based statistical machine translation. However, in translation between language pairs with widely different word orders, such as English-Japanese, long-distance word reordering is required when an input sentence is long. Therefore, when an input sentence for translation is long, the word order possibilities are large. This leads to difficulty in determining the proper word reordering.

Table 1 Average sentence length in three domains. Sentence length is the number of words per English sentence. We used the IWSLT corpus [8] in the travel domain, the WMT08 News Commentary corpus [2] in the news domain, and the NTCIR-8 Patent machine translation corpus [11] in the patent domain.

Domain	Sentence length
Travel	7.7
News	21.0
Patent	30.3

Below is an example of translation by our baseline system without using syntax. This example shows how a failure of word order affects the overall translation quality. **Table 2** gives the meanings of the expressions in the Baseline Output.

Input sentence

> **a rotational position-detecting device 3 that is constructed of a resolver or a rotary encoder** is mounted on a shaft of a rotor , not shown , of the electric motor 1 .

Baseline Output

> 電動 機 1の **回転 位置 検出 装置 3** 、 図示 し ない ロータ の 軸 に 装着 さ れ た **ロータリー エンコーダ や レゾルバ の で 構成 さ れる** 。

The bolded section of the input sentence was translated into two separated parts, in Gothic, in the baseline output. The bolded section of the input sentence refers to a single apparatus. Thus, if the expression is translated as two separate expressions, the original meaning cannot be understood and is lost.

3.2 Reordering Constraint for Phrase-based Statistical Machine Translation

To address this word reordering issue, it is important to utilize syntax to improve word order quality. The syntax of an input sentence is considered useful to determine the word order of a translated sentence. There are methods that use input sentence syntax: reordering constraint methods [4], [19], [27], [35], [36], tree-to-string methods [12], [16], and tree-to-tree methods [6], [17], [38]. In this research, we focused on a reordering constraint method. Reordering constraint methods directly control word order using the syntax of an input sentence. We investigated the effects of parsers on a phrase-based statistical machine translation with reordering constraints. While it would have been interesting to compare the effects of parsers for other methods, we decided it was something to be considered for future work.

Using the aforementioned example, a reordering constraint that translates the bold section of the input sentence into one block reduces incorrect word ordering and improves translation quality.

For this research, we used parsers to obtain the syntax structures in input sentences, and constrained reordering to translate a

Table 2 Meanings of expressions in the Baseline Output in order of the output.

Expressions in the output	Meanings in English
電動 機 1 の	of the electric motor 1
回転 位置 検出 装置 3	**a rotational position-detecting device 3**
図示 し ない	not shown
ロータ の 軸 に 装着 さ れ た	mounted on a shaft of a rotor
ロータリー エンコーダ や レゾルバ の で 構成 さ れる	**is constructed of a resolver or a rotary encoder**

noun phrase distinguished by parsing as one block.

The Moses phrase-based decoder has a function that constrains reordering using zone tags [15]. Moses restricts reordering that violates zones specified by zone tags, and translates one zone to one block. We used this function of the Moses decoder to translate a noun phrase of parsed results into one block. We add zone tags that cover noun phrases to an input sentence. Zone tags can be nested if the new tag does not conflict with other existing tags. Below are examples of an input sentence with zone tags covering noun phrases obtained by the Berkeley parser and its translation output:

Input sentence with zone tags

⟨zone⟩ ⟨zone⟩ **a rotational position-detecting device 3** ⟨/zone⟩ **that is constructed of** ⟨zone⟩ ⟨zone⟩ **a resolver** ⟨/zone⟩ **or** ⟨zone⟩ **a rotary encoder** ⟨/zone⟩ ⟨/zone⟩ ⟨/zone⟩ is mounted on ⟨zone⟩ ⟨zone⟩ a shaft ⟨/zone⟩ of ⟨zone⟩ a rotor ⟨/zone⟩ ⟨/zone⟩ , not shown , of ⟨zone⟩ the electric motor 1 ⟨/zone⟩ .

Output of the input sentence with zone tags

ロータリー エンコーダ や レゾルバ の 回転 位置 検出 装置 3 は 、 図示 し ない 電動 モータ 1 の ロータ の 軸 に 装着 さ れ て いる 。

The bolded section of the input sentence refers to a single apparatus. In the previous output example without zone tags, the section was translated into two separate parts. In contrast, in the output example with zone tags, the section was translated into one part in Gothic. By translating one noun phrase into one part, the translation of the bolded section of the input sentence was able to express a single apparatus.

The mechanism of the zone constraint

Here, we explain the mechanism of the zone constraint in detail. Phrase-based SMT generates a translation sentence by sequencing phrases from beginning of a sentence to the end. Once a word in a zone has been translated, any translation that comes next is restricted: that is, none of the words outside the zone can be translated until all the words inside the zone are translated. There is an exception: when a phrase includes words both inside and outside the zone, and all the words inside the zone will be translated by translating the phrase, the phrase can be used to translate (words inside and outside the zone are simultaneously translated). This restriction makes the translation of an expression covered by a zone into a contiguous sequence of translated phrases in the target language. An example is shown in **Fig. 3**, in which e_i is an English word, s_i is an English phrase, j_i is a Japanese word, and t_i is a Japanese phrase. Figure 3 represents a situation in which t_1 and t_2 have been decided and t_3 is being selected. At this time, since e_3 (inside a zone) has been translated, e_6 and e_7 (outside the zone) are not translated until the remaining e_4 and e_5 in the zone are translated. Therefore, in Fig. 3, English phrase candidates corresponding to t_3 are the phrases with ○, not the phrases with ×. Since phrases crossing zone spans are used for translation (e.g., phrase s_9 in Fig. 3), phrases used to translate an input sentence with zone tags are the same as a case without zone tags. The difference between with and without zone tags is the presence or absence of phrase order restriction.

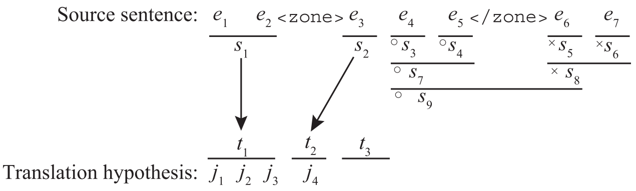
Source sentence: e_1 e_2 <zone> e_3 e_4 e_5 </zone> e_6 e_7
s_1
s_2
○s_3
○s_4
×s_5
×s_6
○ s_7
× s_8
○ s_9
t_1
t_2
t_3
Translation hypothesis: j_1 j_2 j_3 j_4

Fig. 3 Example of zone constraint.

Zone for dependency structure

Dependency structures do not explicitly express noun phrases. We regarded a subtree whose root node is a noun as a noun phrase. A "subtree" consists of a node and all of its descendent nodes. **Figure 4** shows an example of noun phrases extracted from a dependency structure.

3.3 Using Context Documents

Onishi et al. [27] proposed a method that did not use the noun phrases obtained by parsing an input sentence directly, but instead used the noun phrases determined by using the parsing results of a context document, a document that contains an input sentence. This method can determine noun phrases by considering document-level consistency. We explain this method using **Fig. 5**. The numbers in parentheses in Fig. 5 correspond to the following numbers. The method is as follows:

(1) The method parses a context document containing an input sentence.
(2) The method extracts all noun phrases from the parsing results.
(3) The method ranks the noun phrases based on a C-value [10] that gives high rank to phrases with high termhood from nested candidates.
(4) The method searches the list of noun phrases (in order of rank) for expressions that appear in the input sentence and determines the searched expression to be a noun phrase if the expression does not conflict with existing noun phrases.
(5) The method adds zone tags which cover the noun phrases.

The C-value of a phrase p is expressed in the following equation:

$$C\text{-}value(p)=\begin{cases}(l(p)-1)\,n(p) & (|Q|=0)\\ (l(p)-1)\left(n(p)-\frac{t(Q)}{|Q|}\right) & (|Q|>0)\end{cases}$$

where

$l(p)$ is the length of a phrase p,

$n(p)$ is the frequency of p in a document,

Q is the set of phrases that contain p as a subphrase and appear in the document,

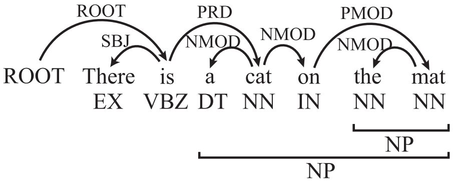

Fig. 4 Example of noun phrases extracted from a dependency tree structure.

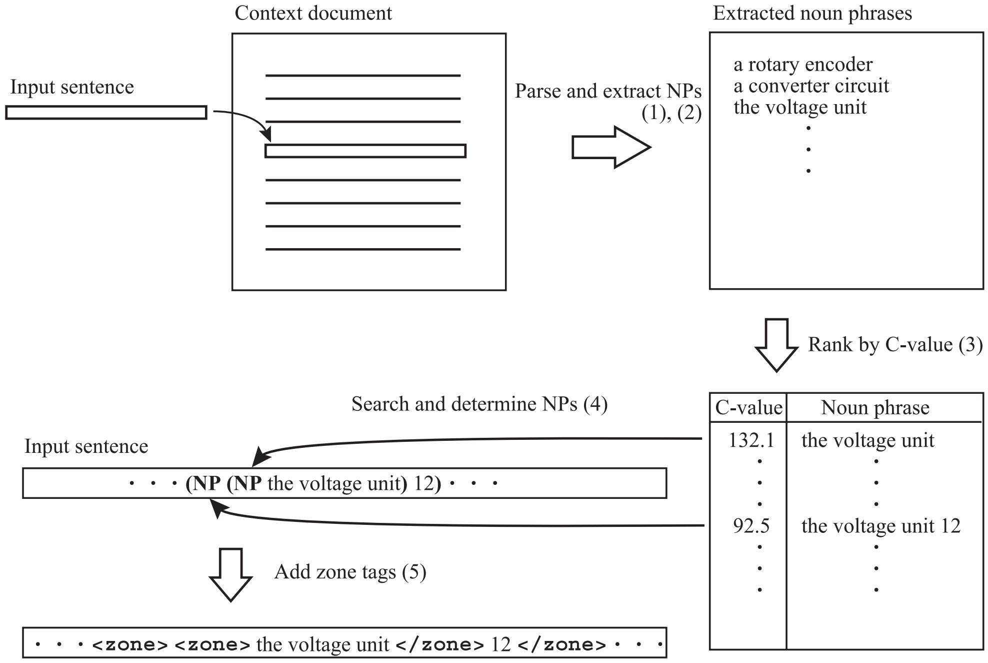

Fig. 5 Method using context document.

$t(Q)$ is the total frequency of phrases in Q, i.e., $t(Q) = \sum_{q \in Q} n(q)$ where q is a phrase, and
$|Q|$ is the number of phrases in Q.

Onishi et al. [27] pointed out that since phrases with large C-values frequently occur in a context document, these phrases are considered a significant unit, i.e., a part of the invention, and are assumed to be translated as single blocks.

3.4 Parser Combination Using Context Documents

We used a combination of parsers in which one parser parsed a context document while another parser parsed the same context document. We used the two documents that had been parsed as parsed context documents and extracted noun phrases from them. The subsequent processes are the same as the processes (3) to (5) described in Section 3.3.

4. Experiment

We conducted English to Japanese patent translation experiments using the NTCIR-8 patent translation task data [11]. This data set consists of approximately 3.2 million English-Japanese sentence pairs, development data of 2,000 sentence pairs, and test data of 1,119 sentences and their single reference data, as shown in **Table 3**. Furthermore, this dataset contains the patent specifications from which the test sentences were extracted. We used these patent specifications as context documents.

We used the same SMT model (i.e., the same phrase table, the same reordering model, the same language model, and the same feature weight) for all the experiments. No zone tags were used in the training. The only difference is the presence or absence of zone tags or the different zone tags in the input sentences. The SMT model is shown as baseline system in the next subsection.

Table 3 Statistics for the NTCIR-8 English to Japanese patent translation task dataset.

Set	Number of sentences
Training	Approximately 3.2 million
Development	2,000
Test	1,119

4.1 Baseline

We used Moses for the machine translation system. The following settings were used:

- GIZA++ and grow-diag-final-and heuristics,
- 5-gram language model with interpolated modified Kneser-Ney discounting,
- msd-bidirectional-fe lexicalized reordering,
- distortion-limit = −1 (unlimited).

The feature weights were tuned by minimum error rate training using the development data.

4.2 Experiment 1

We evaluated parsers based on the effects of the reordering constraints where the parsing results of the test sentences were directly used. We parsed the test sentences using each parser and annotated the zone tags that cover noun phrases as described at Section 3.2.

Results and discussion

Table 4 gives the results of the translations using the reordering constraint of zone tags covering the noun phrases obtained directly by the parsers. "Baseline" indicates the result that did not use a parser and zone tags.

All of the parsers except for COLLINS without modification of output had improved the three automatic values over the baseline automatic values. From these results, it can be seen that all of the seven parsers COLLINS (modify), CHARNIAK, STANFORD, BERKELEY, MST, MALT (train), and ENJU were effective for

Table 4 Comparison between parsers based on the effects of reordering constraints using the parsing results of test data. "Improvements" shows the improvements from Baseline.

	BLEU		NIST		WER	
	Score	Improvements	Score	Improvements	Rate	Improvements
Baseline	38.42	0	8.117	0	0.7105	0
COLLINS	36.97	−1.45**	8.021	−0.097	0.7159	−0.0054
COLLINS (modify)	39.19	0.77**	8.180	0.063	0.6824	0.0281
CHARNIAK	39.24	0.82**	8.185	0.068	0.6727	0.0378
STANFORD	39.56	1.14**	8.208	0.091	0.6703	0.0402
BERKELEY	**39.60**	1.18**	**8.224**	0.107	**0.6684**	0.0421
MST	39.45	1.03**	8.201	0.084	0.6763	0.0342
MALT	38.60	0.18	8.150	0.033	0.6832	0.0273
MALT (train)	39.27	0.85**	8.204	0.087	0.6828	0.0277
ENJU	39.37	0.95**	8.192	0.075	0.6675	0.043

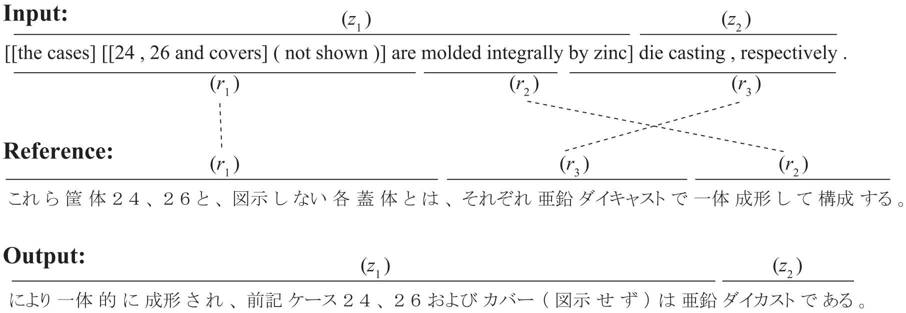
Input:
(z1)
(z2)
[[the cases] [[24 , 26 and covers] (not shown)] are molded integrally by zinc] die casting , respectively .
(r1)
(r2)
(r3)
Reference:
(r1)
(r3)
(r2)
これ ら 筐 体 ２ ４ 、２ ６ と 、図示 し ない 各 蓋 体 と は 、それぞれ 亜鉛 ダイキャスト で 一体 成形 し て 構成 する 。
Output:
(z1)
(z2)
により 一体 的 に 成形 さ れ 、前記 ケース ２ ４ 、２ ６ および カバー （ 図示 せ ず ）は 亜鉛 ダイカスト で ある 。

Fig. 6 Example of a parsing error that caused a translation error. Zone tags are represented by "[" and "]".

patent machine translation.

We checked for what kind of parsing errors caused the translation errors. When a sequential expression in the source language has to be separated into two parts in the target language, if the sequential expression is analyzed as one zone due to parsing errors, words outside the zone cannot be inserted into the translation of the sequential expression. In other words, a correct word order cannot be selected if such parsing errors occur. An example of such parsing error is shown in **Fig. 6**. Each part of the input and the reference of r_1, r_2, and r_3 corresponds. Each part of the input and the output of z_1 and z_2 corresponds. Because z_1 is a zone, the translation of z_2 could not be inserted into the translation of z_1. This caused a word order error.

We investigated what caused the degradation of the COLLINS results. The Collins' parser output NP structures that included periods and commas following noun phrases. For example, "There is a cat ." was analyzed as "There is (NP a cat .)." This was the reason for the degradation. English word order is Subject-Verb-Object (SVO), so sometimes there is an NP at the end of a sentence in English. In contrast, Japanese word order is Subject-Object-Verb (SOV), so an NP is usually not placed at the end of a sentence in Japanese. If a zone includes an NP in English and a period following the NP, a Japanese verb cannot be inserted between the translation of the NP and the translation of the period because the two translations must be contiguous by the zone constraint. What this means is that correct word order cannot be selected due to the zone constraint. We therefore conducted an experiment using the modified NP structures that periods and commas at the end of NP structures were excluded for the Collins' parser. COLLINS (modify) is the modified results of the Collins' parser.

The comparison results of the parsers based on the values of the three automatic measures (BLEU, NIST, and WER) were roughly the same. BERKELEY was the best for all automatic measures.

We calculated a statistical significance test for the differences of the BLEU scores. The "**" mark of the BLEU score differences in Table 4 and Table 5 denotes a statistical significant difference at the significance level of $\alpha = 0.01$ and the "*" mark of the BLEU score differences denotes a statistical significant difference at the significance level of $\alpha = 0.05$ according to the bootstrap resampling test [14].

The difference of BLEU scores between BERKELY and the top parsers STANFORD, MST, MALT (train), and ENJU was not significant at $\alpha = 0.05$ and the difference between BERKELEY and CHARNIAK was significant at $\alpha = 0.05$. From these results, it can be seen that **BERKELEY**, **STANFORD**, **MST**, **MALT (train)**, and **ENJU** were especially effective for patent machine translation among the seven parsers when the noun phrases in an input sentence were obtained by parsing the input sentence directly.

4.3 Experiment 2

We evaluated parsers based on the effects of the reordering constraints where noun phrases were determined using the parsed context documents as described in Section 3.3. We call the method using the parsed context documents "the with-context method," and we call the method using parsing results of the input sentences directly "the without-context method." We also evaluated the effect of the with-context method for each parser.

We also evaluated parser combinations based on the effects of the reordering constraints where noun phrases were determined using the parsed context documents that were parsed by two parsers as described in Section 3.4.

For this experiment, we used the noun phrases from a context

Table 5 Comparison between parsers based on the effects of reordering constraints using the parsing results of context documents. "Improvements" shows the improvements from the method using the parsing results of the test sentences directly.

	BLEU		NIST		WER	
	Score	Inprovements	Score	Improvements	Rate	Inprovements
COLLINS	37.82	0.85**	8.068	0.047	0.7080	0.0079
COLLINS (modify)	39.39	0.20	8.200	0.020	0.6798	0.0026
CHARNIAK	39.60	0.36*	**8.224**	0.039	0.6744	−0.0017
STANFORD	39.66	0.10	8.215	0.007	0.6735	−0.0032
BERKELEY	**39.73**	0.13	8.223	−0.001	**0.6728**	−0.0044
MST	39.54	0.09	8.209	0.008	0.6821	−0.0058
MALT	38.92	0.32	8.153	0.003	0.6887	−0.0055
MALT (train)	39.27	0.00	8.193	−0.011	0.6841	−0.0013
ENJU	39.59	0.22	8.212	0.020	0.6746	−0.0071

Table 6 Comparison between combinations of parsers based on the effects of reordering constraints using the parsing results of context documents. The values given are BLEU/NIST/WER.

	STANFORD	MST	ENJU
BERKELEY	**39.92**/8.232/0.6662	39.67/8.223/0.6736	39.85/**8.239**/0.6658
STANFORD		39.74/8.231/0.6696	39.83/8.234/**0.6650**
MST			39.48/8.207/0.6740

document that had C-values greater than or equal to 1.0. Most of noun phrases had C-values greater than 1.0.

Results and discussion

Table 5 gives the results of the translation using the reordering constraint of zone tags covering the noun phrases in the test data determined using parsed context documents and one parser.

All of the parsers except for COLLINS without modification of output had improved BLEU, NIST, and WER values over the Baseline in Table 4.

We also examined the effects of the with-context method. "Improvements" in Table 5 shows the improvements from the results in Table 4 of the without-context method. The BLEU scores of the with-context method were slightly higher than those of the without-context method for all of the parsers except for MALT (train). The NIST scores of the with-context method were slightly higher than those of the without-context method for all of the parsers except for MALT (train) and BERKELEY. However, WER values of the with-context method were slightly worse than those of the without-context method for all of the parsers except for COLLINS and COLLINS (modify). From these results, we can see that the with-context method slightly improved the BLEU scores but slightly degraded the WER values, and the difference of automatic scores between the results with context and the results without context is small. Therefore, the effect of the with-context method with one parser compared with the without-context method for translation quality is inconclusive.

Table 6 shows the translation results with reordering constraints using context documents parsed by two parsers. We used the top four parsers of the BLEU scores in Table 4, BERKELEY, STANFORD, MST, and ENJU for the parser combinations.

BLEU, NIST, and WER values in Table 6, except for the parser combinations including MST, are better than the single best values of the two parsers in Table 4 and Table 5.

The difference of the BLEU scores between the combination of BERKELEY and STANFORD with context, which achieved the best BLEU score and BERKELEY without context was significant at $\alpha = 0.05$. When comparing results, the difference between BERKELEY with context and BERKELEY without was not significant, whereas there was a significant difference between the combination of parsers with context and BERKELEY without context. The improvements of the parser combinations are not only the BLEU scores but also the NIST scores and the WER values. From these results, it can be seen that **using context documents with a combination of parsers is effective** for patent translation.

There are some disadvantages to the with-context method compared with the without-context method: 1) the range of application is smaller because the method needs context documents and 2) computational costs increase.

We manually compared whether the translation results of the BERKELEY and STANFORD combination, which got the best BLEU score, were better than the Baseline results. We conducted a human evaluation based on the pairwise comparison of sentence pairs for the translations of the 500 input sentences randomly selected in Section 4.4. BERKELEY and STANFORD was better for 211 sentences, Baseline was better for 89 sentences, and the remaining 200 sentences were the same or nearly equal quality. This shows that translation quality of BERKELEY and STANFORD is better than Baseline not only when gauged by automatic evaluation measures but also by human evaluation.

We investigated the causes of the degradation of the BERKELEY and STANFORD results compared with the Baseline results. Part of it was the parsing errors that caused translation errors, which are explained in Section 4.2, but there was more: due to the zone constraint, even if the parsing result was correct, the translation hypotheses that can be produced from combinations of phrases are different from those without zone constraints. Phrase-based SMT cannot search for the global optimal translation and so searches for a suboptimal translation using a beam search technique. If the producible translation hypotheses are changed, then the pruned translation hypotheses are changed during the translation process. As a result, the translation outputs from the translation hypotheses were sometimes incidentally worse than the

Table 7 Parse accuracy of noun phrases based on brackets in patent sentences.

		All NPs		Selected NPs		Bracket recall	Cross brackets
		F-measure	Bracket precision	F-measure	Bracket precision		
without context	COLLINS	41.06	42.81 (1536/3588)	43.63	48.79 (1536/3148)	39.46 (1536/3893)	1.554
	COLLINS (modify)	63.69	67.96 (2333/3433)	67.95	78.45 (2333/2974)	59.93 (2333/3893)	0.698
	CHARNIAK	64.73	59.35 (2771/4669)	72.87	74.65 (2771/3712)	71.18 (2771/3893)	1.078
	STANFORD	67.26	61.05 (2915/4775)	74.79	74.71 (2915/3902)	74.88 (2915/3893)	1.136
	BERKELEY	**69.96**	65.05 (2946/4529)	76.86	78.08 (2946/3773)	75.67 (2946/3893)	0.846
	MST	51.09	56.61 (1812/3201)	55.00	67.21 (1812/2696)	46.55 (1812/3893)	0.948
	MALT	49.67	55.22 (1757/3182)	52.38	62.39 (1757/2816)	45.13 (1757/3893)	1.258
	MALT (train)	50.19	57.09 (1743/3053)	53.20	65.53 (1743/2660)	44.77 (1743/3893)	0.89
	ENJU	62.21	69.03 (2204/3193)	64.31	74.43 (2204/2961)	56.61 (2204/3893)	0.756
with context	COLLINS	42.17	42.70 (1622/3799)	49.33	60.45 (1622/2683)	41.66 (1622/3893)	0.666
	COLLINS (modify)	53.44	54.97 (2024/3682)	62.74	79.09 (2024/2559)	51.99 (2024/3893)	**0.268**
	CHARNIAK	55.08	50.81 (2341/4607)	69.86	83.34 (2341/2809)	60.13 (2341/3893)	0.314
	STANFORD	57.07	52.48 (2435/4640)	71.13	82.43 (2435/2954)	62.55 (2435/3893)	0.362
	BERKELEY	56.99	53.06 (2396/4516)	70.96	83.78 (2396/2860)	61.55 (2396/3893)	0.334
	MST	48.47	49.90 (1834/3675)	58.41	76.83 (1834/2387)	47.11 (1834/3893)	0.41
	MALT	45.30	50.78 (1592/3135)	53.67	78.08 (1592/2039)	40.89 (1592/3893)	0.45
	MALT (train)	47.66	50.92 (1744/3425)	55.97	74.56 (1744/2339)	44.80 (1744/3893)	0.468
	ENJU	56.14	60.18 (2048/3403)	64.07	81.92 (2048/2500)	52.61 (2048/3893)	0.326
	BERKELEY & STANFORD	60.23	51.68 (2809/5435)	**76.94**	82.40 (2809/3409)	72.16 (2809/3893)	0.348
	BERKELEY & MST	56.40	50.05 (2515/5025)	72.13	81.63 (2515/3081)	64.60 (2515/3893)	0.336
	BERKELEY & ENJU	59.32	52.67 (2643/5018)	74.43	82.36 (2643/3209)	67.89 (2643/3893)	0.308
	STANFORD & MST	56.44	49.49 (2556/5165)	72.42	80.73 (2556/3166)	65.66 (2556/3893)	0.366
	STANFORD & ENJU	59.17	51.98 (2673/5142)	74.16	80.61 (2673/3316)	68.66 (2673/3893)	0.382
	MST & ENJU	54.61	51.97 (2240/4310)	66.76	79.49 (2240/2818)	57.54 (2240/3893)	0.346

Baseline translation. In the 89 degraded sentences, 47 sentences included the parsing errors that caused translation errors. The cause of the degradation of the rest 42 sentences is thought to be incidental change.

4.4 Experiment 3

We investigated the relationship between parse accuracy and translation quality. The parse accuracy for patent sentences was unclear, so we did the following: 1) built an annotated corpus, 2) clarified the parse accuracy of the noun phrases for the patent corpus, and 3) checked the relationship between the parse accuracy and translation quality. We randomly selected 500 sentences from the test sentences and manually annotated them with noun phrase tags. We calculated the parse accuracy for noun phrases using the annotated corpus. Bracket accuracy was calculated using a bracket-scoring program named *evalb* [*2].

Results and discussion

Table 7 shows the parse accuracy of noun phrases. In Table 7, the vertical category "without context" means without using context and "with context" means using context to determine noun phrases. "Cross brackets" shows the average number of cross brackets per sentence. "F-measure" shows the harmonic mean of precision and recall. The numbers in parentheses for precision and recall represent the concrete numbers of bracket pairs. The "Bracket precision" of "All NPs" shows the results for all NP structures. To calculate the Bracket precision of "Selected NPs," we calculated the precision for all NP brackets except for those meeting the following two conditions: 1) they do not match any gold bracket pairs, and 2) are included in gold brackets with no embedded gold brackets. The with-context method often adds NP brackets in an NP structure. It is thought that NP brackets in gold NP brackets with no embedded gold brackets will not have any negative effect on the translation in most cases. We feel that precision calculated on the basis of Selected NPs is more meaningful in terms of translation quality than precision calculated on the basis of All NPs. We therefore calculated precision for not only All NPs but also Selected NPs.

First, we will focus on the Selected NPs Bracket F-measure of the "without context" category. In this category, a high F-measure of parse accuracy almost produces a high quality translation. BERKELEY's F-measure was the highest and it also had the best BLEU score, NIST score, and WER value in Table 4.

Next, we focus on the difference between "without context" and "with context." As shown in the results in Table 6, the with-context method using a parser combination improved translation quality. However, as shown in Table 7, the with-context method using a parser combination did not improve the Selected NPs Bracket F-measure compared to the without-context method, which indicates that there must be an important factor other than the F-measure in translation quality.

We then examine the differences in Cross brackets. Shown in Table 7, the with-context method reduced the average number of Cross brackets compared to the number from the without-context method. In general, as the number of brackets reduces, the number of cross brackets also reduces. The Bracket recall of the with-context method using a single parser is lower than that of the without-context method in most cases. However, when the with-context method using a single parser is compared with the without-context method, the reduce rate of Cross brackets is larger than that of Bracket recall. This indicates that the with-context method using a single parser has an effect on reducing Cross brackets. In the case of the with-context method using a parser combination, the reduce rate of the Bracket recall from the

[*2] http://nlp.cs.nyu.edu/evalb/

Table 8 Examples of noun phrase structures. Brackets are represented by "[" and "]".

Parsing result by BERKELEY	(S (NP (DT The) (NN conductor) (NN pattern) (NN 14a)) (VP (VBZ is) (VP (VBN led) (PRT (RP out)) (PP (IN up) (PP (TO to) (NP (DT the) (JJ first) (NN side)))) (NP (NP (NN face) (NN 20b)) (PP (IN of) (NP (NN element) (CD 1))) (S (VP (TO to) (VP (VB be) (VP (ADVP (RB electrically)) (VBN connected) (PP (TO to) (NP (DT the) (JJ other) (NN terminal) (NN electrode) (CD 5)))))))))) (. .))
BERKELEY without context	[The conductor pattern 14a] is led out up to [**the first side**] [[**face 20b**] of [element 1] to be electrically connected to [the other terminal electrode 5]] .
BERKELEY and STANFORD with context	[[The [conductor pattern]] 14a] is led out up to [[the first side] face] 20b of [element 1] to be electrically connected to [the other [terminal electrode] 5] .

Table 9 Example of noun phrase structures including cross brackets. Brackets are represented by "[" and "]".

BERKELEY and STANFORD with context	[[[The cases] [24 , 26]] and covers (not shown) are molded integrally by **zinc**] **die casting** , respectively .

without-context method is smaller than that of the with-context method using a single parser. Then the with-context method using a parser combination is more effective in reducing Cross brackets. **Table 8** shows examples of noun phrases parsed by BERKELEY without context and determined by a combination of BERKELEY and STANFORD with context. The expression surrounded by cross brackets is underlined. The underlined expression crosses the noun phrase of the bolded section. There are no cross brackets in the with-context results. **Table 9** shows an example of noun phrases including cross brackets determined by a combination of BERKELEY and STANFORD with context. The bolded noun phrase crossed the underlined brackets. The results of the with-context method had multiple nested brackets, which degraded the All NPs Bracket precision of the with-context method.

We also investigated the structures in noun phrases obtained by the with-context method. There has been a proposal [33] to create a corpus of the structures in noun phrases. Based on this proposal, we manually annotated structures in noun phrases for 100 sentences randomly selected from the originally annotated 500 sentences. We calculated the agreement between the NP structures that were excluded from All NPs to obtain Selected NPs and the manually annotated structures in noun phrases for the 100 sentences. The results are shown in **Table 10**. From the Bracket precision in Table 10, approximately half of the structures agreed with the manually annotated structures. This indicates that the with-context method can obtain linguistic structures in the NP structures to some extent.

Here, we focus on the difference between the method using parser combinations and others. First, we check the difference between the method using a parser combination and the method using a single parser in the "with context" category. Shown in Table 7, the Bracket recall rates of the method using a parser combination are higher than the Bracket recall rates of the method using a single parser, and the Cross Brackets are comparable.

Next, we check the difference between the with-context method using a parser combination and the without-context method. Shown in Table 7, the Cross brackets of the with-context method using a parser combination are better (lower) than the Cross brackets of the without-context method, and the Bracket F-measure are comparable.

Table 10 Agreement of brackets in noun phrases determined by the method using context.

	Bracket precision	Bracket recall
COLLINS	56.1 (124/221)	20.0 (124/621)
COLLINS (modify)	55.4 (124/224)	20.0 (124/621)
CHARNIAK	48.8 (156/320)	25.1 (156/621)
STANFORD	51.5 (153/297)	24.6 (153/621)
BERKELEY	52.2 (157/301)	25.3 (157/621)
MST	59.2 (139/235)	22.4 (139/621)
MALT	62.2 (120/193)	19.3 (120/621)
MALT (train)	55.5 (112/202)	18.0 (112/621)
ENJU	58.1 (108/186)	17.4 (108/621)
BERKELEY & STANFORD	49.0 (179/365)	28.8 (179/621)
BERKELEY & MST	52.2 (182/349)	29.3 (182/621)
BERKELEY & ENJU	49.9 (167/335)	26.9 (167/621)
STANFORD & MST	52.0 (185/356)	29.8 (185/621)
STANFORD & ENJU	50.2 (165/329)	26.6 (165/621)
MST & ENJU	57.5 (158/275)	25.4 (158/621)

We think that these advantages produced the improvement in translation quality for the with-context method using a parser combination.

Determining the most appropriate syntactic structure of an input sentence using two parsers' outputs is not a trivial problem because of ambiguities in the merging of two structures into one appropriate structure. The with-context method (described in Sections 3.3 and 3.4) can merge two NP structures into one by resolving the ambiguity using statistical information. These advantages indicate that the with-context method is useful for using two parsers to determine NP structures in input sentences.

There is an inconsistency between the MST translation quality and the Selected NPs Bracket F-measure. The MST translation quality scored relatively high in Table 4, but the MST Selected NPs Bracket F-measure in Table 7 showed scores that were not relatively high. Dependency structures cannot represent all the NP boundaries of constituency structures. For example, NPs extracted from the dependency structure of "There is a cat on the mat" are "the mat" and "a cat on the mat" as shown in Fig. 4. "A cat" is not extracted as a noun phrase because both "a" and "on" are descendent nodes of "cat." The lack of ability of dependency structures to represent NP structures degrades the Bracket recall for dependency structures. We think this is the cause of the inconsistency between the MST translation quality and the Selected NPs Bracket F-measure. MST Selected NPs Bracket precision was close to the level of the constituency parsers, meaning

Table 11 Parse accuracy based on words in NP structures in patent sentences.

		Word F-measure	Word precision	Word recall
without context	COLLINS	80.86	87.80 (8861/10092)	74.94 (8861/11824)
	COLLINS (modify)	83.05	93.80 (8810/9392)	74.51 (8810/11824)
	CHARNIAK	94.38	92.52 (11387/12307)	96.30 (11387/11824)
	STANFORD	**94.98**	93.42 (11422/12227)	96.60 (11422/11824)
	BERKELEY	94.69	92.50 (11467/12397)	96.98 (11467/11824)
	MST	93.20	92.29 (11130/12060)	94.13 (11130/11824)
	MALT	90.13	87.40 (11001/12587)	93.04 (11001/11824)
	MALT (train)	91.42	90.31 (10944/12118)	92.56 (10944/11824)
	ENJU	93.57	92.70 (11169/12049)	94.46 (11169/11824)
with context	COLLINS	86.16	88.49 (9926/11217)	83.95 (9926/11824)
	COLLINS (modify)	88.54	94.23 (9873/10477)	83.50 (9873/11824)
	CHARNIAK	94.29	92.69 (11345/12240)	95.95 (11345/11824)
	STANFORD	94.86	93.64 (11365/12137)	96.12 (11365/11824)
	BERKELEY	94.59	92.71 (11415/12312)	96.54 (11415/11824)
	MST	92.76	92.38 (11014/11923)	93.15 (11014/11824)
	MALT	85.36	89.73 (9623/10724)	81.39 (9623/11824)
	MALT (train)	90.93	90.44 (10811/11954)	91.43 (10811/11824)
	ENJU	93.39	92.65 (11132/12015)	94.15 (11132/11824)
	BERKELEY & STANFORD	94.60	91.26 (11610/12722)	98.19 (11610/11824)
	BERKELEY & MST	94.07	91.97 (11384/12378)	96.28 (11384/11824)
	BERKELEY & ENJU	94.54	90.92 (11642/12804)	98.46 (11642/11824)
	STANFORD & MST	94.39	92.69 (11369/12266)	96.15 (11369/11824)
	STANFORD & ENJU	94.75	91.26 (11649/12765)	98.52 (11649/11824)
	MST & ENJU	93.57	91.85 (11274/12274)	95.35 (11274/11824)

that MST parsing accuracy is close to that of other constituency parsers.

For a fairer comparison between constituency parsers and dependency parsers, we checked the accuracy based on words. The accuracy was calculated as follows:

- Word precision = (number of words that are in NP structures of both gold NP brackets and parsing results)/(number of words that are in NP structures of parsing results).
- Word recall = (number of words that are in NP structures of both gold NP brackets and parsing results)/(number of words that are in NP structures of gold NP brackets).

We counted the number of words in NP structures without overlaps. For example, the number of words is 3 for $({}_{\mathrm{NP}}\, w_1\ ({}_{\mathrm{NP}}\, w_2\ w_3))$ where w_i is a word.

Bracket recall and Word recall are based on different things. Bracket recall evaluates NP structures in an NP structure. Bracket recall is based on brackets. On the other hand, Word recall does not evaluate NP structures in an NP structure but does evaluate the outmost NP structures, which can be handled by both constituency parsers and dependency parsers. Word recall is based on words. Although Word F-measure cannot evaluate NP structures included in other NP structures, it does offer a fairly comparable evaluation for the outmost NP structures.

Table 11 shows the parse accuracy based on words in NP structures. The numbers in parentheses for precision and recall represent the concrete numbers of words. MST's Word F-measure achieved a level comparable to the constituency parsers. This and low Cross brackets are thought to be the reasons that MST achieved a high translation quality.

Based on the analyses, we saw that parsing results require not only a high F-measure, but also low Cross brackets for patent machine translation.

Table 12 Comparison among manual annotation, automatic annotation using context, and non-annotation based on the effects of reordering constraints on 500 sentences.

	BLEU	NIST	WER
Baseline	37.97	7.8064	0.7080
BERKELEY	39.36	7.9217	0.6669
STANFORD	39.43	7.9136	0.6687
BERKELEY & STANFORD	39.60	7.9237	0.6623
Manual	**39.88**	**7.9705**	**0.6538**

In addition, we checked translation quality with human annotation. Here, we used the 500 human annotated sentences as the test data. Brackets of noun phrases were converted to zone tags. We compared the results with the "Baseline" which did not use zone tags, the top two parsers BERKELEY and STANFORD using context, and a combination of the two parsers. **Table 12** gives these results. "Manual" indicates the results with human annotations. As shown in Table 12, the correctly parsed results produced high translation quality.

5. Conclusion

We empirically compared the effects of seven parsers on patent machine translation. We used a phrase-based statistical machine translation method that used syntax structures in the source language for reordering constraints. We conducted experiments on English to Japanese patent translation using the NTCIR-8 patent translation task dataset. Parsers were found to be effective for patent machine translation reordering constraints. Most of the parsers, not only the probabilistic CFG parsers but also the dependency parsers and the HPSG parser, were effective when a noun phrase reordering constraint was used. When a method that determined noun phrases using the parsing results of context documents parsed by two parsers was applied, the effectiveness increased. The results of this research substantiating how effective each parser is in patent machine translation can be of service to those who need patent machine translation when they are selecting which parser would be appropriate for patent machine translation. Our future work will investigate the effects of parsers on other statistical machine translation methods using syntax for patent machine translation.

References

[1] Buchholz, S. and Marsi, E.: CoNLL-X Shared Task on Multilingual Dependency Parsing, *Proc. 10th Conference on Computational Natural Language Learning* (*CoNLL-X*), New York City, Association for Computational Linguistics, pp.149–164 (2006) (online), available from ⟨http://www.aclweb.org/anthology/W/W06/W06-2920⟩.

[2] Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C. and Schroeder, J.: Further Meta-Evaluation of Machine Translation, *Proc. 3rd Workshop on Statistical Machine Translation*, Columbus, Ohio, Association for Computational Linguistics, pp.70–106 (2008) (online), available from ⟨http://www.aclweb.org/anthology/W/W08/W08-0309⟩.

[3] Charniak, E.: A Maximum-Entropy-Inspired Parser, *Proc. 1st North American chapter of the Association for Computational Linguistics conference*, pp.132–139 (2000).

[4] Cherry, C.: Cohesive Phrase-Based Decoding for Statistical Machine Translation, *Proc. 46th Annual Meeting of the Association for Computational Linguistics with the Human Language Technology*, pp.72–80 (2008).

[5] Collins, M.: Three Generative, Lexicalised Models for Statistical Parsing, *Proc. 35th Annual Meeting of the Association for Computational Linguistics*, Madrid, Spain, Association for Computational Linguistics, pp.16–23 (online), DOI: 10.3115/976909.979620 (1997).

[6] Cowan, B., Kučerová, I. and Collins, M.: A Discriminative Model for Tree-to-Tree Translation, *Proc. 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, Association for Computational Linguistics, pp.232–241 (2006) (online), available from ⟨http://www.aclweb.org/anthology/W/W06/W06-1628⟩.

[7] Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics, *Proc. 2nd International Conference on Human Language Technology Research*, pp.138–145 (2002).

[8] Eck, M. and Hori, C.: Overview of the IWSLT 2005 Evaluation Campaign, *Proc. International Workshop on Spoken Language Translation* (2005).

[9] Eisner, J.M.: Three New Probabilistic Models for Dependency Parsing: An Exploration, *Proc. 16th International Conference on Computational Linguistics*, pp.340–345 (1996).

[10] Frantzi, K.T. and Ananiadou, S.: Extracting Nested Collocations, *Proc. COLING 1996*, pp.41–46 (1996).

[11] Fujii, A., Utiyama, M., Yamamoto, M., Utsuro, T., Ehara, T., Echizen-ya, H. and Shimohata, S.: Overview of the Patent Translation Task at the NTCIR-8 Workshop, *Proc. NTCIR-8 Workshop Meeting*, pp.371–376 (2010).

[12] Huang, L., Knight, K. and Joshi, A.: Statistical Syntax-Directed Translation with Extended Domain of Locality, *Proc. 7th Conference of the Association for Machine Translation of the Americas*, Cambridge, Massachusetts, USA, AMTA, pp.66–73 (2006).

[13] Klein, D. and Manning, C.D.: Accurate Unlexicalized Parsing, *Proc. 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, Association for Computational Linguistics, pp.423–430 (online), DOI: 10.3115/1075096.1075150 (2003).

[14] Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation, *Proc. EMNLP 2004*, Lin, D. and Wu, D. (Eds.), Barcelona, Spain, Association for Computational Linguistics, pp.388–395 (2004).

[15] Koehn, P. and Haddow, B.: Edinburgh's Submission to all Tracks of the WMT 2009 Shared Task with Reordering and Speed Improvements to Moses, *Proc. 4th Workshop on Statistical Machine Translation*, pp.160–164 (2009).

[16] Liu, Y., Liu, Q. and Lin, S.: Tree-to-String Alignment Template for Statistical Machine Translation, *Proc. 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, Association for Computational Linguistics, pp.609–616 (online), DOI: 10.3115/1220175.1220252 (2006).

[17] Liu, Y., Lü, Y. and Liu, Q.: Improving Tree-to-Tree Translation with Packed Forests, *Proc. Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, Association for Computational Linguistics, pp.558–566 (2009) (online), available from ⟨http://www.aclweb.org/anthology/P/P09/P09-1063⟩.

[18] Marcus, M.P., Santorini, B. and Marcinkiewicz, M.A.: Building a Large Annotated Corpus of English: The Penn Treebank, *Computational Linguistics*, Vol.19, No.2, pp.313–330 (1993).

[19] Marton, Y. and Resnik, P.: Soft Syntactic Constraints for Hierarchical Phrased-Based Translation, *Proc. 46th Annual Meeting of the Association for Computational Linguistics with the Human Language Technology*, pp.1003–1011 (2008).

[20] McDonald, R. and Pereira, F.: Online Learning of Approximate Dependency Parsing Algorithms, *Proc. 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp.81–88 (2006).

[21] Miyao, Y., Sagae, K., Sætre, R., Matsuzaki, T. and Tsujii, J.: Evaluating contributions of natural language parsers to protein-protein interaction extraction, *Bioinformatics*, Vol.25, No.3, pp.394–400 (2009).

[22] Miyao, Y. and Tsujii, J.: Feature Forest Models for Probabilistic HPSG Parsing, *Computational Linguistics*, Vol.34, No.1, pp.81–88 (2008).

[23] Niessen, S., Och, F.J., Leusch, G. and Ney, H.: An evaluation tool for machine translation: Fast evaluation for MT research, *Proc. 2nd International Conference on Language Resources and Evaluation*, pp.39–45 (2000).

[24] Nivre, J.: An Efficient Algorithm for Projective Dependency Parsing, *Proc. 8th International Workshop on Parsing Technologies* (*IWPT 03*), pp.149–160 (2003).

[25] Nivre, J.: Algorithms for Deterministic Incremental Dependency Parsing, *Computational Linguistics*, Vol.34, No.4, pp.513–553 (2008).

[26] Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S. and Yuret, D.: The CoNLL 2007 Shared Task on Dependency Parsing, *Proc. CoNLL Shared Task Session of EMNLP-CoNLL 2007*, Prague, Czech Republic, Association for Computational Linguistics, pp.915–932 (2007) (online), available from ⟨http://www.aclweb.org/anthology/D/D07/D07-1096⟩.

[27] Onishi, T., Utiyama, M. and Sumita, E.: Reordering Constraint Based on Document-Level Context, *Proc. 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA, Association for Computational Linguistics, pp.434–438 (2011) (online), available from ⟨http://www.aclweb.org/anthology/P11-2076⟩.

[28] Papineni, K., Roukos, S., Ward, T. and Zhu, W.-J.: Bleu: A Method for Automatic Evaluation of Machine Translation, *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pp.311–318 (2002).

[29] Petrov, S. and Klein, D.: Improved Inference for Unlexicalized Parsing, *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Proc. Main Conference*, Rochester, New York, Association for Computational Linguistics, pp.404–411 (2007) (online), available from ⟨http://www.aclweb.org/anthology/N/N07/N07-1051⟩.

[30] Pollard, C. and Sag, I.A.: *Head-Driven Phrase Structure Grammar*, University of Chicago Press (1994).

[31] Quirk, C. and Corston-Oliver, S.: The impact of parse quality on syntactically-informed statistical machine translation, *Proc. 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, Association for Computational Linguistics, pp.62–69 (2006) (online), available from ⟨http://www.aclweb.org/anthology/W/W06/W06-1608⟩.

[32] Tsuruoka, Y. and Tsujii, J.: Bidirectional Inference with the Easiest-First Strategy for Tagging Sequence Data, *Proc. Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada, Association for Computational Linguistics, pp.467–474 (2005) (online), available from ⟨http://www.aclweb.org/anthology/H/H05/H05-1059⟩.

[33] Vadas, D. and Curran, J.: Adding Noun Phrase Structure to the Penn Treebank, *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, Association for Computational Linguistics, pp.240–247 (2007) (online), available from ⟨http://www.aclweb.org/anthology/P07-1031⟩.

[34] Xiao, T., Zhu, J., Zhang, H. and Zhu, M.: An Empirical Study of Translation Rule Extraction with Multiple Parsers, *Coling 2010: Posters*, Beijing, China, Coling 2010 Organizing Committee, pp.1345–1353 (2010) (online), available from ⟨http://www.aclweb.org/anthology/C10-2154⟩.

[35] Xiong, D., Zhang, M. and Li, H.: Learning Translation Boundaries for Phrase-Based Decoding, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp.136–144 (2010).

[36] Yamamoto, H., Okuma, H. and Sumita, E.: Imposing Constraints from the Source Tree on ITG Constraints for SMT, *Proc. ACL-08: HLT 2nd Workshop on Syntax and Structure in Statistical Translation* (*SSST-2*), pp.1–9 (2008).

[37] Zhang, H., Wang, H., Xiao, T. and Zhu, J.: The impact of parsing accuracy on syntax-based SMT, *Proc. International Conference on Natural Language Processing and Knowledge Engineering* (*NLP-KE*), pp.1–4 (2006).

[38] Zhang, M., Jiang, H., Aw, A., Li, H., Tan, C.L. and Li, S.: A Tree Sequence Alignment-based Tree-to-Tree Translation Model, *Proc. ACL-08: HLT*, Columbus, Ohio, Association for Computational Linguis-

tics, pp.559–567 (2008) (online), available from ⟨http://www.aclweb.org/anthology/P/P08/P08-1064⟩.

Isao Goto received his M.E. degree in electrical engineering from Waseda University in 1997. He is a researcher of National Institute of Information and Communications Technology. His research interests include machine translation.

Masao Utiyama received his B.S. degree in computer science from University of Tsukuba, Japan in 1992, M.S. degree in computer science from University of Tsukuba in 1994, Ph.D. degree in engineering from University of Tsukuba in 1997. From 1997 to 1999, he was a research associate at Shinshu University, Japan. He has been a member of National Institute of Information and Communications Technology (NICT), Japan since 1999. He is a senior researcher at NICT.

Takashi Onishi received his M.E. degree in Information and Communication Engineering from the University of Tokyo in 2006. He was a researcher of National Institute of Information and Communications Technology from 2009 to 2011. He is currently a researcher of NEC Information and Media Processing Laboratories. His research interests include machine translation.

Eiichiro Sumita received his M.S. degree in computer science from the University of Electro-Communications in 1982 and Ph.D. degree in engineering from Kyoto University in 1999. Dr. Sumita is the group leader of NICT/MASTAR Project/Language Translation Group, and the visiting professor of Kobe University. His research interests include machine translation and e-Learning.