Preordering using a Target-Language Parser via Cross-Language Syntactic Projection for Statistical Machine Translation

ISAO GOTO, National Institute of Information and Communications Technology, NHK, and Kyoto University
MASAO UTIYAMA and EIICHIRO SUMITA, National Institute of Information and Communications Technology
SADAO KUROHASHI, Kyoto University

When translating between languages with widely different word orders, word reordering can present a major challenge. Although some word reordering methods do not employ source-language syntactic structures, such structures are inherently useful for word reordering. However, high-quality syntactic parsers are not available for many languages. We propose a preordering method using a target-language syntactic parser to process source-language syntactic structures without a source-language syntactic parser. To train our preordering model based on ITG, we produced syntactic constituent structures for source-language training sentences by (1) parsing target-language training sentences, (2) projecting constituent structures of the target-language sentences to the corresponding source-language sentences, (3) selecting parallel sentences with highly synchronized parallel structures, (4) producing probabilistic models for parsing using the projected partial structures and the Pitman-Yor process, and (5) parsing to produce full binary syntactic structures maximally synchronized with the corresponding target-language syntactic structures, using the constraints of the projected partial structures and the probabilistic models. Our ITG-based preordering model is trained using the produced binary syntactic structures and word alignments. The proposed method facilitates the learning of ITG by producing highly synchronized parallel syntactic structures based on cross-language syntactic projection and sentence selection. The preordering model jointly parses input sentences and identifies their reordered structures. Experiments with Japanese–English and Chinese–English patent translation indicate that our method outperforms existing methods, including string-to-tree syntax-based SMT, a preordering method that does not require a parser, and a preordering method that uses a source-language dependency parser.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Machine translation*

General Terms: Theory, Algorithms, Design, Experimentation

Additional Key Words and Phrases: Preordering, syntactic projection, constituent structure, inversion transduction grammar

ACM Reference Format:
Goto, I., Utiyama, M., Sumita, E., and Kurohashi, S. 2015. Preordering using a target-language parser via cross-language syntactic projection for statistical machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 14, 3, Article 13 (June 2015), 23 pages.
DOI: http://dx.doi.org/10.1145/2699925

Authors' addresses: I. Goto (corresponding author), NHK Science & Technology Research Laboratories, 1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan; email: goto.i-es@nhk.or.jp; M. Utiyama and E. Sumita, Multilingual Translation Laboratory, National Institute of Information and Communications Technology, 3-5 Hikaridai, Keihanna Science City, Kyoto 619-0289, Japan; emails: {mutiyama, eiichiro.sumita}@nict.go.jp; S. Kurohashi, Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan; email: kuro@i.kyoto-u.ac.jp.

2375-4699/2015/06-ART13 $15.00
DOI: http://dx.doi.org/10.1145/2699925

1. INTRODUCTION

Estimating the appropriate word order for a target language is one of the most difficult problems in statistical machine translation (SMT). This is particularly true when translating between languages with widely different word orders, such as Japanese and English. To address this, a large body of research has been conducted on word reordering, for instance: lexicalized reordering model [Tillman 2004] for phrase-based SMT, hierarchical phrase-based SMT [Chiang 2007], syntax-based SMT [Yamada and Knight 2001], preordering [Xia and McCord 2004], and postordering [Sudoh et al. 2011b].

The preordering framework is useful for word reordering because it can use source-language syntactic structures in a simple way. Specifically, a preordering method using source-language syntactic structures for English-to-Japanese translation has been confirmed to be highly effective [Goto et al. 2011; Sudoh et al. 2011a]. Existing preordering methods that use source-language syntactic structures require a source-language syntactic parser. Unfortunately, high-quality syntactic parsers are not available for many languages.

Preordering methods that do not require a parser are useful in cases where no source-language syntactic parser is available [DeNero and Uszkoreit 2011; Neubig et al. 2012]. Such methods produce preordering rules using word alignments. However, these preordering rules do not use syntactic structures, which are an essential factor in deciding word order. Therefore, the use of syntactic structures is a major challenge for preordering methods that do not require a source-language syntactic parser.

In this article, we propose a novel preordering approach that uses syntactic structures by employing a target-language syntactic parser without requiring a source-language parser. A high-quality target-language constituency parser will be useful for preordering. Source-language syntactic structures and corresponding target-language syntactic structures are expected to be similar in a parallel corpus [Hwa et al. 2005]. The proposed method relies on this expectation. We project target-language syntactic constituent structures in a parallel corpus onto their corresponding source-language sentences through word alignments, which produces partial syntactic structures where the words are from the source language but the phrase labels are from the target-language syntax. We then select parallel sentences with highly synchronized parallel syntactic structures based on the projection. We construct a probabilistic context-free grammar (CFG) model and a probabilistic model for unsupervised part-of-speech (POS) tagging using the partial syntactic structures of the selected parallel sentences and the Pitman-Yor process [Pitman and Yor 1997]. We then parse the source-language training sentences to produce full binary syntactic tree structures using the produced probabilistic models with the projected partial syntactic structure constraints. A preordering model based on inversion transduction grammar (ITG) [Wu 1997] is learned using the full binary syntactic constituent structures of the source-language sentences and word alignments. Input sentences are parsed using the ITG-based preordering model, then their syntactic structures and reordered structures are identified jointly.

Our main contributions are (i) a new effective framework for preordering using a target-language syntactic parser that does not require a source-language syntactic parser, (ii) methods that facilitate the learning of ITG by producing highly synchronized parallel syntactic structures based on cross-language syntactic projection and sentence selection, (iii) a simple method for producing full binary syntactic constituent structures of source-language sentences from the constituent structures of the corresponding target-language sentences using the Pitman-Yor process, and (iv) an

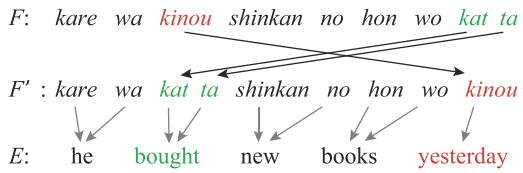

Fig. 1. Example of preordering for Japanese–English translation.

empirical confirmation of the efficacy of Japanese–English and Chinese–English patent translation.

There is a need for translation in situations where all the following are true: (1) a high-quality target-language parser is available, (2) a high-quality source-language parser is not available, and (3) the source-language word order and the target-language word order are largely different, such as in subject-object-verb (SOV) and subject-verb-object (SVO) languages. We propose a method that can be applied in such situations. In our experiments on Japanese–English and Chinese–English translation using the patent data from the NTCIR-9 and NTCIR-10 Patent Machine Translation Tasks [Goto et al. 2011, 2013a], we were able to significantly improve translation quality, as measured by both RIBES [Isozaki et al. 2010] and BLEU [Papineni et al. 2002]. Our method is superior to phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, an existing preordering method without a parser, and an existing preordering method that uses a source-language dependency parser.

The rest of this article is organized as follows: Section 2 describes the preordering framework and previous work; Section 3 provides an overview of the proposed method; Section 4 explains the training method; Section 5 describes the preordering method; Section 6 reports and discusses the experimental results; and Section 7 concludes the article.

2. PREORDERING FOR SMT

Machine translation is defined as the process in which a source-language sentence F is transformed into a target-language sentence E. During this process, word reordering is often necessary. More specifically, long-distance word reordering is necessary when translating between languages with widely different word orders.

The syntactic structure of F is useful for long-distance word reordering. Preordering is an SMT method in which the syntactic structure of F can be handled in a simple way. This is the approach that we take in this article. In the preordering approach, translation is performed as a two-step process, as shown in Figure 1. In the first part of the process, F is reordered to F', which is a source-language word sequence with almost the same word order as the target language. In the second part of the process, F' is translated into E using an SMT method such as phrase-based SMT, which can produce accurate translations when only local reordering is required.

The preordering framework has been widely studied. In most preordering research, the reordering of words is conducted using reordering rules and the syntactic structure of F that is obtained using a source-language syntactic parser. Reordering rules are produced automatically [Xia and McCord 2004; Li et al. 2007; Habash 2007; Dyer and Resnik 2010; Ge 2010; Genzel 2010; Visweswariah et al. 2010; Wu et al. 2011a, 2011b; Lerner and Petrov 2013] or manually [Collins et al. 2005; Wang et al. 2007; Ramanathan et al. 2008; Badr et al. 2009; Xu et al. 2009; Isozaki et al. 2012; Hoshino et al. 2013].

However, if a source-language syntactic parser is not available, then these methods cannot be applied. In such cases, preordering methods that do not require a parser

Table I. Comparison of Preordering Methods based on the Necessity of Syntactic Parsers for Source and Target Languages

Preordering methods	Parser	
	Source	Target
Most existing methods	✓	
[Neubig et al. 2012]		
Proposed method		✓

are useful [Tromble and Eisner 2009; Visweswariah et al. 2011; DeNero and Uszkoreit 2011; Neubig et al. 2012; Khapra et al. 2013].

Methods that induce a parser deserve particular mention because they are similar to our approach. DeNero and Uszkoreit [2011] and Neubig et al. [2012] induce a nonsyntactic parser automatically using a parallel corpus with word alignments. The induced nonsyntactic parser is used to produce binary tree structures of input sentences. The input sentences are then preordered based on the binary tree structures and bracketing transduction grammar (BTG) [Wu 1997]. The resulting binary tree structures are nonsyntactic structures. In contrast, our method utilizes syntactic structures for preordering via a target-language syntactic parser.

Compared with the nonsyntactic structures that are produced by a nonsyntactic parser based on BTG [Neubig et al. 2012], syntactic structures are thought to be superior when making decisions about word reordering for the following two reasons.

— In syntactic structures, a subtree span is expected to be consistent with the span of an expression that has cohesive meanings. For example, clauses are thought to be spans with cohesive meanings, and clauses are expressed by subtrees in syntactic structures. In contrast, in nonsyntactic structures produced by BTG, a subtree span is not always consistent with the span of an expression with cohesive meanings.
— Syntactic structures are richer in terms of information than nonsyntactic structures produced by BTG. Syntactic structures have many phrase label types. In contrast, BTG has only one phrase label type.

Therefore, syntactic structures are thought to be useful when performing word reordering for preordering methods.

Table I compares the necessity of syntactic parsers in existing preordering methods for source and target languages with that of the proposed preordering method. In some cases, a high-quality syntactic parser is not available for the source language, but a high-quality syntactic parser is available for the target language, while the source-language word order and the target-language word order are largely different, such as with SOV and SVO languages. Our method is applicable in these cases.

3. OVERVIEW OF THE PROPOSED METHOD

In this section, we provide an overview of our preordering method.

Our preordering method processes syntactic structures using a target-language parser even when a source-language parser is not available. The syntactic structures of source-language sentences and the syntactic structures of the corresponding target-language sentences are expected to be similar in a parallel corpus [Hwa et al. 2005]. We used this expectation to produce syntactic constituent structures of source-language sentences that are similar to the syntactic constituent structures of the corresponding target-language sentences.

To effectively learn ITG or synchronous CFG [Aho and Ullman 1969], it is important that the level of synchrony between parallel syntactic structures is high. This is because ITG or synchronous CFG rules are learned from the synchronized parts of

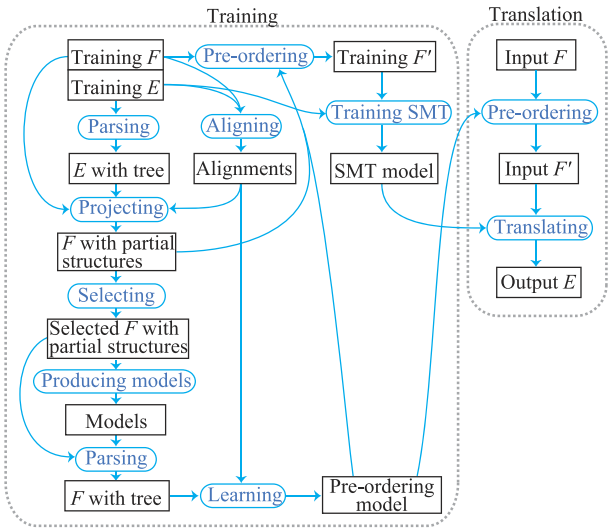

Fig. 2. Overview of our method.

parallel syntactic structures, and such rules cannot be generated from nonsynchronized parts of parallel syntactic structures. That is, it is difficult to effectively learn ITG or synchronous CFG rules from parallel syntactic structures with a low level of synchrony.

Our proposed method facilitates the learning of ITG by producing highly synchronized parallel structures which are then used as training data for ITG, based on the following methods. (i) We produce source-language syntactic structures, which are maximally synchronized with the corresponding target-language syntactic structures, by cross-language syntactic projection. (ii) We select parallel sentences with highly synchronized parallel structures based on cross-language syntactic projection.

Figure 2 shows an overview of our method. Our preordering model is trained via the following steps:

(1) parsing target-language sentences in the parallel training corpus using a syntactic parser to obtain binary[1] tree structures;
(2) projecting the syntactic structures of the training target-language sentences onto the corresponding source-language sentences through word alignments (Section 4.1);
(3) selecting parallel sentences with highly synchronized parallel structures based on the projection (Section 4.2);
(4) producing a probabilistic CFG model and a probabilistic model for unsupervised POS tagging for the source-language using the projected partial syntactic structures (Section 4.3);
(5) parsing the training source-language sentences to produce full binary syntactic tree structures that are highly synchronized with the corresponding target-language syntactic structures. This is conducted using the produced probabilistic models and the projected partial syntactic structures (Section 4.4);
(6) learning the preordering model based on ITG using the full binary syntactic tree structures and word alignments (Section 4.5).

[1]Head binarization is suitable for our method. When trees are not binary trees, our method is applicable if a binarization method is applied.

Input sentences are preordered by jointly parsing and identifying reordering using the ITG-based preordering model.

Our main contribution is an effective new framework for preordering using a target-language parser. Additionally, we propose a new parsing method for source languages that does not require a source-language parser or a source-language POS tagger.

Jiang et al. [2011] developed a method for projecting constituent structures between languages. There are two main differences between our method and theirs. One is the method for estimating CFG rule probabilities. They count the number of CFG rules appearing in tree candidates in each sentence for maximum likelihood estimation of CFG rule probabilities. In this process, they assume a uniform distribution over the projected tree candidates and then calculate the expected counts under this assumption. This looks like a single iteration of the EM algorithm. However, their assumption is incorrect. The expected counts of CFG rules in probable tree candidates should be larger than those of CFG rules in unlikely tree candidates. Our method solves this problem by simply engaging the Pitman-Yor process. The other difference between our method and that of Jiang et al. [2011] is in the requirements. Their method requires source-language POS tags that are produced by a POS tagger. In contrast, our method does not require source-language POS tags.

In Section 4, we describe the training method of our preordering model in detail. In Section 5, we explain the methods for preordering input sentences and the training sentences.

4. TRAINING THE PREORDERING MODEL

In this section, we will explain the five components of the training method for our preordering model, which is employed after the parsing of target-language training sentences is complete.

4.1. Projecting Partial Syntactic Structures

Through word alignments, we project the binary syntactic constituent structures of the target-language sentences in the training parallel corpus onto the corresponding source-language sentences. Partial syntactic structures of the source-language sentences are then obtained. An example of this projection is shown in Figure 3.

The projection is conducted by (1) identifying the span in F corresponding to a subtree span in E through word alignments, and (2) adding the root phrase label of the subtree in E to the span in F. A span in F is the span from the leftmost position to the rightmost position of the source words that are aligned to the target word(s) in the subtree in E. The root phrase label of a projected subtree in E is added to the projected span in F. Note that if any nonaligned words are adjacent to the span in F, then there is a chance that these words should be contained in the span. That is, when there are nonaligned words adjacent to a span in F, there are ambiguities in the span. A projected span that does not contain the adjacent nonaligned words, which is represented by a horizontal solid line in Figure 3, is called a *minimal projected span*. A phrase label is added to a minimal projected span. In Figure 3, a phrase label is not projected to the span covering *kat ta* because the corresponding target expression (bought) is a word instead of a phrase and does not have a phrase label.

To ensure that the projected structures can compose tree structures and consist solely of high-quality structures, we do not project any subtree spans in E when their corresponding spans in F conflict with one other. Here, the conflict is that two subtree spans that do not overlap in E do overlap, except for non-aligned words, when they are projected to F. That is, we only project the subtree spans of E whose corresponding spans of F are also continuous and do not conflict with one other. We choose to project none of the conflicted spans in E.

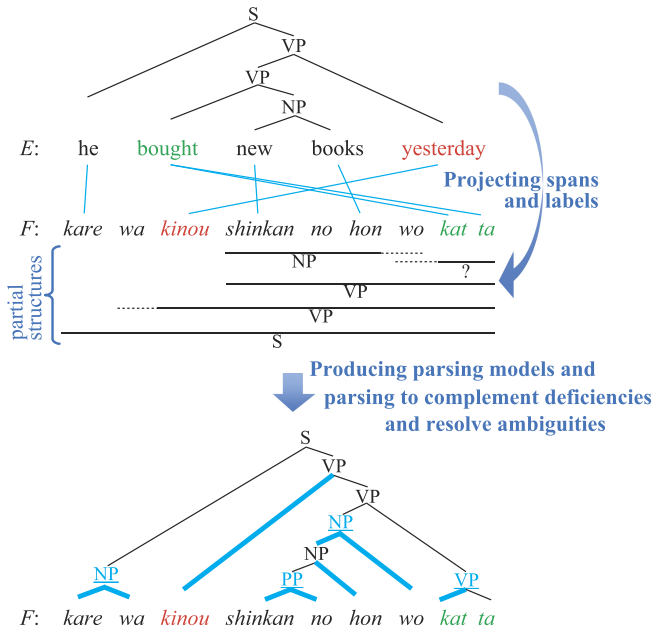

Fig. 3. Example of projecting syntactic structures from E to F and producing a full binary tree structure. The lines between the words in E and the words in F represent word alignments. The horizontal lines represent projected spans and the labels under the horizontal solid lines represent their phrase labels. The dotted lines represent ambiguities in the spans. In the tree structure of F, shown at the bottom part of the figure, the parts that are complemented or resolved ambiguities are represented by underlines for phrase labels and bold lines for structures, which are in blue.

4.2. Selecting Parallel Sentences with Highly Synchronized Structures

We use the projected spans to select parallel sentences with highly synchronized parallel structures. The selected sentences are then used to produce probabilistic models for parsing and full binary tree structures. Let r_1 be the span coverage rate of the projected partial syntactic structures for a source-language sentence. The r_1 for each source-language sentence is calculated by dividing the number of projected spans by the number of words in the sentence minus one.[2] When r_1 is high, the level of synchrony between parallel syntactic structures is high because the projected spans represent the parts that synchronize between the languages without conflict. The source-language training sentences are sorted based on r_1. We select top-ranked unique source-language sentences with high r_1 as the data for the processing described in Sections 4.3 to 4.5. In our translation experiments in Section 6, we selected the top 0.1 million unique source-language sentences.

4.3. Producing Probabilistic Models for Parsing

The projected structures are usually partial structures. As full binary tree structures are required for learning our preordering model, we produce probabilistic models for parsing source-language sentences to produce full binary tree structures.

We will now discuss in detail our method for producing probabilistic models for parsing. The inputs are a source-language sentence F and projected partial syntactic structures of F, described in Section 4.1. The following task characteristics enable the use of

[2]The number of spans in a full binary tree is the number of words in a sentence minus one.

a simple model to produce full binary tree structures. (i) Partial structures are given. (ii) The set of phrase labels is predefined. We also predefine the number of types of POS tags. POS tags for each word are induced automatically.[3]

We build a probabilistic context-free grammar (CFG) model for parsing source-language sentences. We use the Pitman-Yor process (PY) [Pitman and Yor 1997][4] to build the model because it has a "rich-get-richer" property[5] that suits the process of learning a model from partially annotated structures. This is because information contained in annotated structures can be used[6] to infer structures that are not annotated. We also build a probabilistic model for unsupervised POS tagging using the Pitman-Yor process.

A probabilistic CFG is defined by the 4-tuple $G = (\mathcal{F}, V, S, \mathcal{R})$, where $\mathcal{F}$ is the set of terminals, which are source-language words in the training data, V is the set of nonterminals, $S \in V$ is a designated start symbol, and $\mathcal{R}$ is a set of rules. A CFG rule $x \rightarrow \alpha \in \mathcal{R}$ used in this process consists of $x \in V$ and α, which consists of two elements of V. V is defined as $V = \mathcal{L} \cup \mathcal{T}$, where $\mathcal{L}$ is the set of phrase labels for the target-language syntax, $\mathcal{T} = \{1, 2, ..., |\mathcal{T}|\}$ is the set of source-language POS tags represented by numbers, where $|\mathcal{T}|$ is the number of POS tag types, and $\mathcal{L} \cap \mathcal{T} = \varnothing$. Let $f \in \mathcal{F}$ be a source-language word and $F = f_1 f_2 .. f_m$. The probability of a derivation tree D is defined as the product of the probabilities of its component CFG rules and the probabilities of words given their POS tags, as follows.

$$P(D) = \prod_{x \rightarrow \alpha \in \mathcal{R}} P(\alpha|x)^{c(x \rightarrow \alpha, D)} \prod_{i=1}^{m} P(f_i|t_i), \tag{1}$$

where $c(x \rightarrow \alpha, D)$ is the number of times $x \rightarrow \alpha$ is used for the derivation D, $P(\alpha|x)$ is the probability of generating α given its root phrase label x, $t \in \mathcal{T}$ is a POS tag, index i of t indicates the position in F, and $P(f|t)$ is the probability of generating f given its POS tag t. The designated phrase label, S, is used for the phrase label of the root node of a tree.

Our PY models represent probability distributions over CFG rules or source-language words, as follows.

$$P(\alpha|x) \sim \mathrm{PY}_x(d_{\mathrm{cfg}}, \theta_{\mathrm{cfg}}, P_{\mathrm{base}}(\alpha|x)) \text{ and}$$
$$P(f|t) \sim \mathrm{PY}_t(d_{\mathrm{tag}}, \theta_{\mathrm{tag}}, P_{\mathrm{base}}(f|t)),$$

where d_{cfg}, θ_{cfg}, d_{tag}, and θ_{tag} are hyperparameters for the PY models. The hyperparameters are optimized via the auxiliary variable technique [Teh 2006].[7]

[3]POS tags are also thought to be projectable. However, some POS tags cannot be projected. For an example regarding translation between English and Japanese, determiners exist in English but do not exist in Japanese, and post positions exist in Japanese but not in English. In addition, a method that does not project POS tags is simpler than a method that projects POS tags.

[4]Readers unfamiliar with PY can refer to Teh [2006] for a detailed description and estimation method for PY.

[5]When one observation is sampled, then the probability of the posterior distribution for that observation becomes larger than that of the prior distribution.

[6]If a CFG rule appears many times in the annotated structures and has been sampled many times, then it is highly likely that the CFG rule will be sampled as a partial structure that is not annotated.

[7]We put a prior distribution of Beta$(1, 1)$ on d_{cfg} and d_{tag} and a prior distribution of Gamma$(1, 1)$ on θ_{cfg} and θ_{tag}.

The backoff probability distributions, $P_{\text{base}}(\alpha|x)$ and $P_{\text{base}}(f|t)$, are uniform, as follows.

$$P_{\text{base}}(\alpha|x) = \frac{1}{|V|^2} \text{ and}$$
$$P_{\text{base}}(f|t) = \frac{1}{|\mathcal{F}|},$$

where $|V|$ is the number of nonterminal types and $|\mathcal{F}|$ is the lexicon size of the source-language words in the training data. As our CFG rule has two leaf nodes, the number of pair nonterminal node types is $|V|^2$.

Sampling for building the distributions is conducted according to Equation (1) with the following constraints. When projected spans are present, we constrain the sampling such that only the derivation trees that do not conflict with the projected spans are sampled. Here, the conflict is that both a subtree span in the tree derivation and a projected span partially overlap each other. When there is an ambiguity in a projected span, which comprises the minimal projected span and any number of adjacent unaligned word(s), the constraints are as follows. If a sample (a *subtree* span in the tree derivation) does not conflict with the minimal projected span, then the minimal projected span is used as the constraint for the sample. Otherwise, the whole span (the minimal projected span and its adjacent unaligned word(s)) is used as the constraint for the sample. When the projected phrase label for a subtree span in a derivation tree is present, we constrain the sampling such that only the projected phrase label is sampled.

We use a sentence-level blocked Gibbs sampler based on a dynamic programming algorithm [Johnson et al. 2007]. The sampler consists of the following two steps: for each sentence, (1) inside probabilities [Lari and Young 1991] are calculated from the bottom up using the CYK algorithm, and (2) a tree is sampled from the top down according to the inside probabilities for each CFG rule. In the first step, when we calculate the inside probabilities for each phrase label in each cell of the triangular table of the CYK algorithm and save the inside probabilities, we also save the inside probabilities for each CFG rule. In the second step, we sample a CFG rule according to the inside probabilities for the CFG rules in each cell from the top down. To reduce computational costs, we only use N-best POS tags for each word when the inside probabilities are calculated. In our experiment in Section 6, we used five-best POS tags for each word.

The computational complexity of producing probabilistic models for parsing is linearly proportional to the number of training sentences when the data properties are identical except for the data size. The computational cost depends on the amount of nonconstrained parts in the data. When the amount of nonconstrained parts becomes larger, the computational cost becomes larger.

4.4. Parsing to Produce Full Binary Tree Structures

After the probability distributions of the PY models are built, we parse the source-language sentences to produce full binary tree structures that are maximally synchronized with the corresponding target-language structures. The parsing complements deficiencies and resolves ambiguities in the projected partial structures. The deficiencies are insufficient spans or phrase labels in the projected spans and labels to construct a full binary tree structure. The ambiguities of spans are shown as horizontal dotted lines in Figure 3, which cover nonaligned words adjacent to minimal projected spans. We calculate the maximum likelihood full binary tree structures based on the CYK algorithm within the constraints of the minimal projected spans and their phrase labels, using the produced probabilistic CFG model and the produced probabilistic

model for unsupervised POS tagging. The probability for a derivation tree is calculated using Equation (1). The constraints are the same constraints used for sampling when building the probabilistic models. The resulting full binary tree structures comprise the phrase labels of the target-language syntax. An example of the production of a full binary tree structure is shown in Figure 3.

Note that when the full binary trees are generated, all of the minimal projected spans are not necessarily included in the full binary trees if the projected spans have ambiguities. If nonaligned words are located adjacent to the projected spans, then there may be cases in which some minimal projected spans are not included in the full binary tree. For example, when a minimal projected span is $(f_1 f_2) f_3$ and f_3 is a nonaligned word where parentheses denote a span, a full binary tree may be $(f_1 (f_2 f_3))$. This tree does not include the minimal projected span because there are ambiguities in the span when a nonaligned word is adjacent to the span, as explained in Section 4.1.

4.5. Learning the Preordering Model

Learning of our preordering model uses the full binary tree structures of source-language sentences and word alignments.

The preordering model is a model based on two fundamental frameworks [Goto et al. 2013b]: (i) parsing using probabilistic CFG and (ii) the inversion transduction grammar (ITG) [Wu 1997]. In this article, the model combining (i) and (ii) is called the *ITG parsing model* and parsing using ITG is called *ITG parsing*. We use the ITG parsing model for preordering while Goto et al. [2013b] used this model for postordering.

To obtain the training data for the preordering model, we first obtain the reordered structure that produces the word order of F' most similar to the word order of the corresponding E using their word alignments. Figure 4 shows an example of the tree structure of F' calculated from the tree structure of F and word alignments. Reordering is conducted by swapping child nodes in the binary tree structure of F so that Kendall's τ is maximized between F' and E.[8] For each node, we decide whether its child nodes are swapped or not. This decision is made deterministically from the bottom up. The algorithm of this maximization can be expressed as $O(m^2 \log m)$ in complexity, where m is a sentence length. This is because the computational complexity of Kendall's τ can be expressed as $O(m \log m)$ [Knight 1966] for each node in a binary tree. When the scores for candidates are the same[9], we retain the original order.

The nodes whose child nodes are swapped to transform F into F' are then annotated with an "_SW" suffix (indicating "swap/inversion"), and other nodes with two child nodes are annotated with an "_ST" suffix (indicating "straight") in the binary tree for F. Figure 5 shows an example of F and its binary tree structure annotated with the _ST and _SW suffixes. The resulting binary tree syntactic structure of F is augmented with straight or swap/inversion suffixes, which can be regarded as a derivation of ITG between F and F'.

Thus, an ITG model can be learned from the binary tree structures using a probabilistic CFG learning algorithm. This learned model is the ITG parsing model. In this study, we use the state split probabilistic CFG [Petrov et al. 2006] for learning the ITG parsing model. The learned ITG parsing model is our preordering model.

5. PREORDERING SENTENCES

This section explains how to preorder input sentences and training sentences.

[8]Spearman's ρ will also work.

[9]The word order of nonaligned words does not affect Kendall's τ.

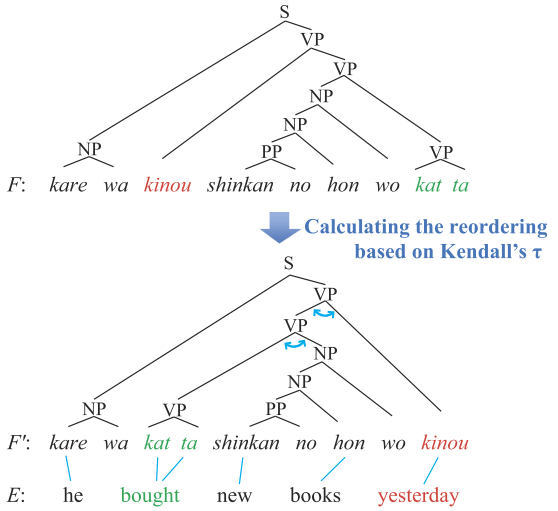

Fig. 4. Example of calculation of the reordering from F to F' based on Kendall's τ.

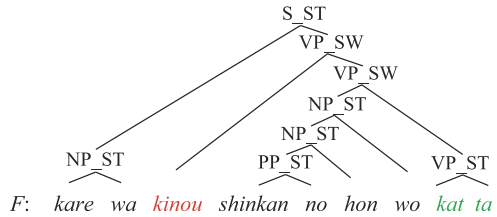

Fig. 5. Example of F and its binary tree structure annotated with _ST and _SW suffixes.

5.1. Preordering Input Sentences

Input sentences are preordered using the ITG parsing model described in Section 4.5. The preordering process is shown in Figure 6. An input sentence F is parsed using the ITG parsing model.[10] When F is parsed, the reordered structure for F' is jointly identified based on ITG. Each nonterminal node of a phrase label in the tree derivation is augmented by either an "_ST" suffix or an "_SW" suffix. The word order for F' is determined by the binary tree derivation with the suffixes of the nonterminal nodes. We swap the child nodes of the nodes augmented with the "_SW" suffix in the binary tree derivation to produce F'.

5.2. Preordering the Training Sentences

After transforming the F of an input sentence into F', we use a phrase-based SMT to translate F' into E. Therefore, a phrase-based SMT requires parallel F' and E sentences to train its translation model. Now, we will explain how to produce F' for the parallel sentences for training the SMT translation model.

If F' in the training data is produced using the same method as for preordering input sentences, then the word order of F' in the training data will be consistent with the word order of the preordered input sentences. However, the method for preordering input sentences is not always the best method for preordering the training data. This

[10]When there are unknown words, we estimate their POS tags using the function of the Berkeley parser [Petrov et al. 2006].

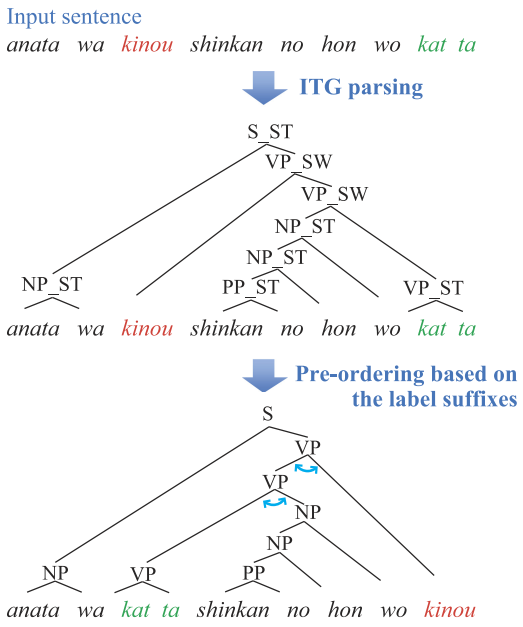

Fig. 6. Preordering an input sentence.

is because a corresponding E already exists in the training data. Thus, we also have to consider the consistency between F' and E in the training data. Methods that do not consider the consistency between F' and E will not be optimal.

It is important to consider the consistency between F' and E. The objective of preordering the training sentences is to build a phrase table. The phrase table is the SMT translation model, which consists of parallel phrase pairs between F' and E and their probabilities. When a pair of corresponding expressions in E and F' are both continuous, they can be extracted as a parallel phrase pair. A span in F that is projected by the method described in Section 4.1 indicates that the span in F and its corresponding span in E are both continuous. If the projected span in F is transformed into a noncontinuous expression in F' by preordering, then a parallel phrase pair for the noncontinuous expression in F' and the corresponding expression in E cannot be extracted as a phrase pair. Therefore, it is optimal that F be reordered into F' under the condition that this problem can be avoided, using (to the maximum possible extent) the same method used for preordering input sentences.

Thus, we preorder F in the training data into F' as follows. Partial syntactic structures are first projected onto the source-language sentences in the training data using the method described in Section 4.1. The source-language sentences are then parsed and reordered using the ITG parsing model (described in Section 5.1) within the constraints of the projected spans. The constraining method is the same as that used in Section 4.4.

6. EXPERIMENT

Our main goal is to translate text between two languages with widely different word orders, such as a SOV and SVO, with a high-quality target-language parser. Therefore, we conduct a Japanese-to-English (JE) translation to test the quality of translation from an SOV language to an SVO language. In addition, we conduct a Chinese-to-

English (CE) translation to test the quality of translation from an SVO language to another SVO language, as Chinese and English are more similar in terms of word order than Japanese and English. We investigate the efficacy of our method by comparing it with other methods. We used the patent data from the NTCIR-9 and NTCIR-10 Patent Machine Translation Tasks [Goto et al. 2011; Goto et al. 2013a] for the experiment.

6.1. Common Settings

The training data and the development data from the NTCIR-9 and NTCIR-10 are the same, but the test data are different. The JE training data consists of approximately 3.18 million sentence pairs and the CE training data consists of 1 million sentence pairs. The development data consists of 2,000 sentence pairs. There are 2,000 test sentences for the NTCIR-9 and 2,300 for the NTCIR-10. The reference data for each test sentence is a single reference translation.

We use Enju [Miyao and Tsujii 2008], which outputs head-binarized trees, to parse the English sentences in the training data. We applied a parsing customization for patent sentences [Isozaki et al. 2012]. MeCab[11] is used for Japanese segmentation, and the Stanford segmenter[12] is used for Chinese segmentation. We adjust the tokenization of alphanumeric characters in Japanese to be the same as for English.

The translation model is trained using sentences that are 40 words or less in length and English side sentences that could be parsed to produce binary syntactic tree structures. Approximately 2.06 million sentence pairs are used to train the translation model for JE. Approximately 0.40 million sentence pairs are used to train the translation model for CE. GIZA++ and grow-diag-final-and heuristics are used to obtain word alignments. To reduce word alignment errors, we remove the articles {a, an, the} in English and the particles {*ga*, *wo*, *wa*} in Japanese before performing word alignments, because these function words do not have corresponding words in the other languages between Japanese–English or Chinese–English. After word alignment, we restore the words that had been removed and shift the word alignment positions to the original word positions.

We use 5-gram language models with modified Kneser-Ney discounting [Chen and Goodman 1998] using the SRILM toolkit [Stolcke et al. 2011]. The language models are trained using the English sentences from the bilingual training data.

The SMT weighting parameters are tuned via MERT [Och 2003] using the development data. To stabilize the MERT results, we tune the parameters three times via MERT using the first half of the development data. We then select the SMT weighting parameter set that performs the best on the second half of the development data based on the BLEU scores from the three SMT weighting parameter sets.

6.2. Training and Settings for the Proposed Method

Next, we describe how the proposed method (PROPOSED) was performed. As the training data for our preordering model, we produce source-language full binary syntactic tree structures for 0.1 million source-language training sentences, which are selected using the method described in Section 4.2.[13] To produce the probabilistic CFG-model and the probabilistic model for unsupervised POS tagging, we use the Gibbs sampler for 100 iterations.[14] We use $|\mathcal{T}| = 50$, which employs the same number of word

[11] `http://mecab.sourceforge.net/`.

[12] `http://nlp.stanford.edu/software/segmenter.shtml`.

[13] We did not conduct experiments using larger training datasets because there would have been a very high computational cost in building probabilistic models for parsing.

[14] For JE, a single thread process ran for five days for 100 iterations on a Xeon processor E5-2680 2.70 GHz with 128GB memory.

classes used in the Moses default setting, where $|\mathcal{T}|$ is the number of POS tag types. The Berkeley parser [Petrov et al. 2006], which is an implementation of the state split probabilistic CFG-based parser, is used to train our preordering model and to parse using the preordering model. We perform six split-merge iterations as the same iteration of the parsing model for English [Petrov et al. 2006]. We use the phrase-based SMT system Moses [Koehn et al. 2007] to translate from F' into E with a distortion limit of 6, which limits the moves of phrases for word reordering to six or less words.

6.3. Training and Settings for the Comparison Methods

We used the following six comparison methods.

— Phrase-based SMT with lexicalized reordering models (PBMT$_L$) [Koehn et al. 2007].
— Hierarchical phrase-based SMT (HPBMT) [Chiang 2007].
— String-to-tree syntax-based SMT (SBMT) [Hoang et al. 2009].
— Phrase-based SMT with a distortion model (PBMT$_D$) [Goto et al. 2014].
— Preordering using a source-language dependency parser (SRCDEP) [Genzel 2010].[15]
— Preordering without using a parser (LADER) [Neubig et al. 2012].[16]

We use Moses [Hoang et al. 2009; Koehn et al. 2007] for PBMT$_L$, HPBMT, SBMT, SRCDEP, and LADER. We use an in-house standard phrase-based SMT decoder compatible with the Moses decoder with a distortion model [Goto et al. 2014] for PBMT$_D$.

For PBMT$_L$, we use the MSD bidirectional lexicalized reordering models [Koehn et al. 2005], which are built using all of the data used to build the translation model.

The distortion models for PBMT$_D$ are trained using 0.2 million source-language sentences[17] from the data used to build the translation model. This setting is the same as that in the experiments by Goto et al. [2014]. For PBMT$_D$, we use source-language POS tags produced by MeCab for Japanese and the Stanford tagger[18] for Chinese.

SRCDEP requires a source-language dependency parser. We use CaboCha[19] [Kudo and Matsumoto 2002] and POS tags produced by MeCab to obtain Japanese dependency structures[20] and use the Stanford parser[21] and POS tags produced by the Stanford tagger to obtain Stanford dependencies for Chinese [Chang et al. 2009]. Note that there are publicly available Japanese dependency parsers but there are no publicly available Japanese constituency parsers. The preordering rules of SRCDEP are built using all of the data used to build the translation model.

[15]There are three variations of the metrics for selecting rules. We implement variant 1 (optimizing crossing score), which achieved the best score for JE translation in the three variations in a study by Genzel [2010].
[16]We use the lader implementation available at `http://www.phontron.com/lader/`.
[17]The JE data is sorted in chronological order. The CE data is sorted at random. The last 0.2 million sentences of each data are used.
[18]`http://nlp.stanford.edu/software/tagger.shtml`.
[19]`https://code.google.com/p/cabocha/`.
[20]The CaboCha parser does not output word-based dependencies, but segment-based dependencies. Each segment, which is called a *bunsetsu*, comprises at least one content word, with or without its subsequent function words. We convert the segment-based dependencies to word-based dependencies as follows: when a punctuation mark is included in a segment, the segment is split into a segment without the punctuation mark and a segment consisting only of the punctuation mark. Each word, with the exception of the last word in a segment, depends on (modifies) the adjacent word to the right. The last word in a segment depends on the headword of the parent (modified) segment. The headword in a segment was the last content word in the segment.

The CaboCha parser does not output dependency relations. We add dependency relations to the word-based dependencies as follows: when the last word in a segment is a particle, we use the particle as the dependency relation between the word and its parent (modified) word because particles are case markers in many cases in Japanese. For other words, we use "none" as their dependency relations to their parent words.
[21]`http://nlp.stanford.edu/software/lex-parser.shtml`.

Table II. Japanese-English Evaluation Results

	Parser		Pre-	NTCIR-9		NTCIR-10	
	Source	Target	ordering	RIBES	BLEU	RIBES	BLEU
PBMT$_{L\text{-}4}$				65.48	26.73	65.53	27.44
PBMT$_{L\text{-}20}$				68.79	30.92	68.30	31.07
HPBMT				70.11	30.29	69.69	30.77
SBMT		✓		72.54	31.94	71.32	32.40
PBMT$_D$				73.54	33.14	72.23	33.87
SRCDEP	✓		✓	71.88	29.23	71.20	29.40
LADER			✓	74.31	32.98	73.98	33.90
PROPOSED		✓	✓	**76.35**	**33.83**	**75.81**	**34.90**

Table III. Chinese-English Evaluation Results

	Parser		Pre-	NTCIR-9		NTCIR-10	
	Source	Target	ordering	RIBES	BLEU	RIBES	BLEU
PBMT$_{L\text{-}4}$				75.02	29.22	74.24	30.65
PBMT$_{L\text{-}10}$				76.11	31.20	75.41	32.34
HPBMT				77.68	32.39	77.45	33.61
SBMT		✓		78.44	32.47	77.68	33.90
PBMT$_D$				77.98	33.03	77.48	34.28
SRCDEP	✓		✓	76.88	28.85	76.14	29.36
LADER			✓	78.18	30.80	77.06	31.12
PROPOSED		✓	✓	**81.61**	**35.16**	**81.05**	**36.22**

The preordering models for LADER are trained using the same 0.1 million source-language sentences and their word alignments as the training data for the preordering models of PROPOSED. We use source-language word classes produced by the Moses toolkit. Note that while the LADER preordering method does not use a parser, the training data for LADER is selected using a target-language parser. We perform 100 iterations to train the LADER preordering model.[22]

For PBMT$_L$, we use distortion limits of 4 or 20 for JE translation and distortion limits of 4 or 10 for CE translation, because a limit of 20 is the best for JE translation and a limit of 10 is the best for CE translation among 10, 20, 30, and ∞, as per studies by Goto et al. [2014] and because Genzel [2010] uses a baseline phrase-based SMT capable of local reordering of up to 4 words. To distinguish between the distortion limits for PBMT$_L$, we indicate the distortion limit as a subscript of PBMT$_L$, such as PBMT$_{L\text{-}20}$ for a distortion limit of 20. For PBMT$_D$, a distortion limit of 20 is used for JE translation and a distortion limit of 10 is used for CE translation. An unlimited max-chart-span is used for HPBMT and SBMT and a distortion limit of 6 is used for the preordering methods of SRCDEP and LADER. The default values are used for the other system parameters.

6.4. Results and Discussion

We evaluate the translation quality based on the case-insensitive automatic evaluation scores from the BLEU-4 [Papineni et al. 2002] and RIBES v1.01 [Isozaki et al. 2010]. RIBES is an automatic evaluation measure based on word-order correlation coefficients between reference sentences and translation outputs. Our main results for

[22]We also tested 200 iterations for JE translation and found that the results with 200 iterations did not improve when compared with the results for 100 iterations.

JE translation are presented in Table II and those for CE translation are presented in Table III. In these tables, check marks indicate usage for that method. Bold numbers indicate that values are not significantly lower than the best result (i.e., nonbold numbers indicate values that are significantly lower than the best result) in each test set and in each evaluation measure.[23] To assess this, we used the bootstrap resampling test at a significance level of $\alpha = 0.01$ [Koehn 2004].

PROPOSED achieved the best scores for both RIBES and BLEU in both the NTCIR-9 and NTCIR-10 datasets, and for both JE and CE translation. Because RIBES is sensitive to global word order and BLEU is sensitive to local word order, this confirms the efficacy of PROPOSED for both global and local word ordering.

Now, we compare the effects of the differences in the approaches. First, we compare our method with three existing methods that do not use a parser and conduct word selection and reordering jointly. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for the standard phrase-based SMT (PBMT$_{L\text{-}20}$), the hierarchical phrase-based SMT (HPBMT), and the phrase-based SMT with a recent distortion model (PBMT$_D$). These results confirm that preordering is effective compared with these methods, which do not use a parser and conduct word selection and reordering simultaneously, for JE patent translation. The tendencies of the CE translation results are the same as those of the JE translation results. These results confirm that preordering is also effective for CE patent translation.

Next, we compare our method with an existing method that uses a target-language syntactic parser, SBMT. The required resources are the same as those for PROPOSED. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for the string-to-tree syntax-based SMT (SBMT). These results confirm that preordering is effective compared with a method that uses target-language syntactic structures and conducts word selection and reordering simultaneously for JE patent translation. The tendencies of the CE translation results are also the same as those of the JE translation results.

We then compare our method with an existing method using a source-language dependency parser, SRCDEP. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for SRCDEP. These results confirm that our method is effective compared with a method that uses a source-language dependency parser [Genzel 2010] for JE patent translation. The tendencies of the CE translation results are the same as those of the JE translation results.

We confirm the effects of SRCDEP for JE (i.e., SOV to SVO) translation.[24] SRCDEP produces BLEU scores that are about two BLEU points higher than those for PBMT$_{L\text{-}4}$. These results are consistent with the experimental results of Genzel [2010]. Genzel [2010] compares their method with their baseline phrase-based SMT, which is capable of local reordering of up to four words. Although SRCDEP produces better BLEU scores than those for PBMT$_{L\text{-}4}$ and better RIBES scores than those for PBMT$_{L\text{-}4}$ and PBMT$_{L\text{-}20}$, the BLEU scores for SRCDEP are lower than those for PBMT$_{L\text{-}20}$. This indicates that even if a source-language dependency parser is used, it is not easy to

[23] We use this indication method because it can clarify the results of a hypothesis test using one result and many baseline results.

[24] Since Genzel [2010] reported the results of translations from English (an SVO language) to SOV or VSO languages, including Japanese, and did not report the results of a translation between English and Chinese (an SVO language to an SVO language), we discuss SRCDEP for JE translation.

Table IV. Effects of the Sentence Selection Method (JE Translation)

	NTCIR-9		NTCIR-10	
	RIBES	BLEU	RIBES	BLEU
LADER without our sentence selection	72.33	32.30	70.96	33.07
LADER with our sentence selection	74.31	32.98	73.98	33.90

improve JE translation quality by preordering.[25] One of the reasons that SRCDEP is unable to achieve scores on par with PROPOSED is thought to be because when SRCDEP changes the order of child nodes, the reordering rules consider only the local information. Reordering, however, should consider sentence-level consistency. For example, an SOV sentence in Japanese should be reordered into an SVO sentence for JE translation. However, when the subject in a sentence is omitted in Japanese, an OV sentence in Japanese should not be reordered into a VO sentence. This is because such sentences are usually translated into sentences in the passive voice, and the objects in Japanese become subjects in the translated sentences. Because SRCDEP preordering rules only consider local information, a rule is unable to handle the difference between SOV and OV when the rule does not consider S, such as when swapping O and V. In contrast, PROPOSED considers sentence-level consistency.

Finally, we compare our method with an existing preordering method that does not use a syntactic parser, LADER. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for LADER.[26] These results confirm that syntactic structures are effective for preordering in JE patent translation.[27] The tendencies of the CE translation results are also the same as those of the JE translation results.

Here, we confirm the effects of our method for selecting the training sentences described in Section 4.2 for JE translation. Because the sentence selection method is an indispensable element of our method, we compare LADER with our sentence selection method, which is LADER in Table II, to LADER without our sentence selection method. We used 0.1 million source-language sentences from the training data as the training data for the preordering model of LADER without our sentence selection method. The size of the 0.1 million sentences is the same as the size of the training data for the

[25]There are also systems with preordering methods that use a source-language dependency parser in the Japanese-to-English translation subtasks at NTCIR-10 and NTCIR-7.

At NTCIR-10, there was one system (name: JEPREORDER, ID: NTITI-je-2) with a source syntax-based preordering method that used manually-produced preordering rules and a Japanese dependency parser with a case structure analyzer [Sudoh et al. 2013]. Compared with the baseline hierarchical phrase-based SMT system (ID: BASELINE1-1) at NTCIR-10, the BLEU score for JEPREORDER is higher than that of the baseline system, but the RIBES score is not better than that of the baseline system in Table 1 in Sudoh et al. [2013].

At NTCIR-7, there was one system (ID: MIT (2)) with a source syntax-based preordering method that used manually produced preordering rules and a Japanese dependency parser [Katz-Brown and Collins 2008]. This system was unable to produce a BLEU score that was better than that of the baseline phrase-based SMT system at NTCIR-7.

[26]Note that although LADER works without a syntactic parser, the scores for LADER in Table II could not be achieved without a syntactic parser, because a syntactic parser is used in the selection process of the training data for the preordering model of LADER. The results for cases when our sentence selection method is not applied for LADER are shown later in this section. We use the same training data for the preordering model of LADER as for the preordering model of PROPOSED to ensure a fair comparison.

[27]There was also a system with a preordering method that does not require a parser in the Japanese-to-English translation subtask at NTCIR-9. The system of the NAIST group [Kondo et al. 2011] used a preordering method [Tromble and Eisner 2009] that learned a preordering model automatically without requiring a parser. This system was unable to produce a BLEU score that was better than those for the baseline systems of phrase-based SMT and hierarchical phrase-based SMT at NTCIR-9, although it could produce a RIBES score that was better than those for the baseline systems.

preordering model of LADER in Table II. The results are shown in Table IV. The RIBES and BLEU scores for LADER with our sentence selection method are higher than those for LADER without our sentence selection method. This comparison confirms the efficacy of our sentence selection method. This result also confirms that the learning of BTG from parallel sentences with highly synchronized parallel structures is effective compared with the learning from parallel sentences with less synchronized parallel structures.

Through the process described in Section 4.1, we assess the percentage of the spans in the source language that do not conflict in all of the spans that could be projected from the target language. As we explain in Section 4.1, to ensure that the projected structures can compose tree structures and consist solely of high-quality structures, we do not project any subtree spans in E when their corresponding spans in F conflict with one other. The nonconflict span rate in the source language is calculated by dividing the number of projected nonconflict spans in the source language by the number of spans in the source language that could be projected without consideration of conflict. The nonconflict span rates for the data selected to build the translation model are 0.782 for Japanese and 0.747 for Chinese.

We also check the average coverage rates of the projected spans, except for the sentence root spans,[28] in the process described in Section 4.1. Let r_2 be the coverage rate of the projected spans, except for the root span, for a source-language sentence. r_2 is calculated by dividing the number of projected spans, except for the sentence root span, by the number of words in a sentence minus two.[29,30] The average r_2 for the data selected to build the translation model (2.06 million Japanese sentences and 0.40 million Chinese sentences) is 0.560 for Japanese and 0.601 for Chinese. The average r_2 for the 0.1 million sentences selected via our sentence selection method (described in Section 4.2) is 0.856 for Japanese and 0.828 for Chinese.[31] With these projected partial structures, full binary tree structures were produced using the methods described in Sections 4.3, 4.4, and 5.2.[32]

In these experiments, we have not compared our method with postordering methods. However, for the same NTCIR-9 test data, the RIBES score (76.35) and the BLEU score (33.83) for PROPOSED are higher than the RIBES score (75.12) and the BLEU score (32.95) reported in Goto et al. [2013b], which were calculated in the same way as ours, for a postordering method [Goto et al. 2013b]. The postordering method of Goto et al. [2013b] used the same state split probabilistic CFG method for the ITG parsing model as our method for the ITG parsing model. In addition, PROPOSED has an advantage over the postordering methods of Sudoh et al. [2011b], Goto et al. [2013b], and Hayashi et al. [2013]. These postordering methods use manually-defined, high-quality preordering rules of head-finalization for translation from English to Japanese [Isozaki et al. 2012], so it is not easy to apply these methods to other language pairs. In contrast, PROPOSED does not require such manually-defined rules, and thus can be applied to other languages.

[28]As sentence root spans are obvious and do not need to be projected, we exclude them to investigate the percentage of spans that are projected.

[29]The number of spans in a full binary tree is the number of words in a sentence minus one. We subtract one from the number of spans to remove the sentence root span.

[30]r_2 can also be used instead of r_1 for sentence selection in Section 4.2.

[31]We also report average r_1. The average r_1 for the data selected to build the translation model is 0.574 for Japanese and 0.613 for Chinese. The average r_1 for the 0.1 million sentences is 0.864 for Japanese and 0.834 for Chinese.

[32]This does not mean that all of the minimal projected spans are included in the full binary trees. The reason for this is given in Section 4.4.

Table V. Evaluation Results for Parsing

	F_1 (CTB5-40)
[Jiang et al. 2011]	49.2*
Proposed method	**56.1**

Note: ∗ denotes "not our experiment."

6.5. Evaluation of Projection

To investigate the effects of our projection method, we compare the parsing quality produced by our method with that produced by the method of Jiang et al. [2011]. We use the same data and evaluation method as Jiang et al. [2011]. We use the same FBIS Chinese–English parallel corpus (LDC2003E14), which consists of 0.24 million sentence pairs, to obtain projected constituent structures. We evaluate our projected parser using the same test data as the subset of Chinese Treebank 5.0 (CTB 5.0; LDC2005T01), which consists of no more than 40 words after the removal of punctuation, just as in Jiang et al. [2011].

We use the same evaluation metric of unlabeled F_1 as Jiang et al. [2011], which is the harmonic mean of the unlabeled precision and recall. This is defined by Klein [2005, pp. 19–22]. The evaluation for unlabeled brackets differs slightly from the standard PARSEVAL metrics: multiplicity of brackets is ignored, brackets with a span of one are ignored, and bracket labels are ignored. Previous research [Jiang et al. 2011; Klein 2005, p. 16] removed punctuation before conducting the evaluations. Followed this, we remove words that have PU punctuation tags in CTB 5.0 after parsing.

We use our method, described in Sections 4.1 to 4.4, 6.1, and 6.2 to obtain projected constituent structures. As a result of the process described in Section 4.1, the nonconflict span rate in the source language for the training data of the FBIS corpus is 0.681. To reduce computational costs, we change one of the settings described in Section 6.2. To produce a probabilistic CFG model and full binary trees, we select the top 50,000 unique source-language sentences from the FBIS corpus, whereas we selected the top 0.1 million unique source-language sentences in Section 6.2.[33] The average coverage rate (r_2) of the projected spans, except for the sentence root spans, for the 50,000 sentences selected to produce full binary tree structures is 0.795. We use the Berkeley parser, which was also used by Jiang et al. [2011] for the same purpose, to build the parsing model from the projected constituent structures and to parse the test data.

Jiang et al. [2011] uses the gold POS tags from CTB 5.0 for parsing and a supervised Chinese POS tagger for tagging the FBIS corpus. In contrast, we do not use the gold POS tags from CTB 5.0 or a supervised Chinese POS tagger. Therefore, a comparison of our method with that of Jiang et al. [2011] would be unfair.

The evaluation results are given in Table V. Although our method does not require source-language POS tags, our method produced an F_1 higher than that of Jiang et al. [2011]. This confirms the efficacy of our projection method.

The quality of projected spans is affected by the quality of word alignments between languages as well as the quality of target-language parse trees. To check the quality of spans projected by the method described in Section 4.1, we use the English translation with annotations[34] (LDC2007T02) of a part of CTB 5.0 and the corresponding Chinese sentences in CTB 5.0. From the data, we extract parallel sentence pairs that have: (1) a one-to-one sentence correspondence, (2) sentence lengths of 40 words or less, and

[33]The average r_2 of the top 0.1 million sentences in the FBIS corpus is 0.668, which is lower than that of the NTCIR-9/10 data (0.856 for Japanese and 0.828 for Chinese). A lower rate increases the number of parse tree candidates and also the computational costs.

[34]We use the annotations to count the number of English sentences corresponding to each Chinese sentence.

(3) English side sentences that could be parsed to produce binary syntactic tree structures. The extracted Chinese–English parallel sentence pairs are called parallel CTB in this article. The parallel CTB comprise 2,477 sentence pairs. We obtain word alignments using the parallel CTB and the FBIS corpus simultaneously. Then, we project spans using our method described in Section 4.1. We check the quality of the minimal projected spans for the parallel CTB. Error spans, which are also called cross brackets, are the projected spans that conflict with subtree spans of CTB 5.0. The conflict is that a projected span and a subtree span in the syntactic tree of CTB 5.0 for a sentence partially overlap each other. The error rate of the minimal projected spans is 0.217, which is calculated by dividing the number of error spans by the number of projected spans, except for the root spans.[35] For the Chinese sentences from the parallel CTB, the nonconflict span rate is 0.689, and the average coverage rate (r_2) of the projected spans, except for the root spans, is 0.608.

7. CONCLUSION

We have presented a preordering method that uses a target-language parser to process syntactic structures without a source-language parser. This is achieved by projecting source-language syntactic structures from the corresponding target-language constituency structures, as well as learning an ITG-based parsing model for the source-language using the projected syntactic structures. Our method, which is based on cross-language syntactic projection and sentence selection, facilitates the learning of ITG by producing highly-synchronized parallel syntactic structures. In the experiments on Japanese-to-English and Chinese-to-English patent translation, our method is significantly better in terms of translation quality as measured by both RIBES and BLEU when compared with phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, an existing preordering method that does not use a parser, and an existing preordering method that uses a source-language dependency parser. As RIBES is sensitive to global word order and BLEU is sensitive to local word order, we conclude that our proposed method is better than the compared methods in terms of global and local word ordering. We also confirm the efficacy of our projection method compared with an existing projection method for constituent structures using the FBIS corpus and Chinese Treebank 5.0. Future work will involve cooperating with a source-language parser when one is available.

ACKNOWLEDGMENTS

I. Goto would like to thank Chenhui Chu for providing sentence alignments between LDC2007T02 and CTB5.0. We would like to thank the three anonymous reviewers for their comments which substantially improved the article.

REFERENCES

A. V. Aho and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci. 3*, 1, 37–56. DOI: http://dx.doi.org/10.1016/S0022-0000(69)80006-1.

Ibrahim Badr, Rabih Zbib, and James Glass. 2009. Syntactic phrase reordering for English-to-Arabic statistical machine translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL'09)*. Association for Computational Linguistics, 86–93. http://www.aclweb.org/anthology/E09-1011.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the 3rd Workshop on Syntax and*

[35]The minimal projected spans, with the exception of the error spans, do not always match the correct spans because there may be ambiguities in the projected spans and the tree structures of CTB5.0 are not binary trees.

Structure in Statistical Translation (SSST-3) at NAACL HLT. Association for Computational Linguistics, 51–59. http://www.aclweb.org/anthology/W09-2307.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98. Computer Science Group, Harvard University.

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist. 33*, 2, 201–228.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, 531–540. DOI: http://dx.doi.org/10.3115/1219840.1219906.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 193–203. http://www.aclweb.org/anthology/D11-1018.

Chris Dyer and Philip Resnik. 2010. Context-free reordering, finite-state translation. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 858–866. http://www.aclweb.org/anthology/N10-1128.

Niyu Ge. 2010. A direct syntax-driven reordering model for phrase-based machine translation. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 849–857. http://www.aclweb.org/anthology/N10-1127.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, 376–384. http://www.aclweb.org/anthology/C10-1043.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 Workshop. In *Proceedings of NTCIR-9*. 559–578.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013a. Overview of the patent machine translation task at the NTCIR-10 Workshop. In *Proceedings of NTCIR-10*. 260–286.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2013b. Post-ordering by parsing with ITG for Japanese-English statistical machine translation. *ACM Trans. Asian Lang. Inf. Process. 12*, 4, Article 17, 22 pages. DOI: http://dx.doi.org/10.1145/2518100.

Isao Goto, Masao Utiyama, Eiichiro Sumita, Akihiro Tamura, and Sadao Kurohashi. 2014. Distortion model based on word sequence labeling for statistical machine translation. *ACM Trans. Asian Lang. Inf. Process. 13*, 1, Article 2, 21 pages. DOI: http://dx.doi.org/10.1145/2537128.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of the Machine Translation Summit XI*. 215–222.

Katsuhiko Hayashi, Katsuhito Sudoh, Hajime Tsukada, Jun Suzuki, and Masaaki Nagata. 2013. Shift-reduce word reordering for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1382–1386. http://www.aclweb.org/anthology/D13-1139.

Hieu Hoang, Philipp Koehn, and Adam Lopez. 2009. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*. 152–159.

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Two-stage preordering for Japanese-to-English statistical machine translation. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 1062–1066. http://www.aclweb.org/anthology/I13-1147.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng. 11*, 3, 311–325. DOI: http://dx.doi.org/10.1017/S1351324905003840.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 944–952.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2012. HPSG-based preprocessing for English-to-Japanese translation. *ACM Trans. Asian Lang. Inf. Process. 11*, 3, Article 8, 16 pages. DOI: http://dx.doi.org/10.1145/2334801.2334802.

Wenbin Jiang, Qun Liu, and Yajuan Lv. 2011. Relaxed cross-lingual projection of constituent syntax. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1192–1201. http://www.aclweb.org/anthology/D11-1110.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov Chain Monte Carlo. In *Human Language Technologies: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 139–146. http://www.aclweb.org/anthology/N/N07/N07-1018.

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

Mitesh M. Khapra, Ananthakrishnan Ramanathan, and Karthik Visweswariah. 2013. Improving reordering performance using higher order and structural features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 315–324. http://www.aclweb.org/anthology/N13-1032.

Dan Klein. 2005. The unsupervised learning of natural language structure. Ph.D. Dissertation, Stanford University.

William R. Knight. 1966. A computer method for calculating Kendall's tau with ungrouped data. *J. Amer. Statist. Assoc. 61*, 314. DOI: http://dx.doi.org/10.2307/2282833.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, Dekang Lin and Dekai Wu Eds. Association for Computational Linguistics, 388–395.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, 177–180. http://www.aclweb.org/anthology/P07-2045.

Shuhei Kondo, Mamoru Komachi, Yuji Matsumoto, Katsuhito Sudoh, Kevin Duh, and Hajime Tsukada. 2011. Learning of linear ordering problems and its application to J-E patent translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR-9*. 641–645.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL'02) (COLING 2002 Post-Conference Workshops)*. 63–69.

K. Lari and S. J. Young. 1991. Applications of stochastic context-free grammars using the Inside-Outside algorithm. *Comput. Speech Lang. 5*, 3, 237–257. DOI: http://dx.doi.org/10.1016/0885-2308(91)90009-F.

Uri Lerner and Slav Petrov. 2013. Source-side classifier preordering for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 513–523. http://www.aclweb.org/anthology/D13-1049.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, 720–727. http://www.aclweb.org/anthology/P07-1091.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Comput. Linguist. 34*, 1, 81–88.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 843–853. http://www.aclweb.org/anthology/D12-1077.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 160–167. DOI: http://dx.doi.org/10.3115/1075096.1075117.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 311–318.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 433–440. DOI: http://dx.doi.org/10.3115/1220175.1220230.

Jim Pitman and Marc Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Ann. Probab. 25*, 2, 855–900.

Ananthakrishnan Ramanathan, Hegde, Jayprasad, Ritesh M. Shah, Pushpak Bhattacharyya, and Sasikumar M. 2008. Simple syntactic and morphological processing can help English-Hindi statistical machine translation. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*. 171–180.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata, Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2011a. NTT-UT statistical machine translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR-9*. 585–592.

Katsuhito Sudoh, Jun Suzuki, Hajime Tsukada, and Masaaki Nagata. 2013. NTT-NII statistical machine translation for NTCIR-10 PatentMT. In *Proceedings of NTCIR-10*. 294–300.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011b. Post-ordering in statistical machine translation. In *Proceedings of the 13th Machine Translation Summit*. 316–323.

Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06. NUS School of Computing.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the HLT-NAACL'04* (Short Papers), Daniel Marcu, Susan Dumais, and Salim Roukos (Eds.), Association for Computational Linguistics, USA, 101–104.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1007–1016. http://www.aclweb.org/anthology/D/D09/D09-1105.

Karthik Visweswariah, Jiri Navratil, Jeffrey Sorensen, Vijil Chenthamarakshan, and Nandakishore Kambhatla. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*. Coling 2010 Organizing Committee, 1119–1127. http://www.aclweb.org/anthology/C10-1126.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A word reordering model for improved machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 486–496. http://www.aclweb.org/anthology/D11-1045.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, 737–745. http://www.aclweb.org/anthology/D/D07/D07-1077.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist. 23*, 3, 377–403.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011a. Extracting preordering rules from chunk-based dependency trees for Japanese-to-English translation. In *Proceedings of the 13th Machine Translation Summit*. 300–307.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011b. Extracting preordering rules from predicate-argument structures. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 29–37. http://www.aclweb.org/anthology/I11-1004.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING'04*. 508–514.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 245–253. http://www.aclweb.org/anthology/N/N09/N09-1028.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 523–530. DOI: http://dx.doi.org/10.3115/1073012.1073079.

Received March 2014; revised September 2014; accepted November 2014