Distortion Model Considering Rich Context for Statistical Machine Translation

Isao Goto[†,‡] **Masao Utiyama**[†] **Eiichiro Sumita**[†]
Akihiro Tamura[†] **Sadao Kurohashi**[‡]
[†]National Institute of Information and Communications Technology
[‡]Kyoto University
goto.i-es@nhk.or.jp
{mutiyama, eiichiro.sumita, akihiro.tamura}@nict.go.jp
kuro@i.kyoto-u.ac.jp

Abstract

This paper proposes new distortion models for phrase-based SMT. In decoding, a distortion model estimates the source word position to be translated next (NP) given the last translated source word position (CP). We propose a distortion model that can consider the word at the CP, a word at an NP candidate, and the context of the CP and the NP candidate simultaneously. Moreover, we propose a further improved model that considers richer context by discriminating label sequences that specify spans from the CP to NP candidates. It enables our model to learn the effect of relative word order among NP candidates as well as to learn the effect of distances from the training data. In our experiments, our model improved 2.9 BLEU points for Japanese-English and 2.6 BLEU points for Chinese-English translation compared to the lexical reordering models.

1 Introduction

Estimating appropriate word order in a target language is one of the most difficult problems for statistical machine translation (SMT). This is particularly true when translating between languages with widely different word orders.

To address this problem, there has been a lot of research done into word reordering: lexical reordering model (Tillman, 2004), which is one of the distortion models, reordering constraint (Zens et al., 2004), pre-ordering (Xia and McCord, 2004), hierarchical phrase-based SMT (Chiang, 2007), and syntax-based SMT (Yamada and Knight, 2001).

In general, source language syntax is useful for handling long distance word reordering. However, obtaining syntax requires a syntactic parser, which is not available for many languages. Phrase-based SMT (Koehn et al., 2007) is a widely used SMT method that does not use a parser.

Phrase-based SMT mainly[1] estimates word reordering using distortion models[2]. Therefore, distortion models are one of the most important components for phrase-based SMT. On the other hand, there are methods other than distortion models for improving word reordering for phrase-based SMT, such as pre-ordering or reordering constraints. However, these methods also use distortion models when translating by phrase-based SMT. Therefore, distortion models do not compete against these methods and are commonly used with them. If there is a good distortion model, it will improve the translation quality of phrase-based SMT and benefit to the methods using distortion models.

In this paper, we propose two distortion models for phrase-based SMT. In decoding, a distortion model estimates the source word position to be translated next (NP) given the last translated source word position (CP). The proposed models are *the pair model* and *the sequence model*. The pair model utilizes the word at the CP, a word at an NP candidate site, and the words surrounding the CP and the NP candidates (context) simultaneously. In addition, the sequence model, which is the further improved model, considers richer context by identifying the label sequence that specify the span from the CP to the NP. It enables our model to learn the effect of relative word order among NP candidates as well as to learn the effect of distances from the training data. Our model learns the preference relations among NP

[1] A language model also supports the estimation.

[2] In this paper, reordering models for phrase-based SMT, which are intended to estimate the source word position to be translated next in decoding, are called distortion models. This estimation is used to produce a hypothesis in the target language word order sequentially from left to right.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 155–165,
Sofia, Bulgaria, August 4-9 2013.

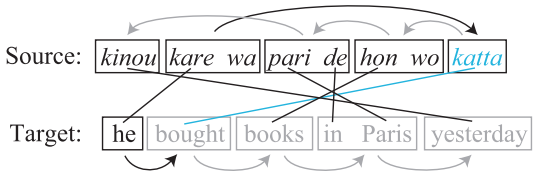

Figure 1: An example of left-to-right translation for Japanese-English. Boxes represent phrases and arrows indicate the translation order of the phrases.

candidates. Our model consists of one probabilistic model and does not require a parser. Experiments confirmed the effectiveness of our method for Japanese-English and Chinese-English translation, using NTCIR-9 Patent Machine Translation Task data sets (Goto et al., 2011).

2 Distortion Model for Phrase-Based SMT

A Moses-style phrase-based SMT generates target hypotheses sequentially from left to right. Therefore, the role of the distortion model is to estimate the source phrase position to be translated next whose target side phrase will be located immediately to the right of the already generated hypotheses. An example is shown in Figure 1. In Figure 1, we assume that only the *kare wa* (English side: "he") has been translated. The target word to be generated next will be "bought" and the source word to be selected next will be its corresponding Japanese word *katta*. Thus, a distortion model should estimate phrases including *katta* as a source phrase position to be translated next.

To explain the distortion model task in more detail, we need to redefine more precisely two terms, the *current position* (CP) and *next position* (NP) in the source sentence. CP is the source sentence position corresponding to the rightmost aligned target word in the generated target word sequence. NP is the source sentence position corresponding to the leftmost aligned target word in the target phrase to be generated next. The task of the distortion model is to estimate the NP[3] from NP candidates (NPCs) for each CP in the source sentence.[4]

[3]NP is not always one position, because there may be multiple correct hypotheses.

[4]This definition is slightly different from that of existing methods such as Moses and (Green et al., 2010). In existing methods, CP is the rightmost position of the last translated source phrase and NP is the leftmost position of the source phrase to be translated next. Note that existing methods do not consider word-level correspondences.

Figure 2: Examples of CP and NP for Japanese-English translation. The upper sentence is the source sentence and the sentence underneath is a target hypothesis for each example. The NP is in bold, and the CP is in bold italics. The point of an arrow with a × mark indicates a wrong NP candidate.

Estimating NP is a difficult task. Figure 2 shows some examples. The superscript numbers indicate the word position in the source sentence.

In Figure 2 (a), the NP is 8. However, in Figure 2 (b), the word (*kare*) at the CP is the same as (a), but the NP is different (the NP is 10). From these examples, we see that distance is not the essential factor in deciding an NP. And it also turns out that the word at the CP alone is not enough to estimate the NP. Thus, not only the word at the CP but also the word at a NP candidate (NPC) should be considered simultaneously.

In (c) and (d) in Figure 2, the word (*kare*) at the CP is the same and *karita* (borrowed) and *katta* (bought) are at the NPCs. *Karita* is the word at the NP and *katta* is not the word at the NP for (c), while *katta* is the word at the NP and *karita* is not the word at the NP for (d). From these examples, considering what the word is at the NP

is not enough to estimate the NP. One of the reasons for this difference is the relative word order between words. Thus, considering relative word order is important.

In (d) and (e) in Figure 2, the word (*kare*) at the CP and the word order between *katta* and *karita* are the same. However, the word at the NP for (d) and the word at the NP for (e) are different. From these examples, we can see that selecting a nearby word is not always correct. The difference is caused by the words surrounding the NPCs (context), the CP context, and the words between the CP and the NPC. Thus, these should be considered when estimating the NP.

In summary, in order to estimate the NP, the following should be considered simultaneously: the word at the NP, the word at the CP, the relative word order among the NPCs, the words surrounding NP and CP (context), and the words between the CP and the NPC.

There are distortion models that do not require a parser for phrase-based SMT. The linear distortion cost model used in Moses (Koehn et al., 2007), whose costs are linearly proportional to the reordering distance, always gives a high cost to long distance reordering, even if the reordering is correct. The MSD lexical reordering model (Tillman, 2004; Koehn et al., 2005; Galley and Manning, 2008) only calculates probabilities for the three kinds of phrase reorderings (monotone, swap, and discontinuous), and does not consider relative word order or words between the CP and the NPC. Thus, these models are not sufficient for long distance word reordering.

Al-Onaizan and Papineni (2006) proposed a distortion model that used the word at the CP and the word at an NPC. However, their model did not use context, relative word order, or words between the CP and the NPC.

Ni et al. (2009) proposed a method that adjusts the linear distortion cost using the word at the CP and its context. Their model does not simultaneously consider both the word specified at the CP and the word specified at the NPCs.

Green et al. (2010) proposed distortion models that used context. Their model (the outbound model) estimates how far the NP should be from the CP using the word at the CP and its context.[5] Their model does not simultaneously consider both the word specified at the CP and the word specified at an NPC. For example, the outbound model considers the word specified at the CP, but does not consider the word specified at an NPC. Their models also do not consider relative word order.

In contrast, our distortion model solves the aforementioned problems. Our distortion models utilize the word specified at the CP, the word specified at an NPC, and also the context of the CP and the NPC simultaneously. Furthermore, our sequence model considers richer context including the relative word order among NPCs and also including all the words between the CP and the NPC. In addition, unlike previous methods, our models learn the preference relations among NPCs.

3 Proposed Method

In this section, we first define our distortion model and explain our learning strategy. Then, we describe two proposed models: *the pair model* and *the sequence model* that is the further improved model.

3.1 Distortion Model and Learning Strategy

First, we define our distortion model. Let i be a CP, j be an NPC, S be a source sentence, and X be the random variable of the NP. In this paper, *distortion probability* is defined as $P(X = j|i, S)$, which is the probability of an NPC j being the NP. Our distortion model is defined as the model calculating the distortion probability.

Next, we explain the learning strategy for our distortion model. We train this model as a discriminative model that discriminates the NP from NPCs. Let J be a set of word positions in S other than i. We train the distortion model subject to

$$\sum_{j \in J} P(X = j|i, S) = 1.$$

The model parameters are learned to maximize the distortion probability of the NP among all of the NPCs J in each source sentence. This learning strategy is a kind of preference relation learning (Evgniou and Pontil, 2002). In this learning, the

[5] They also proposed another model (the inbound model) that estimates reverse direction distance. Each NPC is regarded as an NP, and the inbound model estimates how far the corresponding CP should be from the NP using the word at the NP and its context.

distortion probability of the actual NP will be relatively higher than those of all the other NPCs J.

This learning strategy is different from that of (Al-Onaizan and Papineni, 2006; Green et al., 2010). For example, Green et al. (2010) trained their outbound model subject to $\sum_{c \in C} P(Y = c|i, S) = 1$, where C is the set of the nine distortion classes[6] and Y is the random variable of the correct distortion class that the correct distortion is classified into. Distortion is defined as $j - i - 1$. Namely, the model probabilities that they learned were the probabilities of distortion classes in all of the training data, not the relative preferences among the NPCs in each source sentence.

3.2 Pair Model

The pair model utilizes the word at the CP, the word at an NPC, and the context of the CP and the NPC simultaneously to estimate the NP. This can be done by our distortion model definition and the learning strategy described in the previous section.

In this work, we use the maximum entropy method (Berger et al., 1996) as a discriminative machine learning method. The reason for this is that a model based on the maximum entropy method can calculate probabilities. However, if we use scores as an approximation of the distortion probabilities, various discriminative machine learning methods can be applied to build the distortion model.

Let s be a source word and $s_1^n = s_1 s_2 ... s_n$ be a source sentence. We add a beginning of sentence (BOS) marker to the head of the source sentence and an end of sentence (EOS) marker to the end, so the source sentence S is expressed as s_0^{n+1} ($s_0 = \text{BOS}, s_{n+1} = \text{EOS}$). Our distortion model calculates the distortion probability for an NPC $j \in \{j | 1 \leq j \leq n+1 \ \wedge \ j \neq i\}$ for each CP $i \in \{i | 0 \leq i \leq n\}$

$$P(X = j|i, S) = \frac{1}{Z_i} \exp\left(\mathbf{w}^{\mathrm{T}} \mathbf{f}\left(i, j, S, o, d\right)\right) \tag{1}$$

where

$$o = \begin{cases} 0 & (i < j) \\ 1 & (i > j) \end{cases}, \quad d = \begin{cases} 0 & (|j - i| = 1) \\ 1 & (2 \leq |j - i| \leq 5) \\ 2 & (6 \leq |j - i|) \end{cases},$$

[6] $(-\infty, -8]$, $[-7, -5]$, $[-4, -3]$, -2, 0, 1, $[2, 3]$, $[4, 6]$, and $[7, \infty)$. In (Green et al., 2010), -1 was used as one of distortion classes. However, -1 represents the CP in our definition, and CP is not an NPC. Thus, we shifted all of the distortion classes for negative distortions by -1.

Template
$\langle o \rangle$, $\langle o, s_p \rangle$[1], $\langle o, t_i \rangle$, $\langle o, t_j \rangle$, $\langle o, d \rangle$, $\langle o, s_p, s_q \rangle$[2], $\langle o, t_i, t_j \rangle$, $\langle o, t_{i-1}, t_i, t_j \rangle$, $\langle o, t_i, t_{i+1}, t_j \rangle$, $\langle o, t_i, t_{j-1}, t_j \rangle$, $\langle o, t_i, t_j, t_{j+1} \rangle$, $\langle o, s_i, t_i, t_j \rangle$, $\langle o, s_j, t_i, t_j \rangle$

[1] $p \in \{p | i - 2 \leq p \leq i + 2 \ \vee \ j - 2 \leq p \leq j + 2\}$
[2] $(p, q) \in \{(p, q) | i - 2 \leq p \leq i + 2 \ \wedge \ j - 2 \leq q \leq j + 2 \ \wedge \ (|p - i| \leq 1 \ \vee \ |q - j| \leq 1)\}$

Table 1: Feature templates. t is the part of speech of s.

$\mathbf{w}$ is a weight parameter vector, each element of $\mathbf{f}(\cdot)$ is a binary feature function, and $Z_i = \sum_{j \in \{j | 1 \leq j \leq n+1 \ \wedge \ j \neq i\}} (\text{numerator of Equation 1})$ is a normalization factor. o is an orientation of i to j and d is a distance class.

The binary feature function that constitutes an element of $\mathbf{f}(\cdot)$ returns 1 when its feature is matched and if else, returns 0. Table 1 shows the feature templates used to produce the features. A feature is an instance of a feature template.

In Equation 1, i, j, and S are used by the feature functions. Thus, Equation 1 can utilize features consisting of both s_i, which is the word specified at i, and s_j, which is the word specified at j, or both the context of i and the context of j simultaneously. Distance is considered using the distance class d. Distortion is represented by distance and orientation. The pair model considers distortion using six joint classes of d and o.

3.3 Sequence Model

The pair model does not consider relative word order among NPCs or all the words between the CP and an NPC. In this section, we propose a further improved model, *the sequence model*, which considers richer context including relative word order among NPCs and also including all the words between the CP and an NPC.

In (c) and (d) in Figure 2, *karita* (borrowed) and *katta* (bought) occur in the source sentences. The pair model considers the effect of distances using only the distance class d. If these positions are in the same distance class, the pair model cannot consider the differences in distances. In this case, these are conflict instances during training and it is difficult to distinguish the NP for translation.

Now to explain how to consider the relative word order by the sequence model. The sequence model considers the relative word order by discriminating the label sequence corresponding to the NP from the label sequences corresponding to

Label	Description
C	A position is the CP.
I	A position is a position between the CP and the NPC.
N	A position is the NPC.

Table 2: The "C, I, and N" label set.

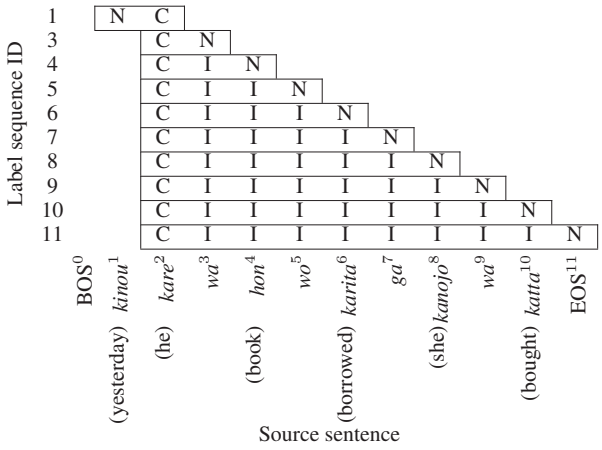

Figure 3: Example of label sequences that specify spans from the CP to each NPC for the case of Figure 2 (c). The labels (C, I, and N) in the boxes are the label sequences.

each NPC in each sentence. Each label sequence corresponds to one NPC. Therefore, if we identify the label sequence that corresponds to the NP, we can obtain the NP. The label sequences specify the spans from the CP to each NPC using three kinds of labels indicating the type of word positions in the spans. The three kinds of labels, "C, I, and N," are shown in Table 2. Figure 3 shows examples of the label sequences for the case of Figure 2 (c). In Figure 3, the label sequences are represented by boxes and the elements of the sequences are labels. The NPC is used as the label sequence ID for each label sequence.

The label sequence can treat relative word order. For example, the label sequence ID of 10 in Figure 3 knows that *karita* exists to the left of the NPC of 10. This is because *karita*^{6} carries a label I while *katta*^{10} carries a label N, and a position with label I is defined as relatively closer to the CP than a position with label N. By utilizing the label sequence and corresponding words, the model can reflect the effect of *karita* existing between the CP and the NPC of 10 on the probability.

For the sequence model, *karita* (borrowed) and *katta* (bought) in (c) and (d) in Figure 2 are not conflict instances in training, whereas they are conflict instances in training for the pair model. The reason is as follows. In order to make the probability of the NPC of 10 smaller than the NPC of 6, instead of making the weight parameters for the features with respect to the word at the position of 10 with label N smaller than the weight parameters for the features with respect to the word at the position of 6 with label N, the sequence model can give negative weight parameters for the features with respect to the word at the position of 6 with label I.

We use a sequence discrimination technique based on CRF (Lafferty et al., 2001) to identify the label sequence that corresponds to the NP. There are two differences between our task and the CRF task. One difference is that CRF discriminates label sequences that consist of labels from all of the label candidates, whereas we constrain the label sequences to sequences where the label at the CP is C, the label at an NPC is N, and the labels between the CP and the NPC are I. The other difference is that CRF is designed for discriminating label sequences corresponding to the same object sequence, whereas we do not assign labels to words outside the spans from the CP to each NPC. However, when we assume that another label such as E has been assigned to the words outside the spans and there are no features involving label E, CRF with our label constraints can be applied to our task. In this paper, the method designed to discriminate label sequences corresponding to the different word sequence lengths is called *partial CRF*.

The sequence model based on partial CRF is derived by extending the pair model. We introduce the label l and extend the pair model to discriminating the label sequences. There are two extensions to the pair model. One extension uses labels. We suppose that label sequences specify the spans from the CP to each NPC. We conjoined all the feature templates in Table 1 with an additional feature template $\langle l_i, l_j \rangle$ to include the labels into features where l_i is the label corresponding to the position of i. The other extension uses sequence. In the pair model, the position pair of (i, j) is used to derive features. In contrast, to descriminate label sequences in the sequence model, the position pairs of (i, k), $k \in \{k | i < k \leq j \ \vee \ j \leq k < i\}$

and (k, j), $k \in \{k | i \leq k < j \ \vee \ j < k \leq i\}$ are used to derive features. Note that in the feature templates in Table 1, i and j are used to specify two positions. When features are used for the sequence model, one of the positions is regarded as k.

The distortion probability for an NPC j being the NP given a CP i and a source sentence S is calculated as:

$$P(X = j|i, S) = \frac{1}{Z_i} \exp\left(\sum_{k \in M \cup \{j\}} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(i, k, S, o, d, l_i, l_k\right) + \sum_{k \in M \cup \{i\}} \mathbf{w}^{\mathrm{T}} \mathbf{f}\left(k, j, S, o, d, l_k, l_j\right) \right) \quad (2)$$

where

$$M = \begin{cases} \{m | i < m < j\} & (i < j) \\ \{m | j < m < i\} & (i > j) \end{cases}$$

and $Z_i = \sum_{j \in \{j | 1 \leq j \leq n+1 \ \wedge \ j \neq i\}}$(numerator of Equation 2) is a normalization factor. Since j is used as the label sequence ID, discriminating j also means discriminating label sequence IDs.

The first term in $\exp(\cdot)$ in Equation 2 considers all of the word pairs located at i and other positions in the sequence, and also their context. The second term in $\exp(\cdot)$ in Equation 2 considers all of the word pairs located at j and other positions in the sequence, and also their context.

By designing our model to discriminate among different length label sequences, our model can naturally handle the effect of distances. Many features are derived from a long label sequence because it will contain many labels between the CP and the NPC. On the other hand, fewer features are derived from a short label sequence because a short label sequence will contain fewer labels between the CP and the NPC. The bias from these differences provides important clues for learning the effect of distances.[7]

[7]Note that the sequence model does not only consider larger context than the pair model, but that it also considers labels. The pair model does not discriminate labels, whereas the sequence model uses label N and label I for the positions except for the CP, depending on each situation. For example, in Figure 3, at position 6, label N is used in the label sequence ID of 6, but label I is used in the label sequence IDs of 7 to 11. Namely, even if they are at the same position, the labels in the label sequences are different. The sequence model discriminates the label differences.

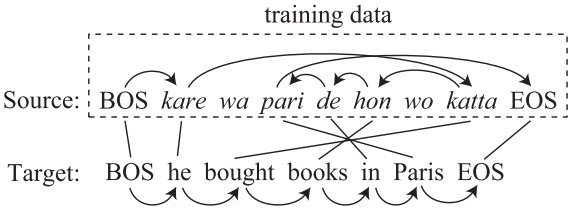

Figure 4: Examples of supervised training data. The lines represent word alignments. The English side arrows point to the nearest word aligned on the right.

3.4 Training Data for Discriminative Distortion Model

To train our discriminative distortion model, supervised training data is needed. The training data is built from a parallel corpus and word alignments between corresponding source words and target words. Figure 4 shows examples of training data. We select the target words aligned to the source words sequentially from left to right (target side arrows). Then, the order of the source words in the target word order is decided (source side arrows). The source sentence and the source side arrows are the training data.

4 Experiment

In order to confirm the effects of our distortion model, we conducted a series of Japanese to English (JE) and Chinese to English (CE) translation experiments.[8]

4.1 Common Settings

We used the patent data for the Japanese to English and Chinese to English translation subtasks from the NTCIR-9 Patent Machine Translation Task (Goto et al., 2011). There were 2,000 sentences for the test data and 2,000 sentences for the development data.

Mecab[9] was used for the Japanese morphological analysis. The Stanford segmenter[10] and tagger[11] were used for Chinese segmentation and POS tagging. The translation model was trained using sentences of 40 words or less from the training data. So approximately 2.05 million sentence pairs consisting of approximately 54 million

[8]We conducted JE and CE translation as examples of language pairs with different word orders and of languages where there is a great need for translation into English.

[9]http://mecab.sourceforge.net/

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

[11]http://nlp.stanford.edu/software/tagger.shtml

Japanese tokens whose lexicon size was 134k and 50 million English tokens whose lexicon size was 213k were used for JE. And approximately 0.49 million sentence pairs consisting of 14.9 million Chinese tokens whose lexicon size was 169k and 16.3 million English tokens whose lexicon size was 240k were used for CE. GIZA++ and grow-diag-final-and heuristics were used to obtain word alignments. In order to reduce word alignment errors, we removed articles {a, an, the} in English and particles {*ga*, *wo*, *wa*} in Japanese before performing word alignments because these function words do not correspond to any words in the other languages. After word alignment, we restored the removed words and shifted the word alignment positions to the original word positions. We used 5-gram language models that were trained using the English side of each set of bilingual training data.

We used an in-house standard phrase-based SMT system compatible with the Moses decoder (Koehn et al., 2007). The SMT weighting parameters were tuned by MERT (Och, 2003) using the development data. To stabilize the MERT results, we tuned three times by MERT using the first half of the development data and we selected the SMT weighting parameter set that performed the best on the second half of the development data based on the BLEU scores from the three SMT weighting parameter sets.

We compared systems that used a common SMT feature set from standard SMT features and different distortion model features. The common SMT feature set consists of: four translation model features, phrase penalty, word penalty, and a language model feature. The compared different distortion model features are: the linear distortion cost model feature (LINEAR), the linear distortion cost model feature and the six MSD bidirectional lexical distortion model (Koehn et al., 2005) features (LINEAR+LEX), the outbound and inbound distortion model features discriminating nine distortion classes (Green et al., 2010) (9-CLASS), the proposed pair model feature (PAIR), and the proposed sequence model feature (SEQUENCE).

4.2 Training for the Proposed Models

Our distortion model was trained as follows: We used 0.2 million sentence pairs and their word alignments from the data used to build the translation model as the training data for our distortion models. The features that were selected and used were the ones that had been counted[12], using the feature templates in Table 1, at least four times for all of the (i, j) position pairs in the training sentences. We conjoined the features with three types of label pairs $\langle \mathrm{C}, \mathrm{I} \rangle$, $\langle \mathrm{I}, \mathrm{N} \rangle$, or $\langle \mathrm{C}, \mathrm{N} \rangle$ as instances of the feature template $\langle l_i, l_j \rangle$ to produce features for SEQUENCE. The L-BFGS method (Liu and Nocedal, 1989) was used to estimate the weight parameters of maximum entropy models. The Gaussian prior (Chen and Rosenfeld, 1999) was used for smoothing.

4.3 Training for the Compared Models

For 9-CLASS, we used the same training data as for our distortion models. Let t_i be the part of speech of s_i. We used the following feature templates to produce features for the outbound model: $\langle s_{i-2} \rangle$, $\langle s_{i-1} \rangle$, $\langle s_i \rangle$, $\langle s_{i+1} \rangle$, $\langle s_{i+2} \rangle$, $\langle t_i \rangle$, $\langle t_{i-1}, t_i \rangle$, $\langle t_i, t_{i+1} \rangle$, and $\langle s_i, t_i \rangle$. These feature templates correspond to the components of the feature templates of our distortion models. In addition to these features, we used a feature consisting of the relative source sentence position as the feature used by (Green et al., 2010). The relative source sentence position is discretized into five bins, one for each quintile of the sentence. For the inbound model[13], i of the feature templates was changed to j. Features occurring four or more times in the training sentences were used. The maximum entropy method with Gaussian prior smoothing was used to estimate the model parameters.

The MSD bidirectional lexical distortion model was built using all of the data used to build the translation model.

4.4 Results and Discussion

We evaluated translation quality based on the case-insensitive automatic evaluation score BLEU-4 (Papineni et al., 2002). We used distortion limits of 10, 20, 30, and unlimited (∞), which limited the number of words for word reordering to a maximum number. Table 3 presents our main results. The proposed SEQUENCE outperformed the baselines for both Japanese to English and Chinese to English translation. This demonstrates the effectiveness of the proposed SEQUENCE. The scores of the proposed SEQUENCE were higher than those

[12] When we counted features for selection, we only counted features that were from the feature templates of $\langle s_i, s_j \rangle$, $\langle t_i, t_j \rangle$, $\langle s_i, t_i, t_j \rangle$, and $\langle s_j, t_i, t_j \rangle$ in Table 1 when j was not the NP, in order to avoid increasing the number of features.

[13] The inbound model is explained in footnote 5.

	Japanese-English				Chinese-English			
Distortion limit	10	20	30	∞	10	20	30	∞
LINEAR	27.98	27.74	27.75	27.30	29.18	28.74	28.31	28.33
LINEAR+LEX	30.25	30.37	30.17	29.98	30.81	30.24	30.16	30.13
9-CLASS	30.74	30.98	30.92	30.75	31.80	31.56	31.31	30.84
PAIR	31.62	32.36	31.96	32.03	32.51	32.30	32.25	32.32
SEQUENCE	32.02	32.96	**33.29**	32.81	**33.41**	**33.44**	**33.35**	**33.41**

Table 3: Evaluation results for each method. The values are case-insensitive BLEU scores. Bold numbers indicate no significant difference from the best result in each language pair using the bootstrap resampling test at a significance level $\alpha = 0.01$ (Koehn, 2004).

	Japanese-English	Chinese-English
HIER	30.47	32.66

Table 4: Evaluation results for hierarchical phrase-based SMT.

of the proposed PAIR. This confirms the effectiveness for considering relative word order and words between the CP and an NPC. The proposed PAIR outperformed 9-CLASS, confirming that considering both the word specified at the CP and the word specified at the NPC simultaneously was more effective than that of 9-CLASS.

For translating between languages with widely different word orders such as Japanese and English, a small distortion limit is undesirable because there are cases where correct translations cannot be produced with a small distortion limit, since the distortion limit prunes the search space that does not meet the constraint. Therefore, a large distortion limit is required to translate correctly. For JE translation, our SEQUENCE achieved significantly better results at distortion limits of 20 and 30 than that at a distortion limit of 10, while the baseline systems of LINEAR, LINEAR+LEX, and 9-CLASS did not achieve this. This indicate that SEQUENCE could treat long distance reordering candidates more appropriately than the compared methods.

We also tested hierarchical phrase-based SMT (Chiang, 2007) (HIER) using the Moses implementation. The common data was used to train HIER. We used unlimited max-chart-span for the system setting. Results are given in Table 4. Our SEQUENCE outperformed HIER. The gain for JE was large but the gain for CE was modest. Since phrase-based SMT is generally faster in decoding speed than hierarchical phrase-based SMT, achieving better or comparable scores is worthwhile.

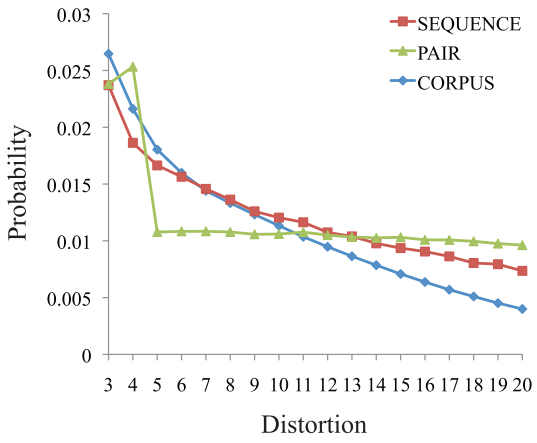

Figure 5: Average probabilities for large distortion for Japanese-English translation.

To investigate the tolerance for sparsity of the training data, we reduced the training data for the sequence model to 20,000 sentences for JE translation.[14] SEQUENCE using this model with a distortion limit of 30 achieved a BLEU score of 32.22.[15] Although the score is lower than the score of SEQUENCE with a distortion limit of 30 in Table 3, the score was still higher than those of LINEAR, LINEAR+LEX, and 9-CLASS for JE in Table 3. This indicates that the sequence model also works even when the training data is not large. This is because the sequence model considers not only the word at the CP and the word at an NPC but also rich context, and rich context would be effective even for a smaller set of training data.

[14] We did not conduct experiments using larger training data because there would have been a very high computational cost to build models using the L-BFGS method.

[15] To avoid effects from differences in the SMT weighting parameters, we used the same SMT weighting parameters for SEQUENCE, with a distortion limit of 30, in Table 3.

To investigate how well SEQUENCE learns the effect of distance, we checked the average distortion probabilities for large distortions of $j - i - 1$. Figure 5 shows three kinds of probabilities for distortions from 3 to 20 for Japanese-English translation. One is the average distortion probabilities in the Japanese test sentences for each distortion for SEQUENCE, and another is this for PAIR. The third (CORPUS) is the probabilities for the actual distortions in the training data that were obtained from the word alignments used to build the translation model. The probability for a distortion for CORPUS was calculated by the number of the distortion divided by the total number of distortions in the training data.

Figure 5 shows that when a distance class feature used in the model was the same (e.g., distortions from 5 to 20 were the same distance class feature), PAIR produced average distortion probabilities that were almost the same. In contrast, the average distortion probabilities for SEQUENCE decreased when the lengths of the distortions increased, even if the distance class feature was the same, and this behavior was the same as that of CORPUS. This confirms that the proposed SEQUENCE could learn the effect of distances appropriately from the training data.[16]

5 Related Works

We discuss related works other than discussed in Section 2. Xiong et al. (2012) proposed a model predicting the orientation of an argument with respect to its verb using a parser. Syntactic structures and predicate-argument structures are useful for reordering. However, orientations do not handle distances. Thus, our distortion model does not compete against the methods predicting orientations using a parser and would assist them if used together.

There are word reordering constraint methods using ITG (Wu, 1997) for phrase-based SMT (Zens et al., 2004; Yamamoto et al., 2008; Feng et al., 2010). These methods consider sentence level consistency with respect to ITG. The ITG constraint does not consider distances of reordering and was used with other distortion models. Our distortion model does not consider sentence level consistency, so our distortion model and ITG constraint methods are thought to be complementary.

There are tree-based SMT methods (Chiang, 2007; Galley et al., 2004; Liu et al., 2006). In many cases, tree-based SMT methods do not use the distortion models that consider reordering distance apart from translation rules because it is not trivial to use distortion scores considering the distances for decoders that do not generate hypotheses from left to right. If it could be applied to these methods, our distortion model might contribute to tree-based SMT methods. Investigating the effects will be for future work.

6 Conclusion

This paper described our distortion models for phrase-based SMT. Our sequence model simply consists of only one probabilistic model, but it can consider rich context. Experiments indicate that our models achieved better performance and the sequence model could learn the effect of distances appropriately. Since our models do not require a parser, they can be applied to many languages. Future work includes application to other language pairs, incorporation into ITG constraint methods and other reordering methods, and application to tree-based SMT methods.

[16] We also checked the average distortion probabilities for the 9-CLASS outbound model in the Japanese test sentences for Japanese-English translation. We averaged the average probabilities for distortions in a distortion span of [4, 6] and also averaged those in a distortion span of [7, 20], where the distortions in each span are in the same distortion class. The average probability for [4, 6] was 0.058 and that for [7, 20] was 0.165. From CORPUS, the average probabilities in the training data for each distortion in [4, 6] were higher than those for each distortion in [7, 20]. However, the converse was true for the comparison between the two average probabilities for the outbound model. This is because the sum of probabilities for distortions from 7 and above was larger than the sum of probabilities for distortions from 4 to 6 in the training data. This comparison indicates that the 9-CLASS outbound model could not appropriately learn the effects of large distances for JE translation.

References

Yaser Al-Onaizan and Kishore Papineni. 2006. Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536, Sydney, Australia, July. Association for Computational Linguistics.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, March.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. Technical report.

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

Theodoros Evgniou and Massimiliano Pontil. 2002. Learning preference relations from data. *Neural Nets Lecture Notes in Computer Science*, 2486:23–32.

Yang Feng, Haitao Mi, Yang Liu, and Qun Liu. 2010. An efficient shift-reduce decoding algorithm for phrased-based machine translation. In *Coling 2010: Posters*, pages 285–293, Beijing, China, August. Coling 2010 Organizing Committee.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

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*, pages 559–578.

Spence Green, Michel Galley, and Christopher D. Manning. 2010. Improved models of distortion cost for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 867–875, Los Angeles, California, June. Association for Computational Linguistics.

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*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

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

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of 18th International Conference on Machine Learning*, pages 282–289.

D.C. Liu and J. Nocedal. 1989. On the limited memory method for large scale optimization. *Mathematical Programming B*, 45(3):503–528.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 609–616, Sydney, Australia, July. Association for Computational Linguistics.

Yizhao Ni, Craig Saunders, Sandor Szedmak, and Mahesan Niranjan. 2009. Handling phrase reorderings for machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 241–244, Suntec, Singapore, August. Association for Computational Linguistics.

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*, pages 160–167, Sapporo, Japan, July. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of Coling 2004*, pages 508–514, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Deyi Xiong, Min Zhang, and Haizhou Li. 2012. Modeling the translation of predicate-argument structure for smt. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 902–911, Jeju Island, Korea, July. Association for Computational Linguistics.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 523–530, Toulouse, France, July. Association for Computational Linguistics.

Hirofumi Yamamoto, Hideo Okuma, and Eiichiro Sumita. 2008. Imposing constraints from the source tree on ITG constraints for SMT. In *Proceedings of the ACL-08: HLT Second Workshop on Syntax and Structure in Statistical Translation (SSST-2)*, pages 1–9, Columbus, Ohio, June. Association for Computational Linguistics.

Richard Zens, Hermann Ney, Taro Watanabe, and Eiichiro Sumita. 2004. Reordering constraints for phrase-based statistical machine translation. In *Proceedings of Coling 2004*, pages 205–211, Geneva, Switzerland, Aug 23–Aug 27. COLING.