Neural Reordering Model Considering Phrase Translation and Word Alignment for Phrase-based Translation

Shin Kanouchi [†], Katsuhito Sudoh [‡], and Mamoru Komachi [†]

[†] Tokyo Metropolitan University
{kanouchi-shin at ed., komachi at}tmu.ac.jp
[‡] NTT Communication Science Laboratories, NTT Corporation
sudoh.katsuhito at lab.ntt.co.jp

Abstract

This paper presents an improved lexicalized reordering model for phrase-based statistical machine translation using a deep neural network. Lexicalized reordering suffers from reordering ambiguity, data sparseness and noises in a phrase table. Previous neural reordering model is successful to solve the first and second problems but fails to address the third one. Therefore, we propose new features using phrase translation and word alignment to construct phrase vectors to handle inherently noisy phrase translation pairs. The experimental results show that our proposed method improves the accuracy of phrase reordering. We confirm that the proposed method works well with phrase pairs including NULL alignments.

1 Introduction

Phrase-based statistical machine translation (PBSMT) (Koehn et al., 2003) has been widely used in the last decade. One major problem with PBSMT is word reordering. Since PBSMT models the translation process using a phrase table, it is not easy to incorporate global information during translation. There are many methods to address this problem, such as lexicalized reordering (Tillmann, 2004; Koehn et al., 2007; Galley and Manning, 2008), distance-based reordering (Koehn et al., 2003), pre-ordering (Wu et al., 2011; Hoshino et al., 2015; Nakagawa, 2015), and post-ordering (Sudoh et al., 2011). However, word reordering still faces serious errors, especially when the word order greatly differs in two languages, such as the case between English and Japanese.

In this paper, we focus on the lexicalized reordering model (LRM), which directly constrains reordering of phrases in PBSMT. LRM addresses the problem of a simple distance-based reordering approach in distant language pairs. However, there are some disadvantages: (1) reordering ambiguity, (2) data sparsity and (3) noisy phrases pairs. Li et al. (2014) addressed the problem of reordering ambiguity and data sparsity using a neural reordering model (NRM) that assigns reordering probabilities on the words of both the current and the previous phrase pairs. Also, Cui et al. (2016) tackled the problem of reordering ambiguity by including much longer context information on the source side than other LRMs including NRMs to determine phrase orientations using Long Short-Term Memory (LSTM).

However, there are a large number of noisy phrase pairs in the phrase table. One of the deficiencies of their NRMs is that they generated a phrase vector by simply embedding word vectors of the source and target language phrases and did not consider the adequacy of the translation between the phrase pair and the alignment of words in the phrases. It may be problematic especially when a phrase contains the NULL alignment, such as "," in "日本 で ||| in Japan ,". In addition, it is difficult to integrate the model of Cui et al. (2016) into stack decoding because their model is now conditioned not only on the words of each phrase pair but also on the history of decoded phrases. Furthermore, because they did not compare their model with the original NRM of Li et al. (2014), it is possible that their model is inferior to the previous approach.

Proceedings of the 3rd Workshop on Asian Translation,
pages 94–103, Osaka, Japan, December 11-17 2016.

Therefore, we propose to use phrase translation probability and word alignment features for NRM to address the problem of noisy phrase pairs. Both intrinsic and extrinsic experiments show that our features indeed improve the original NRM. The main contributions of this paper are as follows:

- We propose a new NRM incorporating phrase translation probabilities and word alignment in a phrase pair as features to handle inherently noisy phrase pairs more correctly.
- The experimental results show that our proposed method improves the accuracy of phrase reordering. In particular, the proposed method works well with phrase pairs including NULL alignments.
- We evaluate the proposed method on Japanese-to-English and English-to-Japanese translation using automatic and human evaluation.

2 Lexicalized Reordering Models

Lexicalized reordering models (LRM) maintain a reordering probability distribution for each phrase pair. Given a sequence of source phrases $\boldsymbol{f} = \overline{f}_{a_1}, \ldots, \overline{f}_{a_i}, \ldots, \overline{f}_{a_I}$, we translate and reorder the phrases to generate a sequence of target phrases $\boldsymbol{e} = \overline{e}_1, \ldots, \overline{e}_i, \ldots, \overline{e}_I$. Here $\boldsymbol{a} = a_1, \ldots, a_I$ expresses the alignment between the source phrase $\overline{f}_{a_i}$ and the target phrase $\overline{e}_i$. The alignment $\boldsymbol{a}$ can be used to represent the phrase orientation o. Three orientations with respect to previous phrase (Monotone, Swap, Discontinuous) are typically used in lexicalized reordering models (Galley and Manning, 2008). However, because global phrase reordering appears frequently in Japanese-to-English translation, Nagata et al. (2006) proposed four orientations instead of three by dividing the Discontinous label. In Figure 1, we show four orientations, called Monotone (Mono), Swap, Discontinuous-right ($\mathrm{D}_{\mathrm{right}}$) and Discontinuous-left ($\mathrm{D}_{\mathrm{left}}$). Using alignments a_i and a_{i-1}, orientation o_i respected to the target phrases $\overline{e}_i$, $\overline{e}_{i-1}$ follows:

$$o_i = \begin{cases} \text{Mono} & (a_i - a_{i-1} = 1) \\ \text{Swap} & (a_i - a_{i-1} = -1) \\ \mathrm{D}_{\text{right}} & (a_i - a_{i-1} > 1) \\ \mathrm{D}_{\text{left}} & (a_i - a_{i-1} < -1) \end{cases} \tag{1}$$

If the reordering probability of every phrase is expressed as $P(o_i|\overline{f}_{a_i}, \overline{e}_i)$, that of the sentence can be approximated as

$$P(a_1^I|\boldsymbol{e}) = \prod_{i=1}^{I} P(o_i|\overline{f}_{a_i}, \overline{e}_i). \tag{2}$$

LRM is a simple method to calculate the reordering probability for each phrase pair statistically.

However, the traditional lexicalized reordering model has the following disadvantages:

- **High ambiguity**: The phrase reordering orientation cannot be determined using only a phrase pair $\overline{f}_{a_i}, \overline{e}_i$, because the phrase reordering orientation may not be unique in a limited context.
- **Data sparsity**: The reordering probability is calculated for each phrase pair $\overline{f}_{a_i}, \overline{e}_i$. The phrase pair, which appears only once in training, amounts to 95%[1] of the entire phrase table. The reordering probability of these phrase pairs cannot be easily established.
- **Noisy phrases**: The reordering model does not consider the adequacy of the translation and the word alignments in phrases. For example, almost identical phrase pairs such as "日本 で ||| in Japan" and "日本 で ||| in Japan ," are often found in a phrase table. The difference between them is whether the phrase include "," which corresponds to the NULL alignment. Phrase tables in a distant language pairs like Japanese and English often contain the NULL alignment and mis-aligned words. On the contrary, there are also many phrase pairs that have crossing alignments such as "日本 で ||| in Japan" and "日本 で ||| Japan is." These locally reversed alignments deteriorate reordering accuracy.

[1] We experimented in the Kyoto Free Translation Task.

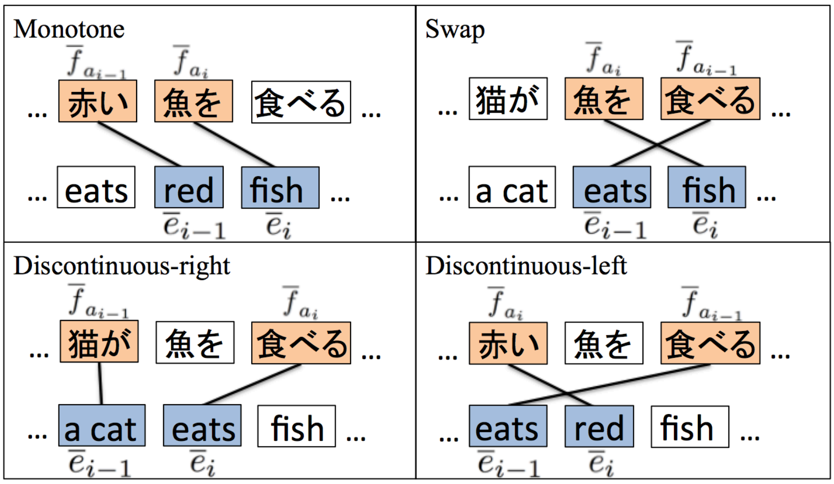

Figure 1: Four orientations, namely Monotone, Swap, Discontinuous-right and Discontinuous-left, are shown. Monotone means that the source phrases $\overline{f}_{a_i}$, $\overline{f}_{a_{i-1}}$ are adjoining and monotonic with respect to the target phrases $\overline{e}_i$, $\overline{e}_{i-1}$. Swap means $\overline{f}_{a_i}$, $\overline{f}_{a_{i-1}}$ are adjoining and swapping. Discontinuous-right means $\overline{f}_{a_i}$, $\overline{f}_{a_{i-1}}$ are separated and monotonic, and Discontinuous-left means $\overline{f}_{a_i}$, $\overline{f}_{a_{i-1}}$ are separated and swapping.

Li et al. (2013) proposed an NRM, which uses a deep neural network to address the problems of high ambiguity and data sparsity. We describe the NRM in the next section and propose our model to improve the NRM to address the problem of noisy phrases in Section 4.

3 Neural Reordering Model

Li et al. (2013) tackled the ambiguity and sparseness problem by distributed representation of phrases. The distributed representation maps *sparse* phrases into a *dense* vector space where elements with similar roles are expected to be located close to each other.

3.1 Distributed Representation of Phrases

Socher et al. (2011) proposed the recursive autoencoder, which recursively compresses a word vector and generates a phrase vector with the same dimension as the word vector. We define a word vector of u dimension $x \in \mathcal{R}^u$, an encoding weight matrix $W_e \in \mathcal{R}^{u \times 2u}$, and a bias term b_e. A phrase vector $p_{1:2}$ is constructed as follows:

$$p_{1:2} = f(W_e[x_1; x_2] + b_e) \tag{3}$$

f is an activation function such as $tanh$, which is used in our experiments.

When a phrase consists of more than two words, we compute a phrase vector $p_{1:n}$ recursively from the phrase vector $p_{1:n-1}$ and the word vector x_n.

$$p_{1:n} = f(W_e[p_{1:n-1}; x_n] + b_e) \tag{4}$$

We learn parameters to minimize the mean squared error between an input vector and the reconstructed vector using the autoencoder.

3.2 Deep Neural Network for Reordering

The NRM consists of an input layer built upon recursive autoencoders and outputs orientation score using a softmax layer (Li et al., 2013). An input is a concatenation of the current and previous phrase vectors in each language $p(\overline{f}_{a_i}), p(\overline{e}_i), p(\overline{f}_{a_{i-1}}), p(\overline{e}_{i-1})$ and an output is the reordering score $P(o_i)$ from the

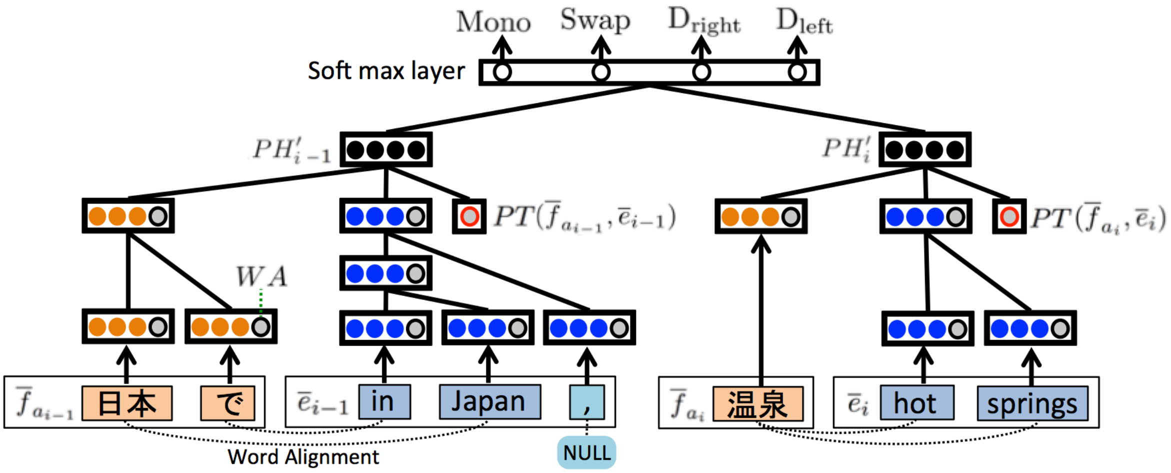
Mono
Swap
$\mathrm{D}_{\mathrm{right}}$
$\mathrm{D}_{\mathrm{left}}$
Soft max layer
PH'_{i-1}
PH'_i
$PT(\overline{f}_{a_{i-1}}, \overline{e}_{i-1})$
$PT(\overline{f}_{a_i}, \overline{e}_i)$
WA
$\overline{f}_{a_{i-1}}$ 日本 で
$\overline{e}_{i-1}$ in Japan ,
$\overline{f}_{a_i}$ 温泉
$\overline{e}_i$ hot springs
Word Alignment
NULL

Figure 2: An NRM considering phrase translation and word alignment. PT represents phrase translation shown in Section 4.1 and WA (gray cells) represent the word alignment shown in Section 4.2.

softmax layer. All the phrases in the same language use the same recursive autoencoder.

$$P(o_i) = \frac{\exp g(o_i)}{\Sigma_{o' \in \{M,S,D_r,D_l\}} \exp g(o')} \tag{5}$$

$$g(o_i) = f(W_r[PH_i; PH_{i-1}] + b_r) \tag{6}$$

$$PH_i = [p(\overline{f}_{a_i}); p(\overline{e}_i)] \tag{7}$$

Here, $o \in \{\mathrm{Mono}, \mathrm{Swap}, \mathrm{D}_{\mathrm{right}}, \mathrm{D}_{\mathrm{left}}\}$ expresses the classes of orientation described in Section 2. $W_r \in \mathcal{R}^{1\times 4n}$ is a weight matrix; PH_i is a phrase pair vector, which concatenates the phrase vectors $p(\overline{f}_{a_i})$ and $p(\overline{e}_i)$; and b_r is a bias term.

We calculate the error of the NRM $E_{nrm}(\theta)$ in each phrase pair using cross entropy.

$$E_{nrm}(\theta) = -\sum_{o} d_i(o) \log P(o) \tag{8}$$

where d_i is a correct reordering represented with a four-dimensional probability distribution. Each dimension corresponds to Mono, Swap, $\mathrm{D}_{\mathrm{right}}$, and $\mathrm{D}_{\mathrm{left}}$.

Finally, we compute the total of error $J(\theta)$, which is the sum of four errors of the recursive autoencoder $E_{rae}(\theta)$ and the error of the NRM $E_{nrm}(\theta)$. α is a hyper parameter controlling the trade-off between the models, and λ is an L_2 regularization coefficient.

$$J(\theta) = \alpha E_{rae}(\theta) + (1-\alpha)E_{nrm}(\theta) + \lambda||\theta||^2 \tag{9}$$

4 NRM with Phrase Translation and Word Alignment

The reordering of the phrase pair depends on each $\overline{f}_{a_i}$ and $\overline{e}_i$. However, the NRM generates a phrase vector merely by embedding a word vector, so that it does not take into account the adequacy of the translation between the phrase pair nor the word alignments. Therefore, in this paper, we embed the phrase translation probability and word alignments as features when we constitute a phrase pair. An overview of the model is illustrated in Figure 2.

4.1 Phrase Translation

We represent the translation probabilities between the phrase pair $\overline{f}_{a_i}$ and $\overline{e}_i$ in a four-dimensional vector $PT(\overline{f}_{a_i}, \overline{e}_i)$ to consider the adequacy of the translation between the phrase pair.

$$PT(\overline{f}_{a_i}, \overline{e}_i) = (P(\overline{e}_i|\overline{f}_{a_i}), P(\overline{f}_{a_i}|\overline{e}_i), lex(\overline{e}_i|\overline{f}_{a_i}), lex(\overline{f}_{a_i}|\overline{e}_i)) \tag{10}$$

Dimension	Description	
1	Word translation probability $P(e\|f)$	
2	Word translation probability $P(f\|e)$	
3	Whether the word aligns to a word where its position is	to the left of the phrase
4		to the center of the phrase
5		to the right of the phrase
6	Whether the word aligns to the NULL word	

Table 1: Word alignment information WA.

$P(\overline{e}_i|\overline{f}_{a_i})$ and $P(\overline{f}_{a_i}|\overline{e}_i)$ represent the translation probability of the phrase of both directions; $lex(\overline{e}_i|\overline{f}_{a_i})$ and $lex(\overline{f}_{a_i}|\overline{e}_i)$ compute the average of the translation probability of the words in the phrase of both directions.

We then concatenate the phrase pair vector PH_i and the phrase translation vector $PT(\overline{f}_{a_i}, \overline{e}_i)$ to obtain a new phrase pair vector PH'_i by using a weight matrix W_t. Again, b_t is a bias term.

$$PH'_i = f(W_t[PH_i; PT(\overline{f}_{a_i}, \overline{e}_i)] + b_t) \tag{11}$$

4.2 Word Alignment

We define a new word vector x', which incorporates word alignment information "WA" comprising six dimensions to the word vector x to propagate alignment information to the phrase vector.

$$x' = [x; WA] \in \mathcal{R}^{u+6} \tag{12}$$

Word translation probabilities are represented in the first two dimensions, and the location of the word alignment is represented in the following three dimensions. In addition, since some words are not aligned, i.e., "NULL Alignment," we create a special dimension corresponding to the NULL word.

Table 1 explains each dimension of WA. For example, the fourth dimension of WA of the word "日本 (Japan)" in Figure 2 is 1 because the aligned word "Japan" is located at the center of the phrase.

5 Experiment

We conduct two kinds of experiments: intrinsic evaluation of reordering accuracy and extrinsic evaluation of MT quality.

5.1 Setting

We use the Kyoto Free Translation Task[2] (KFTT) for our experiment. It is a task for Japanese-to-English translation that focuses on Wikipedia articles. We use KyTea[3] (ver.0.4.7) for Japanese word segmentation and GIZA++ (Och and Ney, 2003) with grow-diag-final-and for word alignment. We extract 70M phrase bigram pairs and automatically annotate the correct reordering orientation using Moses (Koehn et al., 2007). We filter out phrases that appear only once. We randomly divide the parallel corpus into training, development, and test. We retain 10K instances for development and test and use 1M instances for training.

We experimented 15, 25, 50, and 100-dimensional word vectors; 25-dimensional word vectors are used in all experiments involving our model. Thus, we set the vector size of the recursive auto-encoder to 31, to include the 25-dimensional word embeddings and the 6-dimensional WA. In a preliminary experiment, we compare the performance of randomly initialized word vectors with that of word vectors trained by the word2vec model[4]. Based on the result, we use word vectors trained by the word2vec model because of the performance. The word2vec model is pre-trained on English and Japanese versions of Wikipedia.

[2] http://www.phontron.com/kftt/

[3] http://www.phontron.com/kytea/

[4] https://code.google.com/archive/p/word2vec/

		Mono	Swap	$\mathrm{D_{right}}$	$\mathrm{D_{left}}$	Acc.
The ratio of labels [%]		30.39	16.06	31.86	21.69	
Baseline	Lexicalized Reordering Model (LRM)	71.54	36.92	**95.76**	39.33	66.71
	Neural Reordering Model (NRM)	77.06	57.60	70.31	60.63	68.22
Proposed	Phrase Translation (NRM+PT)	76.70	59.78	71.34	60.07	68.53
	Word Alignment (NRM+WA)	76.90	59.84	71.03	**62.38**	69.04
	NRM+PT+WA	**77.53**	**60.83**	72.69	61.78	**69.89**

Table 2: Recall and accuracy of reordering phrases.

Data size	time /epoch	Vocab size		Unknown words		Unknown phrases		Acc.
		ja	en	ja	en	ja	en	
10K	2 min	4,906	4,820	44%	45%	61%	63%	63.50
50K	9 min	10,833	10,880	25%	36%	48%	51%	66.88
200K	35 min	18,239	18,375	13%	22%	36%	39%	68.45
1M	170 min	26,978	27,152	7.3%	13%	24%	28%	**69.89**

Table 3: Data size and the accuracy of reordering. Vocab size reflects the vocabulary in the training data. The numbers of UNK words and UNK phrases are calculated in the test data. A pre-trained word2vec vector was given as the initial value for UNK words. Vocab sizes of test data are en:3,583 and ja:3,470. Phrase sizes of test data are en:8,187 and ja:7,945.

The pre-trained word2vec vector is also used to represent unknown words in the test data. If an unknown word is not included in the word2vec vocabulary, an unknown word vector is used to represent the word. In order to learn the unknown word vector, we randomly choose 1% of the words which appeared only once in the training data. Table 3 shows the size of the vocabulary.

We tune the hyperparameters with the development data up to a maximum of 30 epochs. We use the Adam optimizer with learning rate 0.0001. Our mini-batch size is 25. We drew the hyper parameter α uniformly from 0.05 to 0.3 with the development data and used α = 0.12 in our experiments. We also tried dropout but it did not show the improvements in our experiments.

We implemented the basic NRM and our proposed model using Chainer (Tokui et al., 2015) as a deep learning toolkit. When running on a single CPU (Intel Xeon E5-2697 v3@2.6GHz), it took five days to completely train a model.

5.2 Reordering Evaluation

Table 2 shows the accuracy of reordering. The performance of LRM is calculated from the reordering table created during training[5]. The recall of Mono and $\mathrm{D_{right}}$ are high because LRM uses only a phrase pair $\overline{f}_{a_i}, \overline{e}_i$ and tends to output major labels. On the other hand, NRMs succeed at estimating minor labels because the score is calculated from the phrase bigram pairs. As a result, only the NRM recall of $\mathrm{D_{right}}$ is inferior to LRM, and thus, the overall accuracy improves.

Furthermore, in NRMs, the use of phrase translation and the word alignment improves the accuracy by 0.31 and 0.82 points, respectively. Considering both these features, the accuracy of NRM is improved by 1.67 points.

5.2.1 Data Size

Table 3 shows accuracies according to the size of the training data. The larger the data size, the higher the accuracy, because there are less unknown words and phrases. Note that LRM by Moses cannot calculate phrase orientation score for unknown phrases. Unlike conventional LRM, NRMs can construct a phrase

[5] We mix training data in the test data when we calculate the accuracy of LRM because the score can be calculated only for known phrases. Since the NRM can assign a score to unknown phrases, we use only the training data for NRMs.

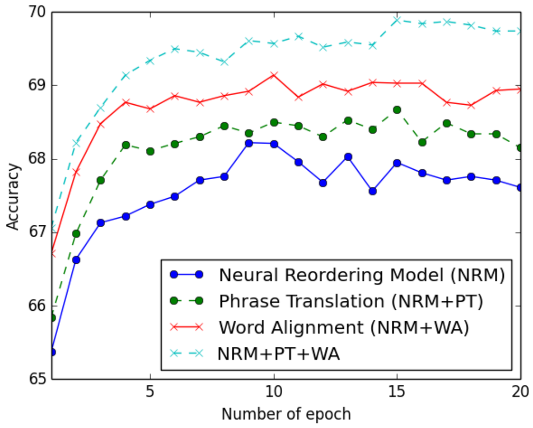

Figure 3: The accuracy of reordering at each epoch.

	Mono	Swap	D_{right}	D_{left}	Acc.
The rate of phrases including NULL Alignment [%]	25.8	45.9	40.8	44.5	
NRM	66.84	57.67	66.79	58.18	62.83
NRM+PT+WA	66.71	62.14	70.56	62.42	66.05

Table 4: Recall and accuracy of reordering phrases that contain NULL alignment.

vector even if the phrase in the test data is not included in the training data. As a result, the accuracy of the trained NRM is superior to that of LRM, only seeing 50K instances.

When we increase the size of the training data, the number of unknown words and unknown phrases decreases and the accuracy is improved further. However, most of the unknown words in the training corpus are named entities such as, “清水寺 (Kiyomizu-dera Temple),” which is a Japanese famous temple, because there are many traditional named entities in the KFTT corpus. Furthermore, it is possible that a new unknown word not in the phrase table appears in decoding. Therefore we expect NRMs to exhibit higher accuracy than LRM owing to their ability to recognize the unknown word.

5.2.2 Number of Epochs

Figure 3 shows the reordering accuracy at each epoch. Our proposed NRM+PT+WA method always achieves better accuracy than the baseline method of NRM. The accuracy is maximized around the 10th epoch in the test data. After that, the accuracy gradually decreases. The test loss shows the same tendency (negatively correlated with accuracy).

5.2.3 Phrase Translation Probabilities

To investigate the relation between the phrase translation feature and accuracy of our proposed method, we bin the test data into each phrase translation probability and evaluate the reordering accuracy.

As a result, the reordering accuracy does not improve in the cases where the translation probability is either too high or too small (e.g., the probability is more than 0.1 or less than 0.001), but overall performance improves a little. In a future study, we investigate as to why the translation probability is helpful for a reordering model.

5.2.4 NULL Alignment

To investigate the relationship between the NULL alignment and the accuracy of our proposed method, we evaluate only the instances when the target side phrases $\overline{e}_i$, $\overline{e}_{i-1}$ contain the words that have at least one NULL alignment. There are 3,788 such instances in the test data.

Table 4 shows the rate of instances including the NULL alignment for each reordering orientation and the accuracy of the corresponding reordering phrases. Considering each reordering orientation, the proposed method improves the recall over the plain NRM by approximately 4 points in each orientation

Method	ja-en			en-ja		
	BLEU	RIBES	WER	BLEU	RIBES	WER
LRM	18.45	65.64	79.87	21.37	67.01	84.73
NRM	19.16*	66.30	79.15*	**22.69***	68.64*	81.68*
NRM+PT+WA	**19.31***	**66.39**	**78.90***	22.61*	**68.65***	**81.57***

Table 5: Evaluation of translation quality. The symbols of * means statistically significant difference for LRM in bootstrap resampling ($p < 0.01$).

of Swap, $\mathrm{D}_{\mathrm{right}}$, and $\mathrm{D}_{\mathrm{left}}$, whereas that of Mono is not improved. This result suggests that the instances of Mono are not affected much by the NULL alignment, because they contain less NULL alignment (See the top row in Table 4). Overall, as compared with the NRM, our proposed method using phrase translation and word alignment improves the accuracy by 3.17 points (1.5 points higher than that of all the test data) for instances including the NULL alignment.

5.3 MT Evaluation

We investigate whether our reordering system improves translation accuracy. We use our reordering model for N-best re-ranking and optimize BLEU (Papineni et al., 2002) using minimum error rate training (MERT) (Och, 2003). We output a 1,000-best candidate list of translations that Moses generated for development data and replace the lexical reordering score of Moses with the score of the proposed method. Then, we re-tune the weights of the Moses features using MERT again. BLEU-4, RIBES (Isozaki et al., 2010a) and WER are used as measures for evaluation.

Table 5 shows the BLEU, RIBES and WER scores of the basic system and our proposed system. Bold scores represent the highest accuracies. When we compare the plain NRM and the proposed method with LRM, we confirm significant differences in BLEU, RIBES and WER scores on Japanese-to-English and English-to-Japanese translations using bootstrap resampling. Unfortunately, the proposed method is not able to identify significant differences in comparison with NRM. The reordering accuracy does not necessarily relate to the translation accuracy because we make the training and test data without checking the decoding step. We consider this to be partly of the reason why the BLEU score did not improve.

We conduct ablation tests to investigate which reordering orientation contributes most to BLEU score. The results show that Swap, which contains mostly NULL alignment, accounts for 0.17 points of improvement of the BLEU score in the proposed method. Other labels contribute only 0.01 - 0.05 points. Consequently, we consider that there is little influence on the translation results, because the change in each label of reordering is small, although the reordering accuracy rate of the NRM and the proposed method differ by 1.67 points.

In addition, we conducted human evaluation on Japanese-English translation by randomly choosing 100 sentences from test data. Two evaluators compared the proposed method with NRM fluency and adequacy. As a result, the proposed method improved fluency (NRM:NRM+PT+WA = 17.5:20) but not adequacy (NRM:NRM+PT+WA = 19:14.5). Although the outputs of two methods are similar, the proposed method favored fluent translation and resulted in slight improvements in BLEU and RIBES.

6 Related Work

There are several studies on phrase reordering of statistical machine translation. They are divided into three groups: in-ordering such as distance-based reordering (Koehn et al., 2003) and lexicalized reordering (Tillmann, 2004; Koehn et al., 2007; Galley and Manning, 2008), pre-ordering (Collins et al., 2005; Isozaki et al., 2010b; Wu et al., 2011; Hoshino et al., 2015; Nakagawa, 2015), and post-ordering (Sudoh et al., 2011). In-ordering is performed during decoding, pre-ordering is performed as pre-processing before decoding and post-ordering is executed as post-processing after decoding. In this section, we explain other reordering methods other than lexicalized reordering.

In early studies on PBSMT, a simple distance-based reordering penalty was used (Koehn et al., 2003).

It worked fairly well for some language pairs with similar word order such as English-French but is not appropriate for distant language pairs including Japanese-English. Lexicalized reordering model (LRM) (Tillmann, 2004; Koehn et al., 2007; Galley and Manning, 2008) introduced lexical constraints of the phrase reordering and not just penalizing long-distance reordering.

Pre-ordering methods can be divided into two types: (1) Rule-based preprocessing methods (Collins et al., 2005; Isozaki et al., 2010b) parse source sentences and reorder the words using hand-crafted rules. (2) Discriminative pre-ordering models (Tromble and Eisner, 2009; Wu et al., 2011; Hoshino et al., 2015; Nakagawa, 2015) learn whether children of each node should be reordered using (automatically) aligned parallel corpus. However, pre-ordering models cannot use the target language information in decoding. Therefore, optimizing phrase ordering using target-side features like phrase translation probability and word alignment is not possible, as done in our proposed method.

Post-ordering methods (Sudoh et al., 2011; Goto et al., 2012) are sometimes used in Japanese-to-English translation. They first translate Japanese input into head final English texts, then reorder head final English texts into English word orders. Post-ordering methods have the advantage of being able to use syntactic features at low computational cost, but need an accurate parser on the target side.

7 Conclusion

In this study, we improved a neural reordering model in PBSMT using phrase translation and word alignment. We proposed phrase translation and word alignment features to construct phrase vectors. The experimental results demonstrate that our proposed method improves the accuracy of phrase reordering. In addition, we showed that the proposed method was effective when phrase pairs included the NULL alignment. Evaluation on Japanese-to-English and English-to-Japanese translations indicated that the proposed method does not exhibits significant improvements in BLEU compared with those of the neural reordering model (Li et al., 2014). In the future, we plan to integrate our reordering model into attentional neural machine translations.

Acknowledgments

We thank the anonymous reviewers for their helpful comments on earlier versions. We also thank Yui Suzuki and Aizhan Imankulova for their annotation.

References

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *ACL*.

Yiming Cui, Shijin Wang, and Jianfeng Li. 2016. Lstm neural reordering feature for statistical machine translation. In *NAACL*.

Michel Galley and Christopher D Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP*.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2012. Post-ordering by parsing for Japanese-English statistical machine translation. In *ACL*.

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, Katsuhiko Hayashi, and Masaaki Nagata. 2015. Discriminative preordering meets Kendall's τ maximization. In *ACL-IJCNLP*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *EMNLP*.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for SOV languages. In *WMT*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL-HLT*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, and Richard Zens. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL (demo)*.

Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for ITG-based translation. In *EMNLP*.

Peng Li, Yang Liu, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. 2014. A neural reordering model for phrase-based translation. In *COLING*.

Masaaki Nagata, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi. 2006. A clustered global phrase reordering model for statistical machine translation. In *COLING-ACL*.

Tetsuji Nakagawa. 2015. Efficient top-down BTG parsing for machine translation preordering. In *ACL-IJCNLP*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Post-ordering in statistical machine translation. In *MT Summit*.

Christoph Tillmann. 2004. A unigram orientation model for statistical machine translation. In *HLT-NAACL*.

Seiya Tokui, Kenta Oono, and Shohei Hido. 2015. Chainer: a next-generation open source framework for deep learning. In *NIPS Workshop on Machine Learning Systems*.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *EMNLP*.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011. Extracting pre-ordering rules from predicate-argument structures. In *IJCNLP*.