**Noname manuscript No.**
(will be inserted by the editor)

# Learning Local Word Reorderings for Hierarchical Phrase-based Statistical Machine Translation

**Jingyi Zhang · Masao Utiyama · Eiichro Sumita · Hai Zhao · Graham Neubig · Satoshi Nakamura**

Received: date / Accepted: date

**Abstract** Statistical models for reordering source words have been used to enhance hierarchical phrase-based statistical machine translation. There are existing word reordering models that learn reorderings for any two source words in a sentence or only for two contiguous words. This paper proposes a series of separate sub-models to learn reorderings for word pairs with different distances. Our experiments demonstrate that reordering sub-models for word pairs with distances less than a specific threshold are useful to improve translation quality. Compared with previous work, our method more effectively and

An earlier version of this paper (zhang et al, 2015) was published as a short paper in ACL 2015. We extended this paper, including comparison with Cao et al (2014)'s method, details about efficiency, results of using a unified model instead of separate sub-models and comparison of machine learning methods for our model.

Jingyi Zhang
National Institute of Information and Communications Technology
3-5 Hikaridai, Keihanna Science City, Kyoto 619-0289, Japan
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama, Ikoma, Nara 630-0192, Japan
E-mail: jingyizhang@nict.go.jp

Masao Utiyama, Eiichro Sumita
National Institute of Information and Communications Technology
3-5 Hikaridai, Keihanna Science City, Kyoto 619-0289, Japan
E-mail: mutiyama/eiichiro.sumita@nict.go.jp

Hai Zhao
Department of Computer Science and Engineering
Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
Shanghai Jiao Tong University, Shanghai 200240, China
E-mail: zhaohai@cs.sjtu.edu.cn

Graham Neubig, Satoshi Nakamura
Graduate School of Information Science, Nara Institute of Science and Technology
Takayama, Ikoma, Nara 630-0192, Japan
E-mail: neubig/s-nakamura@is.naist.jp

efficiently exploits helpful word reordering information, which improves a basic hierarchical phrase-based system by 2.4-3.1 BLEU and keeps the average time of translating one sentence under 10 seconds.

## 1 Introduction

Hierarchical phrase-based machine translation (Chiang, 2005) is capable of jointly expressing lexical choice and reordering with synchronous context-free grammars. However, selecting proper translation rules during decoding is a major challenge, as a large number of hierarchical rules can be applied to any source sentence.

Chiang (2005) used a log-linear model to compute rule weights with features similar to Pharaoh (Koehn et al, 2003). However, to select appropriate rules, more effective criteria are required, and much work has been done for better rule selection. He et al (2008) and Liu et al (2008) used maximum entropy approaches to integrate rich contextual information for target side rule selection. Cui et al (2010) proposed a joint model to select hierarchical rules for both source and target sides. Wang et al (2015) proposed to estimate the semantic similarity between nonterminals and their phrasal substitutions during decoding to favor translation rules with high similarities.

In addition, word or phrase reordering models have also been integrated into hierarchical phrase-based SMT (Hayashi et al, 2010; Huck et al, 2013; Nguyen and Vogel, 2013; Cao et al, 2014). Among these, Hayashi et al (2010) demonstrated the effectiveness of using word reordering within hierarchical phrase-based SMT by integrating Tromble and Eisner (2009)'s word reordering model into the hierarchical translation model.

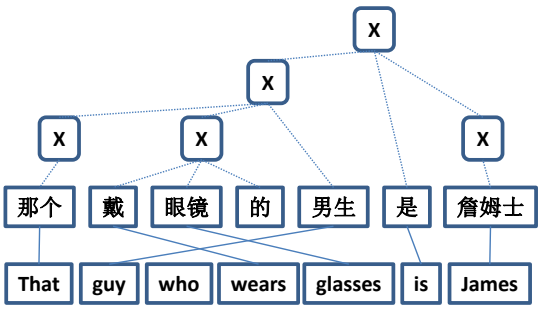


**Fig. 1** A translation example.

The word reordering model helps score the reordering of words during translation, reducing the number of reordering errors caused by selecting the wrong translation rules or using them in the wrong order. Figure 1 shows a Chinese-to-English translation example that demonstrates how the word reordering model can help hierarchical phrase-based SMT. In this translation,

the rule "X1 X2 男生→ X1 *guy* X2" is applied to the Chinese input sentence. Alternatively, the rule "X1 X2 男生→ X1 X2 *guy*" can also be applied to the input sentence but will cause an incorrect translation in this particular case. If the word pair "眼镜(*glasses*) 男生(*guy*)" can be predicted to be reversed by the reordering model, then the translation system will prefer the translation rule adopted in Figure 1 that reverses this word pair.

Hayashi et al (2010)'s method demonstrated that reordering models can help resolve this problem, but one deficiency of the method is that the computation cost is quite expensive, as the model predicts reorderings for all word pairs in the input sentence. That is, if the input sentence length is $n$, this model needs to calculate reorderings for $O\left(n^2\right)$ word pairs.

In contrast, Feng et al (2013) proposed a word reordering model that only estimates reorderings for two contiguous source words, and applied their model to phrase-based SMT. Because this limitation of the model results in a complexity of $O\left(n\right)$, it reduces computation cost significantly compared with Tromble and Eisner (2009)'s model, and it still achieves significant reordering improvements over the baseline system.

In this paper, we strike a balance between these two approaches. Specifically, we incorporate word reordering information into hierarchical phrase-based SMT by training a series of separate reordering sub-models for word pairs with different distances. In Chinese-to-English and Japanese-to-English translation experiments, the translation performance achieved consistent improvements as more sub-models for longer distance reorderings were integrated, but the improvement levelled off quickly. In other words, sub-models for reordering distance longer than a given threshold did not improve translation quality significantly. In the experiments section, we also give detailed analyses of why reordering sub-models for longer distances were not as useful for translation quality.

By predicting local reorderings shorter than a given threshold, our model exploits more reordering information than Feng et al (2013), while preventing the quadratic explosion in computation time of Hayashi et al (2010)'s method. In addition, our reordering model learned by feed-forward neural network (FNN) achieves better performance than the more traditional linear model.

This paper is organized as follows: in Section 2, we review previous related work; Section 3 describes our approach; we present experiments in Section 4 and make conclusions in Section 5.

## 2 Related Work

Reordering modeling has been extensively studied for phrase-based SMT (Koehn et al, 2003). Because it is bilingual phrase pairs that are used as the translation unit for phrase-based SMT, most reordering models used in phrase-based SMT learn reordering of phrase pairs and implicitly make an assumption that

word reorderings within phrase pairs are correct (Koehn et al, 2005; Zens and Ney, 2006; Ni et al, 2009; Li et al, 2014).

These existing reordering models are not suitable for hierarchical phrase-based SMT, which does not use phrase pairs as translation units. Huck et al (2013) introduced a reordering model for hierarchical phrase-based translation, which determines and estimates the orientations of nonterminals in translation rules. Nguyen and Vogel (2013) proposed to integrate phrase-based reordering features into hierarchical phrase-based SMT by mapping a HPB derivation into a discontinuous phrase-based translation path, which enhanced the HPB model significantly. However, there are some forms of HPB rules which cannot be mapped into a reasonable sequence of phrase pairs and non-terminals. Cao et al (2014) proposed a lexicalized reordering model which is built directly on HPB rules and compatible with any kind of HPB rules.

Compared to the proposed word reordering model, these phrase-based reordering models limited to learning the reordering of contiguous phrases. When phrase length is short, in extreme cases, when phrase length is one, their models only learn reordering for contiguous word pairs, while our model releases such a constraint and can be applied to two source words with longer distances. And our experiments showed that reordering prediction for word pairs with distance 2,3... can improve translation qualities significantly as well.

Bisazza and Federico (2013) modelled reordering as the problem of deciding whether a given input word should be translated after another. However, two source words that are aligned to contiguous target words may have long distance, which makes this classification task harder than determining local reorderings as in our model.

There is also some work that exploits syntactic information to help reorderings in hierarchical phrase-based translation (Gao et al, 2011; Kazemi et al, 2015; Marton and Resnik, 2008). However, high quality parsers are not always available and parsing errors can influence the performance of these methods significantly.

## 3 Our Approach

In this section, we introduce the proposed model and how to integrate it into the translation system.

### 3.1 Modeling

Let $e_1^m = e_1, \ldots, e_m$ be a target translation of $f_1^l = f_1, \ldots, f_l$ and $A$ be word alignment links between $e_1^m$ and $f_1^l$. Our model estimates the reordering probability of the source sentence as follows:

$$\Pr\left(f_1^l, e_1^m, A\right) \approx \prod_{n=1}^{N} \prod_{i,j:1\leq i<j\leq l, j-i=n} \Pr\left(f_1^l, e_1^m, A, i, j\right) \quad (1)$$

where $\Pr\left(f_1^l, e_1^m, A, i, j\right)$ is the reordering probability of the word pair $\langle f_i, f_j \rangle$ during translation and $N$ is the maximum distance considered by the reordering model, which is empirically determined by supposing that estimating reorderings longer than $N$ does not improve translation performance significantly.

Previous word reordering models (Tromble and Eisner, 2009; Feng et al, 2013) consider the reordering of a source word pair to be reversed or not. When a source word is aligned to several discontinuous target words, it can be hard to determine if a word pair is reversed or not as shown in Figure 2. They solved this problem by only using one alignment from multiple alignment links and ignoring the others. For example, in Figure 2 the alignment between "放弃(*give up*)" and "*up*" is ignored. In contrast, our model handles all alignment links to cover more word reordering patterns.

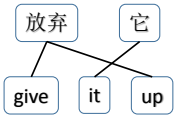


**Fig. 2** Multiple alignment links.

Suppose that $f_i$ is aligned to $\pi_i\,(\pi_i \geq 0)$ target words. When $\pi_i > 0$, $\{a_{ik}|1 \leq k \leq \pi_i\}$ stands for the positions of target words aligned to $f_i$. If $\pi_i = 0$ or $\pi_j = 0$, $\Pr\left(f_1^l, e_1^m, A, i, j\right) = 1$,[1] otherwise,

$$\Pr\left(f_1^l, e_1^m, A, i, j\right) = \prod_{u=1}^{\pi_i} \prod_{v=1}^{\pi_j} \Pr\left(o_{ijuv} | f_{i-3}^{j+3}, e_{a_{iu}}, e_{a_{jv}}\right) \tag{2}$$

where

$$o_{ijuv} = \begin{cases} 0\ (a_{iu} \leq a_{jv}) \\ 1\ (a_{iu} > a_{jv}) \end{cases}. \tag{3}$$

Here, $o_{ijuv}$ indicates whether the translation $e_{a_{iu}}$ of $f_i$ and the translation $e_{a_{jv}}$ of $f_j$ should be reordered in the target side.

Next, we need a model to estimate the probability of each $o_{ijuv}$. As mentioned in the introduction, we train a series of sub-models,

$$M_1, M_2, \ldots, M_N$$

to learn reorderings for word pairs with different distances. In other words, for the word pair $\langle f_i, f_j \rangle$ with distance $j - i = n$, its reordering probability $\Pr\left(o_{ijuv} | f_{i-3}^{j+3}, e_{a_{iu}}, e_{a_{jv}}\right)$ is estimated by $M_n$. Different sub-models are trained and integrated into the translation system separately.

[1] In translation experiments, we also tried adding a new penalty feature (how many source words in the input sentence are unaligned) to penalize unaligned words. However, this feature did not influence translation performance significantly.

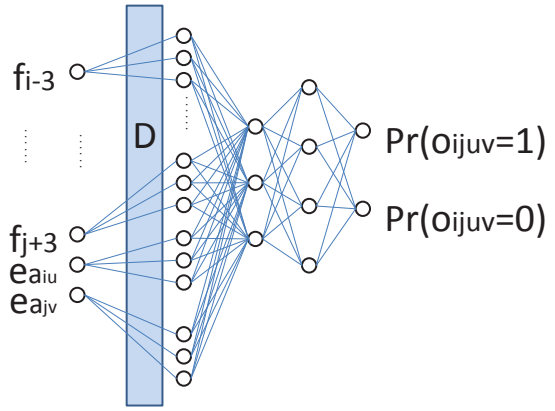


**Fig. 3** Neural reordering model.

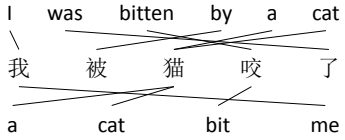


**Fig. 4** Alignment examples.

Each sub-model $M_n$ is implemented by an FNN, which has the same structure with the neural language model in Vaswani et al (2013) as shown in Figure 3. The input to $M_n$ is a sequence of $n+9$ words: $f_{i-3}^{j+3}, e_{a_{iu}}, e_{a_{jv}}$. In Hayashi's model, only source-side contextual features were used. Because multiple correct translations may exist for an input sentence and different translations need different reorderings of the source sentence as shown in Figure 4, our model also integrates target-side features $e_{a_{iu}}, e_{a_{jv}}$ for reordering. The input layer projects each word into an embedding vector using a matrix of input word embeddings, $D$. These embeddings are followed by two hidden layers that combine all of the input embeddings. Thus unlike when using the averaged perceptron algorithm of Hayashi et al (2010), we do not need to manually design features to achieve high accuracy. The output layer has two neurons that calculate $\Pr(o_{ijuv} = 1)$ and $\Pr(o_{ijuv} = 0)$.

The backpropagation algorithm (Rumelhart et al, 1986) is used to train the parameters for each reordering sub-model. The training instances for each sub-model are extracted from the word-aligned parallel corpus according to Algorithm 1. For example, the word pair "戴(*wears*) 男生(*guy*)" in Figure 1 will be extracted as a positive instance for $M_3$. The input of this instance is as follows: "<*s*> <*s*> 那个 戴 眼镜 的 男生 是 詹姆士 </*s*> *wears guy*", where "<*s*>" and "</*s*>" represent the beginning and ending of a sentence. If a word never occurs or only occurs once in the training corpus, we replace it with a special symbol "<*unk*>".

**Algorithm 1** Extract training instances.

**Require:** A pair of parallel sentence $f_1^l$ and $e_1^m$ with word alignment links.
**Ensure:** Training examples for $M_1, M_2, \ldots, M_N$.

```
for i = 1 to l − 1 do
  for j = i + 1 to l do
    if j − i ≤ N then
      for u = 1 to π_i do
        for v = 1 to π_j do
          if a_iu ≤ a_jv then
            (f_{i−3}^{j+3}, e_{a_iu}, e_{a_jv}, 0) is a negative instance for M_{j−i}
          else
            (f_{i−3}^{j+3}, e_{a_iu}, e_{a_jv}, 1) is a positive instance for M_{j−i}
          end if
        end for
      end for
    end if
  end for
end for
```

### 3.2 Decoding

In the hierarchical phrase-based model, a translation rule $r$ is defined as:

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle ,$$

where $X$ is a nonterminal, $\gamma$ and $\alpha$ are respectively source and target strings of terminals and nonterminals, and $\sim$ is the alignment between nonterminals and terminals in $\gamma$ and $\alpha$.

Each rule has several features and the feature weights can be tuned by any optimization algorithm (Och, 2003; Chiang, 2012; Hopkins and May, 2011). To integrate our model into the hierarchical phrase-based translation system, a new feature $score_n(r)$ is added to each rule $r$ for each $M_n$.[2]

Suppose that $r$ is applied to the input sentence $f_1^l$, where

- $r$ covers the source span $[f_\varphi, f_\vartheta]$
- $\gamma$ contains nonterminals $\{X_k | 1 \leq k \leq K\}$
- $X_k$ covers the span $[f_{\varphi_k}, f_{\vartheta_k}]$

Then,

$$score_n(r) = \sum_{\langle i,j \rangle \in S - \bigcup_{k=1}^{K} S_k \wedge j-i=n} \log\left(\Pr\left(f_1^l, e_1^m, A, i, j\right)\right) \tag{4}$$

where

$$S : \{\langle i,j \rangle | \varphi \leq i < j \leq \vartheta\}$$
$$S_k : \{\langle i,j \rangle | \varphi_k \leq i < j \leq \vartheta_k\}$$

For example, if a rule "X1 X2 男生→ X1 *guy* X2" is applied to the input sentence in Figure 1, then

[2] Note that these scores are correspondingly calculated for different sub-models $M_n$ and the sub-model weights are tuned separately.

$$[f_\varphi, f_\vartheta] = [1,5]\,; [f_{\varphi_1}, f_{\vartheta_1}] = [1,1]\,; [f_{\varphi_2}, f_{\vartheta_2}] = [2,4]$$
$$S - \bigcup_{k=1}^{K} S_k = \left\{ \begin{array}{l} \langle 1,2\rangle, \langle 1,3\rangle, \langle 1,4\rangle, \langle 1,5\rangle, \\ \langle 2,5\rangle, \langle 3,5\rangle, \langle 4,5\rangle \end{array} \right\}$$

We can see from these reordering features that $score_n(r)$ cannot be calculated before decoding, because the information about $\{X_k | 1 \le k \le K\}$ is needed. And to calculate neural network probabilities, we also need source-side context information from the input sentence, which means feature scores can be different for one translation rule when it is applied to different source sentences. Thus, this new feature must be calculated separately for each input source sentence. However, since our model does not use target $n$-gram information, therefore, we do not need to consider future costs for these reordering features as we do when using $n$-gram language models in hierarchical phrase-based system.

One concern in using target features is the computational efficiency, because reordering probabilities have to be calculated during decoding. However, we can cache probabilities to reduce the expensive neural network computation using hash tables. That is, for each sequence $\left(f_{i-3}^{j+3}, e_{a_{iu}}, e_{a_{jv}}, o_{ijuv}\right)$, our model only needs to calculate the reordering probability $\Pr\left(o_{ijuv} | f_{i-3}^{j+3}, e_{a_{iu}}, e_{a_{jv}}\right)$ once. Note that we clear the cache for each input sentence. This is because the reordering probabilities calculated for one sentence are seldom used for other sentences, and the lookup will slow down somewhat as the hash table size grows.[3]

## 4 Experiments

In this section, we give detailed experimental results and analyses regarding our approach, and compare our approach with previous reordering models.

### 4.1 Setting

We evaluated the proposed approach for Chinese-to-English (CE) and Japanese-to-English (JE) translation tasks. The official datasets for the patent machine translation task at NTCIR-9 (Goto et al, 2011) were used in our experiments. The detailed statistics for training, development and test sets are given in Table 1.

In NTCIR-9, the development and test sets were both provided for the CE task while only the test set was provided for the JE task. Therefore, we used the sentences from the NTCIR-8 JE test set as the development set for the

[3] As we are using a cache, memory usage is a concern, but the size of the cache for each sentence is negligible compared to the size of the translation and language models, and thus the memory footprint is not increased significantly.

| | | | SOURCE | TARGET |
|---|---|---|---|---|
| CE | TRAINING | #Sents | 954K | |
| | | #Words | 37.2M | 40.4M |
| | | #Vocab | 288K | 504K |
| | DEV | #Sents | 2K | |
| | | #Words | 75.4K | 77.5K |
| | TEST | #Sents | 2K | |
| | | #Words | 55.5K | 58.1K |
| JE | TRAINING | #Sents | 3.14M | |
| | | #Words | 118M | 104M |
| | | #Vocab | 150K | 273K |
| | DEV | #Sents | 2K | |
| | | #Words | 74.6K | 66.5K |
| | TEST | #Sents | 2K | |
| | | #Words | 77.8K | 69.5K |

**Table 1** Data sets.

JE task. The word segmentation was done by BaseSeg (Zhao et al, 2006) for Chinese and Mecab[4] for Japanese.

To learn neural reordering models, the training and development sets were combined to obtain symmetric word alignments using GIZA++ (Och and Ney, 2003) and the *grow-diag-final-and* heuristic (Koehn et al, 2003). The reordering instances extracted from the aligned training and development sets were used as the training and validation data for learning neural reordering models. We trained our model on the training data iteratively and stopped the training process when validation perplexity stopped decreasing. The validation data was randomly split into two parts. One part was used to stop the training process and the other part was used to calculate accuracies of reordering models. Neural reordering models were trained by the toolkit NPLM (Vaswani et al, 2013). For the CE task, training instances extracted from all the 954K sentence pairs were used to train neural reordering models and the numbers of training instances are 40.0M, 38.4M, 37.1M, 36.0M for $M_1$, $M_2$, $M_3$, $M_4$, respectively. For the JE task, training instances are from 1M sentence pairs that were randomly selected from all the 3.14M sentence pairs and the numbers of instances are 38.8M, 36.6M, 35.5M, 34.3M for $M_1$, $M_2$, $M_3$, $M_4$, respectively.

We implemented Hayashi et al (2010)'s model to compare with our approach. The training instances for their model were extracted from the same sentence pairs as ours and instance sizes are 771.6M and 725.6M for CE and JE, respectively. We also implemented Cao et al (2014)'s phrase reordering model for comparison. The whole rule table extracted from the aligned training set was used as training data for their model.

For each translation task, a recent version of the Moses hierarchical phrase-based decoder (Koehn et al, 2007) with the training scripts was used as the baseline system. We used the default parameters for Moses. A 5-gram language model was trained on the target side of the training corpus by IRST

[4] http://sourceforge.net/projects/mecab/files/

| | | BASE | Hayashi | Cao | $M_1^1$ | $M_1^2$ | $M_1^3$ | $M_1^4$ |
|---|---|---|---|---|---|---|---|---|
| CE | Average | 33.14 | 34.36 | 34.70 | 34.66 | 35.82 | 35.85 | **35.93** |
| | Deviation | 0.19 | 0.14 | 0.13 | 0.09 | 0.26 | 0.18 | 0.20 |
| JE | Average | 30.03 | 30.92 | 31.69 | 31.53 | 32.11 | 32.50 | **32.58** |
| | Deviation | 0.18 | 0.21 | 0.16 | 0.19 | 0.18 | 0.17 | 0.13 |

**Table 2** Translation results (BLEU).

| CE | BASE | $M_1^1$ | $M_1^2$ | $M_1^3$ |
|---|---|---|---|---|
| $M_1^1$ | $\gg$ | | | |
| $M_1^2$ | $\gg$ | $\gg$ | | |
| $M_1^3$ | $\gg$ | $\gg$ | $-$ | |
| $M_1^4$ | $\gg$ | $\gg$ | $-$ | $-$ |
| JE | BASE | $M_1^1$ | $M_1^2$ | $M_1^3$ |
| $M_1^1$ | $\gg$ | | | |
| $M_1^2$ | $\gg$ | $\gg$ | | |
| $M_1^3$ | $\gg$ | $\gg$ | $\gg$ | |
| $M_1^4$ | $\gg$ | $\gg$ | $\gg$ | $-$ |

**Table 3** Significance test results using bootstrap resampling w.r.t. BLEU scores. The symbol $\gg$ represents a significant difference at the $p < 0.01$ level; $-$ means not significantly different at $p = 0.05$.

LM Toolkit[5] with improved Kneser-Ney smoothing. Since both CE and JE language pairs have quite different word orders, we set the distortion limit (max chart span) to be 20. The test set (2K sentences) contains 1.31K and 1.73K sentences with length longer than 20 for the CE and JE tasks, respectively.

We integrated our reordering models into BASE. Each sub-model weight was tuned by MERT (Och, 2003) together with other feature weights (language model, word penalty, etc.) under the log-linear framework (Och and Ney, 2002).

### 4.2 Result and Analysis

Table 2 gives detailed translation results and Table 3 shows significance test results using bootstrap resampling (Koehn, 2004). "Hayashi" represents the method of Hayashi et al (2010), "Cao" represents the method of Cao et al (2014) and "$M_1^j$ $(j = 1, 2, 3, 4)$" means that BASE was augmented with the reordering scores calcuated from a series of sub-models $M_1$ to $M_j$. For example, $M_1^3$ means $M_1$, $M_2$ and $M_3$ are integrated; $M_1^4$ means $M_1$, $M_2$, $M_3$ and $M_4$ are integrated. We ran MERT 4 times for each experiment and show the average BLEU score with the standard deviation.

We can see that, with 4 sub-models integrated, our method outperformed both the Hayashi and Cao models significantly. Note that integrating only $M_1$, which predicts reordering for two contiguous source words, has already given a BLEU improvement of 1.8% and 1.2% over BASE on CE and JE, respectively.

[5] http://hlt.fbk.eu/en/irstlm

As more sub-models for longer distance reordering are integrated, the translation performance improved consistently, although the improvement leveled off quickly. For the CE and JE tasks, $M_n$ with $n \geq 3$ and $n \geq 4$, respectively, did not give a further performance improvement at a significant level. We also did extra experiments with much longer reordering sub-models, in which we trained a model $M_{15}$ for word pairs with distance 15, then integrated both $M_{15}$ and $M_1^4$ into BASE. However, the translation results had no significant improvement compared to BASE augmented with $M_1^4$.

Why did the improvement level off quickly? In other words, why do long distance reordering models have much less leverage over translation performance than short ones?

First, the prediction accuracy decreases as the reordering distance increases. Table 4 gives prediction accuracies on the validation data for each sub-model. One reason for accuracy decreasing is that the input size of the sub-model grows as the reordering distance increases. Namely, there is more context information between words that are farther apart, which is harder to capture with limited training data and simple models that do not explicitly consider information about the syntactic structure of the sentence.

| Sub-model | $M_1$ | $M_2$ | $M_3$ | $M_4$ |
|---|---|---|---|---|
| CE | 93.9 | 92.8 | 92.2 | 91.2 |
| JE | 92.9 | 91.3 | 90.1 | 89.3 |

**Table 4** Classification accuracy of our model (%)

Second, we attribute the decrease in influence of the longer reordering models to the redundancy of the predictions among the reordering sub-models. That is, a long distance word reordering can often be determined by a series of shorter word reordering pairs. For example, in Figure 1, if word pairs "男生(*guy*) 是(*is*)" and "是(*is*) 詹姆士(*James*)" are both predicted to be not reversed, the reordering for "男生(*guy*) 詹姆士(*James*)" can be logically determined to be not reversed without prediction. As a result, sometimes predictions for longer reorderings will not be useful for the translation process. In fact, although the longest distance of source words in Figure 1 is 6, the longest distance of word pairs whose reorderings need to be predicted in order to accurately determine the ordering of all the words is 4.

But still, some predictions for longer reorderings are useful. For example, the reordering of "戴(*wears*) 男生(*guy*)" cannot be determined when "戴(*wears*) 眼镜(*glasses*)" is predicted to be not reversed and "眼镜(*glasses*) 男生(*guy*)" is reversed. This is the reason why the translation performance improves as more sub-models are integrated.

Table 5 gives a translation example to demonstrate how our model improves the reordering during translating.[6] As shown, the distance between source

[6] Note that "4" and "5" in source and target sentences are original source and target words. This sentence pair is from a patent translation corpus and there is a figure in the article, where the light source is labeled as 4 and the optical fiber is labeled as 5.

| | |
|---|---|
| S | 该(the) 照明(lighting) 设备(device) 1 还(further) 具有(has) 耦合(coupled) 到 固态(solid-state) 光源(light source) 4 的 光纤(optical fiber) 5 。 |
| R | the lighting device 1 further has an optical fiber 5 which is coupled to the solid-state light source 4 . |
| B | the illumination apparatus 1 also has coupled to the optical fiber of the solid-state light source 4 5 . |
| $M_1^1$ | the illumination apparatus 1 also has a fiber coupled to the solid-state light source 4 5 . |
| $M_1^2$ | the illumination apparatus 1 also has a fiber coupled to the solid-state light source 4 5 . |
| $M_1^3$ | the illumination apparatus 1 also has a fiber 5 coupled to the solid-state light source 4 . |
| $M_1^4$ | the illumination apparatus 1 also has a fiber 5 coupled to the solid-state light source 4 . |

**Table 5** Translation examples. S: input sentence, R: reference sentence, B: translation result of BASE, $M_1^j$ ($j = 1, 2, 3, 4$): translation result with $M_1^j$ being integrated.

| Reordering Distance | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| CE | 90.1 | 88.3 | 87.0 | 85.6 |
| JE | 85.3 | 81.9 | 80.6 | 78.8 |

**Table 6** Classification accuracy of Hayashi model (%).

| Reordering Distance | | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| Hayashi | CE | 82.4 | 76.5 | 73.6 | 72.6 |
| | JE | 67.8 | 60.9 | 57.0 | 55.8 |
| Our model | CE | 95.3 | 93.8 | 92.7 | 91.6 |
| | JE | 93.9 | 91.6 | 90.3 | 89.1 |

**Table 7** Classification accuracy for one-to-one alignment links (%).

words “4” and “5” is 3, and after $M_3$ being integrated into BASE, this word pair can be correctly reordered.

Note that if we only integrate $M_4$ into BASE, the translation quality of BASE was improved in preliminary experiments. However, $M_4$ cannot predict reorderings for word pairs with distance less than 4. So $M_1^3$ will be still needed for predicting reorderings of word pairs with distance 1,2,3. But after $M_1^3$ being integrated, $M_4$ did not provide a large improvement due to the redundancy of the predictions among different reordering sub-models.

Now we analyze the reasons that our model outperformed the Hayashi and Cao models, respectively, as shown in Table 2.

Table 6 shows the reordering prediction accuracies of Hayashi model for word pairs with different distances. Note that Hayashi’s model predicts reorderings for all word pairs, but only prediction accuracies for word pairs with distance 4 or less are shown. The definitions of classification accuracy for our method and Hayashi’s model are slightly different. In Hayashi’s model, one word pair is counted as one reordering instance. In contrast, one word pair with multiple alignment links may contain several reordering instances for our model, and if one source word is not aligned to any target word, we do not consider the reordering about this source word. For a direct comparison, Ta-

ble 7 shows the reordering classification accuracy for two words that were both aligned to exactly one target word. As shown in Table 7, our approach learned reorderings much better than the Hayashi model. This is easy to understand, since our model was trained by feed-forward neural networks on a high dimensional space and incorporated rich context information, while Hayashi's model used the averaged perceptron algorithm and manually crafted features.

Our model also outperformed Cao's model, which already had a strong improvement compared to BASE. Since their model needs to be trained on the whole rule table and the hierarchical translation rule table is quite large, the training process will be very time-consuming. Thus they only used simply relative frequency and the add 0.5 smoothing technique to estimate the reordering probability. In other words, it is hard to use other features in their model due to efficiency issues. Besides, their model only estimates reorderings for contiguous phrase pairs.

### 4.3 Efficiency

In this sub-section, we perform a group of experiments to show how much the caching strategy can bring about efficiency improvements. We used a computer with Xeon E5-4650 CPU and CentOS 6.3 to translate all input sentences in the CE test set. Table 8 gives the hit rate (HR) of caching and the average translation time for one sentence with and without caching. The average translation time for BASE was 3.92 seconds.

$$HR = \frac{T_{cache}}{T_{cache} + T_{calculate}}$$

Here, $T_{cache}$ was the number of times that we could find the reordering probability in the cache; $T_{calculate}$ was the number of times that the reordering probability could not be found in the cache and then had to be calculated by the neural reordering model.

| Sub-models | Caching (sec) | No Caching (sec) | Hit Rate (%) |
|---|---|---|---|
| $M_1^1$ | 4.60 | 102.65 | 99.85 |
| $M_1^2$ | 6.56 | 212.95 | 99.84 |
| $M_1^3$ | 8.50 | 330.27 | 99.86 |
| $M_1^4$ | 10.11 | 442.39 | 99.88 |

**Table 8** Translation time and hit rate.

According to the results in Table 8, we can see that the hit rates were quite high. Using a cache in decoding, in most cases we just need to perform look up in hash tables to get the reordering probabilities. This results in high efficiency as hash table lookup is much faster than calculating neural networks.

### 4.4 One Model vs. Mutiple Sub-models

Different from using one model to learn reordering for all word pairs, our model learns reordering with several separate sub-models. Different sub-models can be trained entirely separately, and we can take advantage of this easy parallelism to train models in a more reasonable time.

However, theoretically, one unified model will have better performance since separate sub-models do not share training instances. Suppose that the training corpus contains these two sentences "*I like sunny days*" and "*I like sunny and warm days*". The word pair "*I days*" occurs twice in the training corpus for the unified model and once for two separate sub-models, respectively. This indicates that the unified model suffers less from data sparsity. To test these effects, we did some extra experiments and let one neural network learn for word pairs with distance 4 or less. This neural network has the same structure as $M_4$ with 13 inputs. For word pairs with distance 1,2,3,4, the inputs are

$$f_{i-3}, \ldots, f_i, f_j, \ldots, f_{j+3}, e_{a_{iu}}, e_{a_{jv}}, null, null, null$$
$$f_{i-3}, \ldots, f_i, f_j, \ldots, f_{j+3}, e_{a_{iu}}, e_{a_{jv}}, f_{i+1}, null, null$$
$$f_{i-3}, \ldots, f_i, f_j, \ldots, f_{j+3}, e_{a_{iu}}, e_{a_{jv}}, f_{i+1}, f_{i+2}, null$$
$$f_{i-3}, \ldots, f_i, f_j, \ldots, f_{j+3}, e_{a_{iu}}, e_{a_{jv}}, f_{i+1}, f_{i+2}, f_{i+3}$$

Here, *null* is a specific symbol that represents a default position.

Table 9 shows the reordering prediction accuracy of this model for word pairs with different distances. Table 10 gives the translation result after integrating this model into BASE to predict reordering for word pairs with different distances. The corresponding original results using multiple sub-models are also shown for a direct comparison.

| Reordering | Distance | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| One | CE | 93.9 | 93.0 | 92.2 | 91.3 |
| | JE | 92.8 | 91.6 | 90.3 | 89.2 |
| Multiple | CE | 93.9 | 92.8 | 92.2 | 91.2 |
| | JE | 92.9 | 91.3 | 90.1 | 89.3 |

**Table 9** Classification accuracy of using one unified model (%).

| Reordering | Distance | 1 | 1,2 | 1,2,3 | 1,2,3,4 |
|---|---|---|---|---|---|
| One | CE | 34.50 | 35.70 | 35.50 | 35.74 |
| | JE | 31.42 | 32.00 | 32.47 | 32.54 |
| Multiple | CE | 34.66 | 35.82 | 35.85 | 35.93 |
| | JE | 31.53 | 32.11 | 32.50 | 32.58 |

**Table 10** Translation performance of using one unified model (BLEU).

As can be seen, using one model or using multiple sub-models to learn reordering have nearly the same classification and translation performance. This

shows that using separate models does not hurt performance while keeping the merit of training efficiency. This means that although one unified model is theoretically more robust to sparse data, when the training corpus becomes large, separate sub-models can also learn the reorderings well, and the performance difference between one unified model and separate sub-models is negligible.

### 4.5 Comparison of Machine Learning Methods

To analyze the influence of machine learning method choice for our model, we also tried the averaged perceptron algorithm to learn each sub-model, which was used by Hayashi et al (2010) for their model training. The features are given in Table 11.

| |
|---|
| $f_k, f_k e_{a_{iu}}, f_k e_{a_{jv}}\ (i-3 \leq k \leq j+3)$ |
| $f_i f_j, f_i f_j e_{a_{iu}}, f_i f_j e_{a_{jv}}, f_i f_j e_{a_{iu}} e_{a_{jv}}$ |
| $f_i f_k, f_j f_k, f_i f_k e_{a_{iu}}, f_j f_k e_{a_{jv}},$ $f_i f_j f_k, f_i f_j f_k e_{a_{iu}}, f_i f_j f_k e_{a_{jv}}, f_i f_j f_k e_{a_{iu}} e_{a_{jv}}$ $(i-3 \leq k \leq j+3, k \neq i, k \neq j)$ |

**Table 11** Features.

Table 12 and 13 show prediction accuracies and translation performance using AP algorithm. The corresponding original results using FNN are also shown for a direct comparison.

| Sub-model | | $M_1$ | $M_2$ | $M_3$ | $M_4$ |
|---|---|---|---|---|---|
| AP | CE | 93.4 | 92.2 | 90.7 | 89.7 |
| | JE | 92.3 | 90.6 | 89.1 | 87.7 |
| FNN | CE | 93.9 | 92.8 | 92.2 | 91.2 |
| | JE | 92.9 | 91.3 | 90.1 | 89.3 |

**Table 12** Classification accuracy of using AP for model training (%).

| Sub-models | | $M_1^1$ | $M_1^2$ | $M_1^3$ | $M_1^4$ |
|---|---|---|---|---|---|
| AP | CE | 34.27 | 34.54 | 34.47 | 34.43 |
| | JE | 30.74 | 31.79 | 31.60 | 31.92 |
| FNN | CE | 34.66 | 35.82 | 35.85 | 35.93 |
| | JE | 31.53 | 32.11 | 32.50 | 32.58 |

**Table 13** Translation performance of using AP for model training (BLEU).

As can be seen, classifiers learned by feedforward neural networks perform better than the averaged perceptron algorithm and the translation perfor-

mance with the neural reordering model outperformed that with the reordering model learned by the AP algorithm, which means the difference of training methods is an important factor explaining why our model outperformed Hayashi et al (2010)'s model in Table 2. However, FNNs are not suitable for use in the Hayashi model since the training and decoding time for FNN is already quite long. Using FNN for Hayashi et al (2010)'s model will cost nearly one minute to translate one sentence according to our experiments, while our most complex model took about 10 seconds as shown in Table 8.[7]

## 5 Conclusion

In this paper, we adopt a series of separate sub-models to reorder source word pairs with different distances and integrate this model into hierarchical phrase-based SMT. Experiments and analyses have shown that only reordering predictions for word pairs with distances less than a specific threshold improved translation performance clearly, and longer distance reordering sub-models were not as helpful for translation quality. With only sub-models for short distance reorderings being used, training and decoding for our model are much more efficient compared to previous models, while keeping the majority of helpful word reordering information. Besides, our reordering model is learned by feed-forward neural networks and incorporates rich context information for better performance. On both Chinese-to-English and Japanese-to-English translation tasks, the proposed model outperformed the previous models significantly.

## References

Bisazza A, Federico M (2013) Dynamically shaping the reordering search space of phrase-based statistical machine translation. Transactions of the Association for Computational Linguistics 1:327–340

Cao H, Zhang D, Li M, Zhou M, Zhao T (2014) A lexicalized reordering model for hierarchical phrase-based translation. In: The 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, pp 1144–1153

Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: The 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, Michigan, pp 263–270

Chiang D (2012) Hope and fear for discriminative training of statistical translation models. The Journal of Machine Learning Research 13(1):1159–1187

Cui L, Zhang D, Li M, Zhou M, Zhao T (2010) A joint rule selection model for hierarchical phrase-based translation. In: The 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp 6–11

[7] Cache was used in all experiments.

Feng M, Peter JT, Ney H (2013) Advancements in reordering models for statistical machine translation. In: The 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, pp 322–332

Gao Y, Koehn P, Birch A (2011) Soft dependency constraints for reordering in hierarchical phrase-based translation. In: The 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK., pp 857–868

Goto I, Lu B, Chow KP, Sumita E, Tsou BK (2011) Overview of the patent machine translation task at the NTCIR-9 workshop. In: The 9th NII Test Collection for IR Systems Workshop Meeting, Tokyo,Japan, pp 559–578

Hayashi K, Tsukada H, Sudoh K, Duh K, Yamamoto S (2010) Hierarchical phrase-based machine translation with word-based reordering model. In: The 23rd International Conference on Computational Linguistics (Coling 2010), Beijing, China, pp 439–446

He Z, Liu Q, Lin S (2008) Improving statistical machine translation using lexicalized rule selection. In: The 22nd International Conference on Computational Linguistics, Manchester, UK, pp 321–328

Hopkins M, May J (2011) Tuning as ranking. In: The 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, UK., pp 1352–1362

Huck M, Wuebker J, Rietig F, Ney H (2013) A phrase orientation model for hierarchical machine translation. In: The Eighth Workshop on Statistical Machine Translation, Sofia, Bulgaria, pp 452–463

Kazemi A, Toral A, Way A, Monadjemi A, Nematbakhsh M (2015) Dependency-based reordering model for constituent pairs in hierarchical smt. In: The 18th Annual Conference of the European Association for Machine Translation, Antalya, Turkey, pp 43–50

Koehn P (2004) Statistical significance tests for machine translation evaluation. In: The 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp 388–395

Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: The 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Edmonton, Canada, pp 48–54

Koehn P, Axelrod A, Birch A, Callison-Burch C, Osborne M, Talbot D, White M (2005) Edinburgh system description for the 2005 IWSLT speech translation evaluation. In: The International Workshop on Spoken Language Translation, Pittsburgh,USA, pp 68–75

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: Open source toolkit for statistical machine translation. In: The 45th Annual Meeting of the Association for Computational Linguistics :the Demo and Poster Sessions, Prague, Czech Republic, pp 177–180

Li P, Liu Y, Sun M, Izuha T, Zhang D (2014) A neural reordering model for phrase-based translation. In: The 25th International Conference on Compu-

tational Linguistics: Technical Papers, Dublin, Ireland, pp 1897–1907
Liu Q, He Z, Liu Y, Lin S (2008) Maximum entropy based rule selection model for syntax-based statistical machine translation. In: The 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, Hawaii, pp 89–97
Marton Y, Resnik P (2008) Soft syntactic constraints for hierarchical phrased-based translation. In: The 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Columbus, Ohio, pp 1003–1011
Nguyen T, Vogel S (2013) Integrating phrase-based reordering features into a chart-based decoder for machine translation. In: The 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Sofia, Bulgaria, pp 1587–1596
Ni Y, Saunders C, Szedmak S, Niranjan M (2009) Handling phrase reorderings for machine translation. In: The ACL-IJCNLP 2009 Conference Short Papers, Suntec, Singapore, pp 241–244
Och FJ (2003) Minimum error rate training in statistical machine translation. In: The 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, pp 160–167
Och FJ, Ney H (2002) Discriminative training and maximum entropy models for statistical machine translation. In: The 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pp 295–302
Och FJ, Ney H (2003) A systematic comparison of various statistical alignment models. Computational linguistics 29(1):19–51
Rumelhart D, Hinton G, Williams R (1986) Learning representations by back-propagating errors. Nature 323(6088) pp 533–536
Tromble R, Eisner J (2009) Learning linear ordering problems for better translation. In: The 2009 Conference on Empirical Methods in Natural Language Processing, Singapore, pp 1007–1016
Vaswani A, Zhao Y, Fossum V, Chiang D (2013) Decoding with large-scale neural language models improves translation. In: The 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, pp 1387–1392
Wang X, Xiong D, Zhang M (2015) Learning semantic representations for non-terminals in hierarchical phrase-based translation. In: The 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, pp 1391–1400
Zens R, Ney H (2006) Discriminative reordering models for statistical machine translation. In: The Workshop on Statistical Machine Translation, New York City, pp 55–63
zhang j, Utiyama M, Sumita E, Zhao H (2015) Learning word reorderings for hierarchical phrase-based statistical machine translation. In: The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Beijing, China, pp 542–548

Zhao H, Huang CN, Li M (2006) An improved Chinese word segmentation system with conditional random field. In: The Fifth SIGHAN Workshop on Chinese Language Processing, Sydney, Australia, pp 162–165