Learning Word Reorderings for Hierarchical Phrase-based Statistical Machine Translation

Jingyi Zhang[1,2], **Masao Utiyama**[1], **Eiichro Sumita**[1], **Hai Zhao**[3,4]
[1]National Institute of Information and Communications Technology,
3-5Hikaridai, Keihanna Science City, Kyoto 619-0289, Japan
[2]Graduate School of Information Science, Nara Institute of Science and Technology,
Takayama, Ikoma, Nara 630-0192, Japan
[3]Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
[4]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
jingyizhang/mutiyama/eiichiro.sumita@nict.go.jp
zhaohai@cs.sjtu.edu.cn

Abstract

Statistical models for reordering source words have been used to enhance the hierarchical phrase-based statistical machine translation system. Existing word reordering models learn the reordering for any two source words in a sentence or only for two continuous 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 distance less than a specific threshold are useful to improve translation quality. Compared with previous work, our method may more effectively and efficiently exploit helpful word reordering information.

1 Introduction

The hierarchical phrase-based model (Chiang, 2005) is capable of capturing rich translation knowledge with the synchronous context-free grammar. But selecting proper translation rules during decoding is a challenge as a huge number of hierarchical rules can be applied to one 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. A lot of 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.

Hayashi et al. (2010) demonstrated the effectiveness of using word reordering information within hierarchical phrase-based SMT by integrating Tromble and Eisner (2009)'s word reordering model into decoder as a feature, which estimates the probability of any two source words in a sentence being reordered during translating. Feng et al. (2013) proposed a word reordering model to learn reorderings only for continuous words, which reduced computation cost a lot compared with Tromble and Eisner (2009)'s model and still achieved significant reordering improvement over the baseline system.

In this paper, 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. We will demonstrate that the translation performance achieves consistent improvement as more sub-models for longer distance reorderings being integrated, but the improvement levels off quickly. That means sub-models for reordering distance longer than a given threshold do not improve translation quality significantly. Compared with previous models (Tromble and Eisner, 2009; Feng et al., 2013), our method makes full use of helpful word reordering information and also avoids unnecessary computation cost for long distance reorderings. Besides, our reordering model is learned by feed-forward neural network (FNN) for better performance and uses efficient caching strategy to further reduce time cost.

Phrase reordering models have also been integrated into hierarchical phrase-based SMT. Phrase reordering models were originally developed for phrase-based SMT (Koehn et al., 2005; Zens and Ney, 2006; Ni et al., 2009; Li et al., 2014) and

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 542–548,
Beijing, China, July 26-31, 2015.

could not be used in hierarchical phrase-based model directly. Nguyen and Vogel (2013) and Cao et al. (2014) proposed to integrate phrase-based reordering features into hierarchical phrase-based SMT. However, their work limited to learning the reordering of continuous phrases. For short phrases, in extreme cases, when phrase length is one, their model only learned reordering for continuous word pairs like Feng et al. (2013)'s work, while our model can be applied to word pairs with longer distances.

2 Our Approach

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 alignments between e_1^m and f_1^l, our model estimates the reordering probability of the source sentence as follows:

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

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 translating; N is the maximum distance for source word reordering, which is empirically determined by supposing that estimating reorderings longer than N does not improve translation performance any more.

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 uncontinuous target words, it can be hard to determine if a word pair is reversed or not. They solved this problem by only using one alignment from multiple alignments and ignoring the others. In contrast, our model handles all alignments as shown below.

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

$$\begin{aligned} &\Pr\left(f_1^l, e_1^m, A, i, j\right) \\ &= \prod_{u=1}^{\pi_i} \prod_{v=1}^{\pi_j} \Pr\left(o_{ijuv} | f_{i-3}, ..., f_{j+3}, e_{a_{iu}}, e_{a_{jv}}\right) \\ &where \\ &o_{ijuv} = \begin{cases} 0 & (a_{iu} \le a_{jv}) \\ 1 & (a_{iu} > a_{jv}) \end{cases} \end{aligned} \quad (2)$$

We train a series of sub-models,

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

Algorithm 1 Extract training instances.

Require: A pair of parallel sentence f_1^l and e_1^m with word alignments.
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}, ..., f_{j+3}, e_{a_iu}, e_{a_jv}, 0) is a negative instance for M_{j−i}
          else
            (f_{i−3}, ..., f_{j+3}, e_{a_iu}, e_{a_jv}, 1) is a positive instance for M_{j−i}
```

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

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). The input to M_n is a sequence of $n + 9$ words: $f_{i-3}, ..., f_{j+3}, e_{a_{iu}}, e_{a_{jv}}$. The input layer projects each word into a high dimensional vector using a matrix of input word embeddings. Two hidden layers can combine all input data[1]. The output layer has two neurons that give $\Pr\left(o_{ijuv} = 1 | f_{i-3}, ..., f_{j+3}, e_{a_{iu}}, e_{a_{jv}}\right)$ and $\Pr\left(o_{ijuv} = 0 | f_{i-3}, ..., f_{j+3}, e_{a_{iu}}, e_{a_{jv}}\right)$.

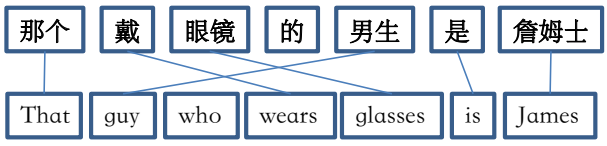

Figure 1: A Chinese-English sentence pair.

The backpropagation algorithm is used to train these reordering sub-models. 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

[1]If we choose the averaged perceptron algorithm to learn reordering task as used in (Hayashi et al., 2010), we need to artificially select n-gram features, which is not necessary for FNN.

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 training corpus, we replace it with a special symbol $<unk>$.

3 Integration into the Decoder

In the hierarchical phrase-based model, a translation rule r is like:

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

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

Each rule has several features and the feature weights are tuned by the minimum error rate training (MERT) algorithm (Och, 2003). 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. The score of this feature is calculated during decoding. Note that these scores are correspondingly calculated for different sub-models M_n and the sub-model weights are tuned separately.

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

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

Then

$$\begin{aligned} & score_n\,(r) \\ = & \sum_{\langle i,j\rangle \in S - \bigcup_{k=1}^{K} S_k \wedge j-i=n} \log\left(\Pr\left(f_1^l, e_1^m, A, i, j\right)\right) \\ & where \\ & S : \{\langle i,j\rangle \,|\varphi \leq i < j \leq \vartheta\} \\ & S_k : \{\langle i,j\rangle \,|\varphi_k \leq i < j \leq \vartheta_k\} \end{aligned}$$

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

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

One concern in using target features is the computational efficiency, because reordering probabilities have to be calculated during decoding. So we cache probabilities to reduce the expensive neural network computation in experiments.

4 Experiments

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. The detailed statistics for training, development and test sets are given in Table 1.

			SOURCE	TARGET
CE	TRAINING	#Sents	954K	
		#Words	37.2M	40.4M
		#Vocab	288K	504K
	DEV	#Sents	2K	
	TEST	#Sents	2K	
JE	TRAINING	#Sents	3.14M	
		#Words	118M	104M
		#Vocab	150K	273K
	DEV	#Sents	2K	
	TEST	#Sents	2K	

Table 1: Data sets.

In NTCIR-9, the development and test sets were both provided for 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 JE task. The word segmentation was done by BaseSeg (Zhao et al., 2006; Zhao and Kit, 2008; Zhao et al., 2010; Zhao and Kit, 2011; Zhao et al., 2013) for Chinese and Mecab[2] for Japanese.

To learn neural reordering models, the training and development sets were put together 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 respectively for learning neural reordering models. Neural reordering models were trained by the toolkit NPLM (Vaswani et al., 2013). For CE task, training instances extracted from all the 1M sentence pairs were used to train neural reordering models. For JE task, training instances were from 1M sentence pairs that were randomly selected from all the 3.14M sentence pairs.

We also implemented Hayashi et al. (2010)'s model for comparison. The training instances for their model were extracted from the same sentence pairs as ours.

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

	Base	Hayashi model	M_1^1	M_1^2	M_1^3	M_1^4
CE	32.95	34.25	34.78	35.75	35.97	36.05
JE	30.13	30.70	31.35	32.07	32.40	32.60

(a) BLEU scores

CE	*Base*	Hayashi model	M_1^1	M_1^2	M_1^3
Hayashi model	$\gg$				
M_1^1	$\gg$	$\gg$			
M_1^2	$\gg$	$\gg$	$\gg$		
M_1^3	$\gg$	$\gg$	$\gg$	$>$	
M_1^4	$\gg$	$\gg$	$\gg$	$>$	$-$
JE	*Base*	Hayashi model	M_1^1	M_1^2	M_1^3
Hayashi model	$\gg$				
M_1^1	$\gg$	$\gg$			
M_1^2	$\gg$	$\gg$	$\gg$		
M_1^3	$\gg$	$\gg$	$\gg$	$\gg$	
M_1^4	$\gg$	$\gg$	$\gg$	$\gg$	$-$

(b) Significance test results using bootstrap sampling (Koehn, 2004) w.r.t. BLEU scores. The symbol $\gg$ represents a significant difference at the $p < 0.01$ level; $>$ represents a significant difference at the $p < 0.05$ level; $-$ means not significantly different at $p = 0.05$.

Table 2: Translation results.

For each translation task, the recent version of the Moses hierarchical phrase-based decoder (Koehn et al., 2007) with the training scripts was used as the baseline system *Base*. We used the default parameters for Moses. A 5-gram language model was trained on the target side of the training corpus by IRST LM Toolkit[3] with the improved Kneser-Ney smoothing.

We integrated our reordering models into *Base*. Table 2 gives detailed translation results. "Hayashi model" represents the method of (Hayashi et al., 2010). "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.

As shown in Table 2, integrating only M_1, which predicts reordering for two continuous source words, has already given BLEU improvement 1.8% and 1.2% over baseline on CE and JE, respectively. As more sub-models for longer distance reordering being integrated, the translation performance improved consistently, though the improvement leveled off quickly. For CE and JE tasks, M_n with $n \geq 3$ and $n \geq 4$, respectively, cannot give further performance improvement at any significant level.

Why did the improvement level off quickly?

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

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

(a) Our model

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

(b) Hayashi model

Table 3: Classification accuracy (%).

In other words, why do long distance reordering models have a much less leverage over translation performance than short ones?

First, the prediction accuracy decreases as the reordering distance increasing. Table 3a gives classification accuracies on the validation data for each sub-model. The reason for accuracy decreasing is that the input size of sub-model grows as reordering distance increasing. Namely, long distance reordering needs to consider more complicated context.

Second, we attribute the influence decrease of the longer reordering models to the redundancy of the predictions among different reordering models. For example, in Figure 1, when 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 further reordering model prediction. That means, sometimes, a long distance word reordering can be determined by a series of shorter word reordering pairs.

But still, some predictions for longer reordering 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 translation performance improves as more sub-models being integrated.

As shown in Table 2, with 4 sub-models being integrated, our model improved baseline system significantly and also outperformed Hayashi model clearly. It is easy to understand, since our model was trained by feed-forward neural network on a high dimensional space and incorporated rich context information, while Hayashi model used the averaged perceptron algorithm and simple features. Table 3b shows the prediction accuracies

of Hayashi model. Note that Hayashi model predicts reorderings for all word pairs, but only prediction accuracies for word pairs with distance 4 or less are shown. Compared with Table 3a, the prediction accuracy of our model is much higher than Hayashi model. Actually, FNN is not suitable for Hayashi model since the computation cost for Hayashi model is quite expensive. Using FNN to reorder all word pairs could cost nearly one minute to translate one sentence according to our experiments, while integrating 4 sub-models only cost 10 seconds[4].

Compared with Hayashi model, our model not only speeds up decoding time but also reduces the training time. Training for Hayashi model is much slower since word pairs with all different distances are used as training data. By using separate sub-models, we can train each sub-model one by one and stop when translation performance cannot be improved any more. However, despite of efficiency, one unified model will theoretically have better performance than separate sub-models since separate sub-models do not share training instances and the unified model will suffer less from data sparsity. So, we did some extra experiments and trained a neural network which had the same structure as M_4 to learn reorderings for all word pairs with distance 4 or less, instead of using 4 separate neural networks. A specific word $null$ was used since word pairs with distance 1,2,3 do not have enough inputs for M_4. The significance test results showed that translation performance had no significant difference between one unified model and multiple sub-models. This is because the training corpus for our model is quite large, so separate training sets are sufficient for each sub-model to learn the reorderings well. Besides, using neural networks to learn these sub-models on a continuous space can relieve the data sparsity problem to some extent.

Note that if we only integrate M_4 into Base, the translation quality of Base can be improved in our preliminary experiments. But 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 will not be needed due to the redundancy of the predictions among different reordering models.

[4]Note that cache was used in all our experiments to reduce the expensive neural network computation cost and turned out to be very useful. Without caching, integrating 4 sub-models could cost nearly 7 minutes to translate a sentence.

5 Conclusion

In this paper, we propose to enhance hierarchical phrase-based SMT by training a series of separate sub-models to learn reorderings for word pairs with distances less than a specific threshold, based on the experimental fact that longer distance reordering models are not quite helpful for translation quality. Compared with Hayashi et al. (2010)'s work, our model is much more efficient and keeps all helpful word reordering information. Besides, our reordering model is learned by feed-forward neural network and incorporates rich context information for better performance. On both Chinese-to-English and Japanese-to-English translation tasks, the proposed model outperforms the previous model significantly.

Acknowledgments

Masao Utiyama and Hai Zhao are corresponding authors. This work was done when the first author was a master's student at Shanghai Jiao Tong University. Hai Zhao was supported by the National Natural Science Foundation of China under Grants 60903119, 61170114 and 61272248, the National Basic Research Program of China under Grant 2013CB329401, the Science and Technology Commission of Shanghai Municipality under Grant 13511500200, the European Union Seventh Framework Program under Grant 247619, the Cai Yuanpei Program (CSC fund 201304490199, 201304490171), and the art and science interdiscipline funds of Shanghai Jiao Tong University under Grant 14X190040031(14JCRZ04).

References

Hailong Cao, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2014. A lexicalized reordering model for hierarchical phrase-based translation. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1144–1153.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270.

Lei Cui, Dongdong Zhang, Mu Li, Ming Zhou, and Tiejun Zhao. 2010. A joint rule selection model

for hierarchical phrase-based translation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 6–11.

Minwei Feng, Jan-Thorsten Peter, and Hermann Ney. 2013. Advancements in reordering models for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 322–332.

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 The 9th NII Test Collection for IR Systems Workshop Meeting*, pages 559–578.

Katsuhiko Hayashi, Hajime Tsukada, Katsuhito Sudoh, Kevin Duh, and Seiichi Yamamoto. 2010. Hierarchical phrase-based machine translation with word-based reordering model. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 439–446.

Zhongjun He, Qun Liu, and Shouxun Lin. 2008. Improving statistical machine translation using lexicalized rule selection. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 321–328.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54.

Philipp Koehn, Amittai Axelrod, Alexandra Birch, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *The International Workshop on Spoken Language Translation*, pages 68–75.

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.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395.

Peng Li, Yang Liu, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. 2014. A neural reordering model for phrase-based translation. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1897–1907.

Qun Liu, Zhongjun He, Yang Liu, and Shouxun Lin. 2008. Maximum entropy based rule selection model for syntax-based statistical machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 89–97.

ThuyLinh Nguyen and Stephan Vogel. 2013. Integrating phrase-based reordering features into a chart-based decoder for machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1587–1596.

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.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1007–1016.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392.

Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 55–63.

Hai Zhao and Chunyu Kit. 2008. Exploiting unlabeled text with different unsupervised segmentation criteria for chinese word segmentation. *Research in Computing Science*, 33:93–104.

Hai Zhao and Chunyu Kit. 2011. Integrating unsupervised and supervised word segmentation: The role of goodness measures. *Information Sciences*, 181(1):163–183.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for chinese word segmentation. *ACM Transactions on Asian Language Information Processing (TALIP)*, 9(2):5.

Hai Zhao, Masao Utiyama, Eiichiro Sumita, and Bao-Liang Lu. 2013. An empirical study on word segmentation for chinese machine translation. In *Computational Linguistics and Intelligent Text Processing*, pages 248–263.