A Tagging-style Reordering Model for Phrase-based SMT

Minwei FENG Hermann NEY
Human Language Technology and Pattern Recognition Group,
Computer Science Department,
RWTH Aachen University,
Aachen, Germany
feng@cs.rwth-aachen.de, ney@cs.rwth-aachen.de

ABSTRACT
For current statistical machine translation system, reordering is still a major problem for language pairs like Chinese-English, where the source and target language have significant word order differences. In this paper we propose a novel tagging-style reordering model. Our model converts the reordering problem into a sequence labeling problem, i.e. a tagging task. For the given source sentence, we assign each source token a label which contains the reordering information for that token. We also design an unaligned word tag so that the unaligned word phenomenon is automatically covered in the proposed model. Our reordering model is conditioned on the whole source sentence. Hence it is able to catch long dependencies in the source sentence. The decoder makes use of the tagging information as soft constraints so that in the test phase (during translation) our model is very efficient. The model training on large scale tasks requests notably amounts of computational resources. We carried out experiments on five Chinese-English NIST tasks trained with BOLT data. Results show that our model improves the baseline system by 0.98 BLEU 1.21 TER on average.

KEYWORDS: statistical machine translation, reordering, conditional random fields.

Proceedings of the Workshop on Reordering for Statistical Machine Translation, pages 17–26,
COLING 2012, Mumbai, December 2012.

1 Introduction

The systematic word order difference between two languages pose a challenge for current statistical machine translation (SMT) systems. The system has to decide in which order to translate the given source words. This problem is known as the reordering problem. As shown in (Knight, 1999), if arbitrary reordering is allowed, the search problem is NP-hard.

In this paper, we propose a novel tagging style reordering model. Our model converts the reordering problem into a sequence labeling problem, i.e. a tagging task. For a given source sentence, we assign each source token a label which contains the reordering information for that token. We also design an unaligned word tag so that the unaligned word phenomenon is automatically covered in the proposed model. Our model is conditioned on the whole source sentence. Hence it is able to capture the long dependencies in the source sentence. We choose the conditional random fields (CRFs) approach for the tagging model. Although utilizing CRFs on large scale task requests a notable amount of computational resources, the decoder makes use of the tagging information as soft constraints. Therefore, the training procedure of our model is computationally expensive while in the test phase (during translation) our model is very efficient.

The remainder of this paper is organized as follows: Section 2 reviews the related work for solving the reordering problem. Section 3 introduces the basement of this research: the principle of statistical machine translation. Section 4 describes the proposed model. Section 5 provides the experimental configuration and results. Conclusion will be given in Section 6.

2 Related Work

Many ideas have been proposed to address the reordering problem. Within the phrase-based SMT framework there are mainly three stages where improved reordering could be integrated:

1. Reorder the source sentence. So that the word order of source and target sentences is similar. Usually it is done as the preprocessing step for both training data and test data.
2. In the decoder, add models in the log-linear framework or constraints in the decoder to reward good reordering options or penalize bad ones.
3. In the reranking framework.

For the first point, (Wang et al., 2007) used manually designed rules to reorder parse trees of the source sentences as a preprocessing step. Based on shallow syntax, (Zhang et al., 2007) used rules to reorder the source sentences on the chunk level and provide a source-reordering lattice instead of a single reordered source sentence as input to the SMT system. Designing rules to reorder the source sentence is conceptually clear and usually easy to implement. In this way, syntax information can be incorporated into phrase-based SMT systems. However, one disadvantage is that the reliability of the rules is often language pair dependent.

In the second category, researchers try to inform the decoder on what a good reordering is or what a suitable decoding sequence is. (Zens and Ney, 2006) used a discriminative reordering model to predict the orientation of the next phrase given the previous phrase. (Mariño et al., 2006) presents a translation model that constitutes a language model of a sort of "bilanguage" composed of bilingual units. From the reordering point of view, the idea is that the correct reordering is to find the suitable order of translation units. (Cherry, 2008) puts the syntactic cohesion as a soft constraint in the decoder to guide the decoding process to choose those translations that do not violate the syntactic structure of the source sentence. Adding new

features in the log-linear framework has the advantage that the new feature has access to the whole search space. Another advantage of methods in this category is that we let the decoder decide the weights of features, so that even if one model gives wrong estimation sometimes, it can still be corrected by other models. Our work in this paper belongs to this category.

In the reranking step, the system has the last opportunity to choose a good translation. (Och et al., 2004) describe the use of syntactic features in the rescoring step. They report the most useful feature is IBM Model 1 score. The syntactic features contribute very small gains. Another disadvantage of carrying out reordering in reranking is the representativeness of the N-best list is often a question mark.

3 Translation System Overview

In this section, we are going to describe the phrase-based SMT system we used for the experiments. In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \dots f_j \dots f_J$. The objective is to translate the source into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. The strategy is among all possible target language sentences, we will choose the one with the highest probability:

$$\hat{e}_i^{\hat{I}} = \arg\max_{I, e_1^I} \{Pr(e_1^I | f_1^J)\} \tag{1}$$

We model $Pr(e_1^I | f_1^J)$ directly using a log-linear combination of several models (Och and Ney, 2002):

$$Pr(e_1^I | f_1^J) = \frac{\exp\Big(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\Big)}{\sum_{I', e'^{I'}_1} \exp\Big(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J)\Big)} \tag{2}$$

The denominator is to make the $Pr(e_1^I | f_1^J)$ to be a probability distribution and it depends only on the source sentence f_1^J. For search, the decision rule is simply:

$$\hat{e}_i^{\hat{I}} = \arg\max_{I, e_1^I} \Big\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \Big\} \tag{3}$$

The model scaling factors λ_1^M are trained with Minimum Error Rate Training (MERT).

In this paper, the phrase-based machine translation system is utilized (Och et al., 1999; Zens et al., 2002; Koehn et al., 2003). The translation process consists in segmenting of the source sentence according to the phrase table which is built from the word alignment. The translation of each of these segments consists just in extracting the target side from the phrase pair. With the corresponding target side, the final translation is the composition of these translated segments. In this last step, reordering is allowed.

4 Tagging-style Reordering Model

In this section, we describe the proposed model. First we will describe the training process. Then we explain how to use the model in the decoder.

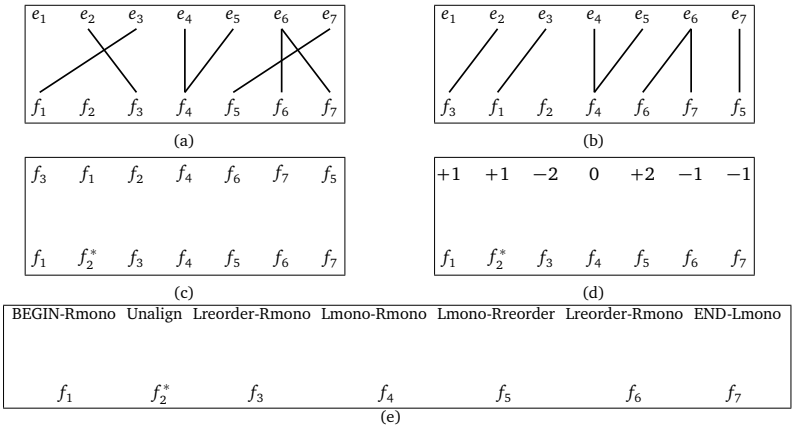

Figure 1: Modeling process illustration.

4.1 Modeling

Figure 1 demonstrates the modeling steps. The first step is word alignment training. Figure 1(a) is an example after GIZA++ training. If we regard this alignment as a translation result, i.e. given the source sentence f_1^7, the system translates it into the target sentence e_1^7. The alignment link set $\{a_1 = 3, a_3 = 2, a_4 = 4, a_4 = 5, a_5 = 7, a_6 = 6, a_7 = 6\}$ reveals the decoding process, i.e. the alignment implies the order in which the source words should be translated, e.g. the first generated target word e_1 has no alignment, we can regard it as a translation from a NULL source word; then the second generated target word e_2 is translated from f_3. We reorder the source side of the alignment to get Figure 1(b). Figure 1(b) implies the source sentence decoding sequence information, which is depicted in Figure 1(c). Using this example we describe the strategies we used for special cases in the transformation from Figure 1(b) to Figure 1(c):

- ignore the unaligned target word, e.g. e_1
- the unaligned source word should follow its preceding word, the unaligned feature is kept with a $*$ symbol, e.g. f_2^* is after f_1
- when one source word is aligned to multiple target words, only keep the alignment that links the source word to the first target word, e.g. f_4 is linked to e_5 and e_6, only $f_4 - e_5$ is kept. In other words, every source word appears only once in the source decoding sequence.
- when multiple source words are aligned to one target word, put together the source words according to their original relative positions, e.g. e_6 is linked to f_6 and f_7. So in the decoding sequence, f_6 is before f_7.

Now Figure 1(c) shows the original source sentence and its decoding sequence. By using the strategies above, it is guaranteed that the source sentence and its decoding sequence has the exactly same length. Hence the relation can be modeled by a function $F(f)$ which assigns a value for each of the source word f. Figure 1(d) manifests this function. The positive function

values mean that compared to the original position in the source sentence, its position in the decoding sequence should move right, and vice versa. If the function value is 0, the word's position in original source sentence and its decoding sequence is same. For example, f_1 is the first word in the source sentence but it is the second word in the decoding sequence. So its function value is +1 (move right one position).

Now Figure 1(d) converts the reordering problem into a sequence labeling or tagging problem. To move the computational cost to a reasonable level, we do a final simplification step in Figure 1(e). Suppose the longest sentence length is 100, then according to Figure 1(d), there are 200 tags (from -99 to +99 plus the unalign tag). As we will see later, this number is too large for our task. We instead design nine tags. For a source word f_j in one source sentence f_1^J, the tag of f_j will be one of the following:

BEGIN-Rmono $j = 1$ and f_{j+1} is translated *after* f_j (Rmono for right monotonic)

BEGIN-Rreorder $j = 1$ and f_{j+1} is translated *before* f_j (Rreorder for right reordered)

END-Lmono $j = J$ and f_{j-1} translated *before* f_j (Lmono for left monotonic)

END-Lreorder $j = J$ and f_{j-1} translated *after* f_j (Lreorder for left reordered)

Lmono-Rmono $1 < j < J$ and f_{j-1} translated *before* f_j and f_j translated *before* f_{j+1}

Lmono-Rreorder $1 < j < J$ and f_{j-1} translated *before* f_j and f_j translated *after* f_{j+1}

Lreorder-Rmono $1 < j < J$ and f_{j-1} translated *after* f_j and f_j translated *before* f_{j+1}

Lreorder-Rreorder $1 < j < J$ and f_{j-1} translated *after* f_j and f_j translated *after* f_{j+1}

Unalign f_j is an unaligned source word

Up to this point, we have converted the reordering problem into a tagging problem with nine tags. The transformation in Figure 1 is conducted for all the sentence pairs in the bilingual training corpus. After that, we have built an "annotated" corpus for the training. For this supervised structure learning task, we choose the approach conditional random fields (CRFs) (Lafferty et al., 2001; Sutton and Mccallum, 2006; Lavergne et al., 2010). More specifically, we adopt the linear-chain CRFs. However, even for the simple linear-chain CRFs, the complexity of learning and inference grows quadratically with respect to the number of output labels and the amount of structural features which are with regard to adjacent pairs of labels. Hence, to make the computational cost as low as possible, two measures have been taken. Firstly, as described above we reduce the number of tags to nine. Secondly, we add source sentence part-of-speech (POS) tags to the input. For features with window size one to three, both source words and its POS tags are used. For features with window size four and five, only POS tags are used.

4.2 Decoding

Once the CRFs training is finished, we make inference on develop and test corpora. After that we get the labels of the source sentences that need to be translated. In the decoder, we add a new model which checks the labeling consistence when scoring an extended state. During the search, a sentence pair (f_1^J, e_1^I) will be formally splitted into a segmentation S_1^K which consists of K phrase pairs. Each $s_k = (i_k; b_k, j_k)$ is a triple consisting of the last position i_k of the kth target phrase $\tilde{e}_k$. The start and end position of the kth source phrase $\tilde{f}_k$ are b_k and j_k. Suppose the search state is now extended with a new phrase pair $(\tilde{f}_k, \tilde{e}_k)$:

$$\tilde{f}_k := f_{b_k} \dots f_{j_k} \tag{4}$$

$$\tilde{e}_k := e_{i_{k-1}+1} \dots e_{i_k} \tag{5}$$

We have access to the old coverage vector, from which we know if the left neighboring source word f_{b_k-1} and the right neighboring source word f_{j_k+1} of the new phrase have been translated. We also have the word alignment within the new phrase pair, which is stored during the phrase extraction process. Based on the old coverage vector and alignment, we can repeat the transformation in Figure 1 to calculate the labels for the new phrase. The added model will then check the consistence between the calculated labels and the labels predicted by the CRFs. The number of source words that have inconsistent labels is regarded as penalty and then the penalty is added as a new feature into the log-linear framework.

5 Experiments

In this section, we describe the baseline setup, the CRFs training results and translation experimental results.

5.1 Experimental Setup

Our baseline is a phrase-based decoder, which includes the following models: an n-gram target-side language model (LM), a phrase translation model and a word-based lexicon model. The latter two models are used for both directions: $p(f|e)$ and $p(e|f)$. Additionally we use phrase count features, word and phrase penalty. The reordering model for the baseline system is the distance-based jump model which uses linear distance. This model does not have hard limit. We list the important information regarding the experimental setup below. All those conditions have been kept same in this work.

- lowercased training data (Table 1) from the BOLT task
  alignment trained with GIZA++
- development corpus: NIST06 test corpora: NIST02 03 04 05 and 08
- 5-gram LM (1 694 412 027 running words) trained by SRILM toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing
  training data: target side of bilingual data.
- BLEU (Papineni et al., 2001) and TER (Snover et al., 2005) reported
  all scores calculated in lowercase way.
- Wapiti toolkit (Lavergne et al., 2010) used for CRFs

	Chinese	**English**
Sentences	5 384 856	
Running Words	115 172 748	129 820 318
Vocabulary	1 125 437	739 251

Table 1: training data statistics

5.2 CRFs Training Results

Table 1 contains the data statistics used for translation model and LM. For the reordering model, we take two further filtering steps. Firstly, we delete the sentence pairs if the source sentence length is one. When the source sentence has only one word, the translation will be always monotonic and the reordering model does not need to learn this. Secondly, we delete the sentence pairs if the source sentence contains more than three contiguous unaligned words.

When this happens, the sentence pair is usually low quality hence not suitable for learning. The main purpose of the two filtering steps is to further lay down the computational burden. We then divide the corpus into three parts: train, validation and test. The source side data statistics for CRFs training is given in Table 2 (target side has only 9 labels). The toolkit Wapiti

	train	**validation**	**test**
Sentences	2 973 519	400 000	400 000
Running Words	62 263 295	8 370 361	8 382 086
Vocabulary	454 951	149 686	150 007

Table 2: reordering model training data statistics

(Lavergne et al., 2010) is used in this paper. We choose the classical optimization algorithm limited memory BFGS (L-BFGS) (Liu and Nocedal, 1989). For regularization, Wapiti uses both the ℓ^1 and ℓ^2 penalty terms, yielding the elastic-net penalty of the form

$$\rho_1 \cdot \parallel \theta \parallel_1 + \frac{\rho_2}{2} \cdot \parallel \theta \parallel_2^2 \tag{6}$$

In this work, we use as many features as possible because ℓ^1 penalty $\rho_1 \parallel \theta \parallel_1$ is able to yield sparse parameter vectors, i.e. using a ℓ^1 penalty term implicitly performs the feature selection. On a cluster with two AMD Opteron(tm) Processor 6176 (total 24 cores), the training time is about 16 hours, peak memory is around 120G. Several experiments have been done to find the suitable hyperparameters ρ_1 and ρ_2. We choose the model with lowest error rate on the validation corpus for the translation experiments. The error rate of the chosen model on test corpus is 25.75% for token error rate and 69.39% for sequence error rate. The error rate values are much higher than what we usually see in part-of-speech tagging task. The main reason is that the "annotated" corpus is converted from word alignment which contains a lot of errors. However, as we will show later, the learned CRFs model helps to improve the translation quality. The feature template we set initially will generate 722 999 637 features. After training 36 902 363 features are kept.

5.3 Translation Results

Results are summarized in Table 3. Automatic measure BLEU and TER scores are provided. Also we report significance testing results on both BLEU and TER. We perform bootstrap resampling with bounds estimation as described in (Koehn, 2004). We use the 95% confidence threshold (denoted by ‡ in the table) to draw significance conclusions. Besides the five test corpora, we add a column **avg.** to show the average improvements. We also add a column **Index** for score reference convenience.

From Table 3 we see that our proposed reordering model is able to improve the baseline by 0.98 BLEU and 1.21 TER on average. The largest BLEU improvement 1.11 is from NIST04 and the largest TER improvement 1.57 is from NIST03. For line 2 and 6, the significance test was done and most scores are better than their corresponding baseline values with more than 95% confidence (scores marked with ‡).

We also compare our model with the widely used Moses Lexicalized Reordering Model (Koehn et al., 2007). Line 3 and 7 are the results. Results show that for BLEU both model achieve almost same results (average improvement 0.98 BLEU versus 0.99 BLEU). For TER, our tagging-style reordering model is 0.25 points better (average improvement 1.21 TER versus 0.96 TER). When

Systems	NIST02	NIST03	NIST04	NIST05	NIST08	avg.	Index
			BLEU scores				
baseline	33.60	34.29	35.73	32.15	26.34	-	1
baseline+CRFs	34.53	35.19	36.56‡	33.30‡	27.41‡	0.98	2
baseline+Moses	34.87	34.90	36.40	33.43	27.45	0.99	3
baseline+CRFs+Moses	35.41	35.63	37.24	33.98	27.47	1.52	4
			TER scores				
baseline	61.36	60.48	59.12	60.94	65.17	-	5
baseline+CRFs	60.14‡	58.91‡	57.91‡	59.77‡	64.30‡	1.21	6
baseline+Moses	60.07	59.08	58.42	59.74	64.50	0.96	7
baseline+CRFs+Moses	59.33	58.48	57.44	59.12	64.43	1.65	8

Table 3: Experimental results

the tagging-style reordering model is used together with the lexicalized reordering model, further improvements have been observed. Results are presented in line 4 and 8. The two models improve the baseline by 1.52 BLEU and 1.65 TER on average.

6 Conclusion

In this paper, a novel tagging style reordering model has been proposed. By our modeling method, the reordering problem is converted into a sequence labeling problem so that the whole source sentence is taken into consideration for the reordering decisions. By adding an unaligned word tag, the unaligned word phenomenon is automatically covered in the proposed model. Although the training phase of our model is computationally expensive, its usage for decoding is quite simple. In practice, this algorithm does not significantly increase memory or computation requirements during decoding.

We choose CRFs to accomplish the relational learning task. The learning task needs 120G memory and lasts for 16 hours. Both ℓ^1 and ℓ^2 penalty are used in regularization. Hence the feature selection is automatically conducted. For test corpus, the token error rate is 25.75% and sequence error rate is 69.39%.

We utilize the CRFs model as soft constraints in the decoder. Experimental results show that our model is stable and improves the baseline system by 0.98 BLEU and 1.21 TER. Most of the scores are better than their corresponding baseline values with more than 95% confidence.

The comparison with Moses Lexicalized Reordering Model has been done. Results show that our model achieve the same performance with the lexicalized reordering model on BLEU measure. For TER the tagging-style reordering model is 0.25 points better. By applying the tagging-style reordering model and lexicalized reordering model together, further improvements can be achieved. The lexicalized reordering model only captures the dependency between neighboring phrases while our model uses the whole source sentence information.

7 Acknowledgements

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. 4911028154.0. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Defense Advanced Research Projects Agency (DARPA).

References

Cherry, C. (2008). Cohesive phrase-based decoding for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 72–80, Columbus, Ohio.

Knight, K. (1999). Decoding complexity in word-replacement translation models. *Comput. Linguist.*, 25(4):607–615.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395, Barcelona, Spain.

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., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (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*, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lavergne, T., Cappé, O., and Yvon, F. (2010). Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics.

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

Mariño, J. B., Banchs, R. E., Crego, J. M., de Gispert, A., Lambert, P., Fonollosa, J. A. R., and Costa-Jussà, M. R. (2006). *N*-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., and Radev, D. (2004). A smorgasbord of features for statistical machine translation. In *Proceedings of NAACL-HLT-04*, pages 161–168, Boston, Massachusetts, USA.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL-02*, pages 295–302, Philadelphia, Pennsylvania, USA.

Och, F. J., Tillmann, C., and Ney, H. (1999). Improved alignment models for statistical machine translation. In *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP99)*, pages 20–28, University of Maryland, College Park, MD, USA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2001). Bleu: a method for automatic evaluation of machine translation. (RC22176 (W0109-022)).

Snover, M., Dorr, B., Schwartz, R., Makhoul, J., Micciulla, L., and Weischedel, R. (2005). A study of translation error rate with targeted human annotation. Technical Report LAMP-TR-126, CS-TR-4755, UMIACS-TR-2005-58, University of Maryland, College Park, MD.

Stolcke, A. (2002). Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP-02*, pages 901–904, Denver, Colorado,USA.

Sutton, C. and Mccallum, A. (2006). *Introduction to Conditional Random Fields for Relational Learning*. MIT Press.

Wang, C., Collins, M., and Koehn, P. (2007). Chinese syntactic reordering for statistical machine translation. In *Proceedings of the EMNLP/CoNLL-07*, pages 737–745, Prague, Czech Republic.

Zens, R. and Ney, H. (2006). Discriminative reordering models for statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation at HLT-NAACL-06*, pages 55–63, New York City, NY.

Zens, R., Och, F. J., and Ney, H. (2002). Phrase-based statistical machine translation. In *German Conference on Artificial Intelligence*, pages 18–32. Springer Verlag.

Zhang, Y., Zens, R., and Ney, H. (2007). Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In *Proceedings of the NAACL-HLT-07/AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 1–8, Morristown, NJ, USA.