Chunk-Level Reordering of Source Language Sentences with Automatically Learned Rules for Statistical Machine Translation

Yuqi Zhang and Richard Zens and Hermann Ney

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6 – Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{yzhang,zens,ney}@cs.rwth-aachen.de

Abstract

In this paper, we describe a source-side reordering method based on syntactic chunks for phrase-based statistical machine translation. First, we shallow parse the source language sentences. Then, reordering rules are automatically learned from source-side chunks and word alignments. During translation, the rules are used to generate a reordering lattice for each sentence. Experimental results are reported for a Chinese-to-English task, showing an improvement of 0.5%–1.8% BLEU score absolute on various test sets and better computational efficiency than reordering during decoding. The experiments also show that the reordering at the chunk-level performs better than at the POS-level.

1 Introduction

In machine translation, reordering is one of the major problems, since different languages have different word order requirements. Many reordering constraints have been used for word reorderings, such as ITG constraints (Wu, 1996), IBM constraints (Berger et al., 1996) and local constraints (Kanthak et al., 2005). These approaches do not make use of any linguistic knowledge.

Several methods have been proposed to use syntactic information to handle the reordering problem, e.g. (Wu, 1997; Yamada and Knight, 2001; Gildea, 2003; Melamed, 2004; Graehl and Knight, 2004; Galley et al., 2006). One approach makes use of bitext grammars to parse both the source and target languages. Another approach makes use of syntactic information only in the target language. Note that these models have radically different structures and parameterizations than phrase-based models for SMT.

Another kind of approaches is to use syntactic information in rescoring methods. (Koehn and Knight, 2003) apply a reranking approach to the sub-task of noun-phrase translation. (Och et al., 2004) and (Shen et al., 2004) describe the use of syntactic features in reranking the output of a full translation system, but the syntactic features give very small gains.

In this paper, we present a strategy to reorder a source sentence using rules based on syntactic chunks. It is possible to integrate reordering rules directly into the search process, but here, we consider a more modular approach: easy to exchange reordering strategy. To avoid hard decisions before SMT, we generate a source-reordering lattice instead of a single reordered source sentence as input to the SMT system. Then, the decoder uses the reordered source language model as an additional feature function. A language model trained on the reordered source-side chunks gives a score for each path in the lattice. The novel ideas in this paper are:

- reordering of the source sentence at the chunk level,
- representing linguistic chunks-reorderings in a lattice.

The rest of this paper is organized as follows. Section 2 presents a review of related work. In Sections 3, we review the phrase-based translation system used in this work and propose the framework of the new reordering method. In Section 4, we introduce the details of the reordering rules, how they are defined and how to extract them. In Section 5, we explain how to apply the rules and how to generate reordering lattice. In Section 6, we present some results that show that the chunk-level source reordering is helpful for phrase-based statistical machine translation. Finally, we conclude this paper and discuss future work in Section 7.

2 Related Work

Beside the reordering methods during decoding, an alternative approach is to reorder the input source sentence to match the word order of the target sentence.

Some reordering methods are carried out on syntactic source trees. (Collins et al., 2005) describe a method for reordering German for German-to-English translation, where six transformations are applied to the surface string of the parsed source sentence. (Xia and McCord, 2004) propose an approach for translation from French-to-English. This approach automatically extracts rewrite patterns by parsing the source and target sides of the training corpus. These rewrite patterns can be applied to any input source sentence so that the rewritten source and target sentences have similar word order. Both methods need a parser to generate trees of source sentences and are applied only as a preprocessing step.

Another kind of source reordering methods besides full parsing is based on Part-Of-Speech (POS) tags or word classes. (Costa-jussà and Fonollosa, 2006) view the source reordering as a translation task that translate the source language into a reordered source language. Then, the reordered source sentence is taken as the single input to the standard SMT system.

(Chen et al., 2006) automatically extract rules from word alignments. These rules are defined at the POS level and the scores of matching rules are used as additional feature functions during rescoring. (Crego and Mariño, 2006) integrate source-side reordering into SMT decoding. They automatically learn rewrite patterns from word alignment and represent the patterns with POS tags. To our knowledge no work is reported on the reordering with shallow parsing.

Decoding lattices were already used in (Zens et al., 2002; Kanthak et al., 2005). Those approaches used linguistically uninformed word-level reorderings.

3 System Overview

In this section, we will describe the phrase-based SMT system which we use for the experiments. Then, we will give an outline of the extentions with the chunk-level source reordering model.

3.1 The Baseline Phrase-based SMT System

In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \ldots f_j \ldots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \ldots e_i \ldots e_I$. Among all possible target language sentences, we will choose the sentence with the highest probability:

$$\hat{e}_1^{\hat{I}} = \operatorname*{argmax}_{I,e_1^I} \left\{ Pr(e_1^I|f_1^J) \right\} \quad (1)$$

$$= \operatorname*{argmax}_{I,e_1^I} \left\{ Pr(e_1^I) \cdot Pr(f_1^J|e_1^I) \right\} \quad (2)$$

This decomposition into two knowledge sources is known as the source-channel approach to statistical machine translation (Brown et al., 1990). It allows an independent modeling of the target language model $Pr(e_1^I)$ and the translation model $Pr(f_1^J|e_1^I)$. The target language model describes the well-formedness of the target language sentence. The translation model links the source language sentence to the target language sentence. The argmax operation denotes the search problem, i.e., the generation of the output sentence in the target language.

A generalization of the classical source-channel approach is the direct modeling of the posterior probability $Pr(e_1^I|f_1^J)$. Using a log-linear model

(Och and Ney, 2002), we obtain:

$$Pr(e_1^I|f_1^J) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum\limits_{I', e'^{I'}_1} \exp\left(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J)\right)} \tag{3}$$

The denominator represents a normalization factor that depends only on the source sentence f_1^J. Therefore, we can omit it during the search process. As a decision rule, we obtain:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \tag{4}$$

The log-linear model has the advantage that additional models $h(\cdot)$ can be easily integrated into the overall system. The model scaling factors λ_1^M are trained according to the maximum entropy principle, e.g., using the GIS algorithm. Alternatively, one can train them with respect to the final translation quality measured by an error criterion (Och, 2003).

The log-linear model is a natural framework to integrate many models. The baseline system uses the following models:

- phrase translation model
- phrase count features
- word-based translation model
- word and phrase penalty
- target language model (6-gram)
- distortion model (assigning costs based on the jump width)

All the experiments in the paper are evaluated without rescoring. More details about the baseline system can be found in (Mauser et al., 2006)

3.2 Source Sentence Reordering Framework

Encouraged by the work of (Xia and McCord, 2004) and (Crego and Mariño, 2006), we also reorder the source language side. Compared to reordering on the target language side, one advantage is the efficiency since the reordering lattice can be translated monotonically as in (Zens et al., 2002). Another advantage is that there is correct sentence information for the reordering methods, because the source sentences are always given. Syntactic reordering on target language is difficult, since the methods will degrade much because of the errors in hypothesis.

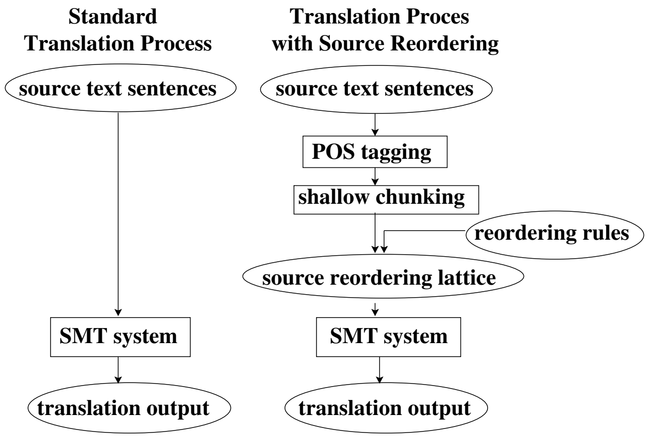

Figure 1: Illustration of the translation process with and without source reordering.

We apply reordering at the syntactic chunk level which can been seen as an intermediate level between full parsing and POS tagging. Figure 1 shows the differences between the new translation framework and the standard translation process. A reordering lattice replaces the original source sentence as the input to the translation system. The use of a lattice avoids hard decisions before translation. To generate the reordering lattice, the source sentence is first POS tagged and chunk parsed. Then, reordering rules are applied to the chunks to generate the reordering lattice.

Reordering rules are the key information for source reordering. They are automatically learned from the training data. The details of these two modules will be introduced in Section 5.

4 Reordering Rules

There has been much work on learning and applying reordering rules on source language, such as (Nießen and Ney, 2001; Xia and McCord, 2004; Collins et al., 2005; Chen et al., 2006; Crego and Mariño, 2006; Popović and Ney, 2006). The reordering rules could be composed of words, POS tags or syntactic tags of phrases. In our work, a rule is composed of chunk tags and POS tags. There is

Table 1: Examples of reordering rules. (lhs: chunk and POS tag sequence, rhs: permutation)

no.	lhs	rhs
1.	$NP_0\ PP_1\ u_2\ n_3$	0 1 2 3
2.	$NP_0\ PP_1\ u_2\ n_3$	3 0 1 2
3.	$DNP_0\ NP_1\ VP_2$	0 1 2
4.	$DNP_0\ NP_1\ VP_2$	1 0 2
5.	$DNP_0\ NP_1\ m_2$	0 1 2
6.	$DNP_0\ NP_1\ m_2\ ad_3$	3 0 1 2
7.	$DNP_0\ NP_1\ m_2\ ad_3\ v_4$	4 3 0 1 2

no hierarchical structure in a rule.

4.1 Definition of Reordering Rules

First, we show some rule examples in Table 1. A reordering rule consists of a left-hand-side (lhs) and a right-hand-side (rhs). The left-hand-side is a syntactic rule (chunk or POS tags), while the right-hand-side is the reordering positions of the rule. Different rules can share the same left-hand-side, such as rules no. 1, 2 and no. 3, 4. The rules record not only the *real* reordered chunk sequence, but also the monotone chunk sequences, like no. 1, 3 and 5. Note that the same tag sequence can appear multiple times according to different contexts, such as $DNP_0\ NP_1\ m_2$ # 0 1 2 in rules no. 5, 6, 7.

4.2 Extraction of Reordering Rules

The extraction of reordering rules is based on the word alignment and the source sentence chunks. Here, we train word alignments in both directions with GIZA++ (Och and Ney, 2003). To get alignment with high accuracy, we use the intersection alignment here.

For a given word-aligned sentence pair (f_1^J, e_1^I, a_1^J), the source word sequence f_1^J is first parsed into a chunk sequence F_1^K. Accordingly, the word-to-word alignment a_1^J is changed to a chunk-to-word alignment $\tilde{a}_1^K$ which is the combination of the target words aligned to the source words in a chunk. It is defined as:

$$\tilde{a}_k = \{i | i = a_j \wedge j \in [j_k, j_{k+1} - 1]\}$$

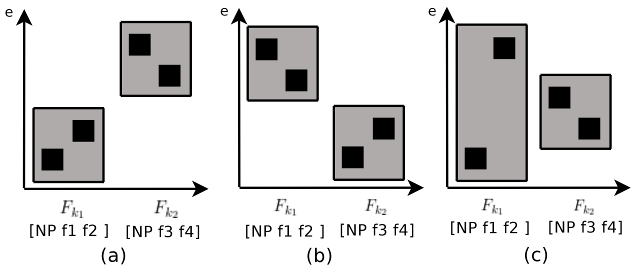

Figure 2: Illustration of three kinds of phrases: (a)monotone phrase, (b)reordering phrase, (c)cross phrase. The black box is a word-to-word alignment. The gray box is a chunk-to-word alignment.

Here, j_k denotes the position of the first source word in k^{th} chunk. The new alignment is $1 : m$ from source chunks to target words. It also means $\tilde{a}_k$ is a set of positions of target words.

We apply the standard phrase extraction algorithm (Zens et al., 2002) to $(F_1^K, e_1^I, \tilde{a}_1^K)$. Discarding the cross phrases, we keep the other phrases as rules. In a cross phrase, at least two chunk-word alignments overlap on the target language side. An example of a cross phrase is illustrated in Figure 2(c). Figure 2(a) and (b) illustrate the phrases for reordering rules, which could be monotone phrases or reordering phrases.

5 Reordering Lattice Generation

5.1 Parsing the Source Sentence

The first step of chunk parsing is word segmentation. Then, a POS tagger is usually needed for further syntactic analysis. In our experiments, we use the tool of "Inst. of Computing Tech., Chinese Lexical Analysis System (ICTCLAS)" (Zhang et al., 2003), which does the two tasks in one pass.

Referring to the description of the chunking task in CoNLL-2000[1], instead of English, a Chinese chunker is processed and evaluated. Each word is assigned a chunk tag, which contains the name of the chunk type and "B" for the first word of the chunk and "I" for each other word in the chunk. The "O" chunk tag is used for tokens which are not part of any chunk. We use the maximum entropy tool YAS-

[1] http://www.cnts.ua.ac.be/conll2000/chunking/

Figure 3: Example of applying rules. The left part is the used rules. The right part is the generated new orders of source words.

MET[2] to learn the chunking model. The model is based on a combination of word and POS tags. Since specific training and test data are not available for Chinese chunking, we convert subtrees of the Chinese treebank (LDC2005T01) into chunks. As there are many ways to choose a subtree, we uses the minimum subtree with the following constraints:

- a subtree has more than one child,
- the children of a subtree are all leaves.

Compared to chunking of English as in CoNLL-2000, there are more chunk types (24 instead of 6) and no single-word chunks. These two aspects make chunking for Chinese harder.

5.2 Applying Reordering Rules

First, we search the reordering rules, in which the chunk sequence matches any tag sequence in the input sentence. A source sentence has many paths generated by the rules . For a word uncovered by any rules, its POS tag is used. Each path corresponds to one sentence permutation.

The left part of the Figure 3 shows seven possible coverages, the right part is the reordering for each coverage. Some of the reorderings are identical, like the permutations in line 1, 3 and 5. That is because one word sequence is memorized by several rules in different contexts.

5.3 Lattice Weighting

All reorderings of an input sentence S are compressed and stored in a lattice. Each path is a possible reordering S' and is given a weight W. In this paper, the weight is computed using a source language model $p(S')$. The weight is used directly in the decoder, integrated into Equation (4). There is also a scaling factor for this weight, which is optimized together with other scaling factors on the development data. The probability of the reordered source sentence is calculated as follows: for a reordered source sentence $w_1 w_2 ... w_n$, the trigram language model is:

$$p(S') \quad = \quad \prod_{n=1}^{N} p(w_n | w_{n-2}, w_{n-1}) \tag{5}$$

Beside a word N-gram language model, a POS tag N-gram model or a chunk tag N-gram model could be used as well.

In this paper, we use a word trigram model. The model is trained on reordered training source sentences. A training source sentence is parsed into chunks. In the same way as described in Section 4.2, word-to-word alignments is converted to chunk-to-word alignments. We reorder the source chunks to monotonize the chunk-to-word alignments. The chunk boundaries are kept when this reordering is done.

6 Experiments

6.1 Chunking Result

In this section, we report results for chunk parsing. The annotation of the data is derived from the Chinese treebank (LDC2005T01). The corpus is split into two parts: 1000 sentences are randomly se-

[2] http://www-i6.informatik.rwth-aachen.de/web/Software /index.html

Table 2: Statistics of training and test corpus for chunk parsing.

	train	test
sentences	17 785	1 000
words	486 468	21 851
chunks	105 773	4 680
words out of chunks	244 416	10 282

Table 3: Chunk parsing result on 1000 sentences.

accuracy	precision	recall	F-measure
74.51%	65.2%	61.5%	63.3

lected as test data. The remaining part is used for training. The corpus is from the newswire domain.

Table 2 shows the corpus statistics. For the 4 680 chunks in the test set, the chunker has found 4 414 chunks, of which 2 879 are correct. Following the criteria of CoNLL-2000, the chunker is evaluated using the F-score, which is a combination of precision and recall. The result is shown in Table 3.

The accuracy is evaluated at the word level, the other three metrics are evaluated at the chunk level. The results at the chunk level are worse than at the word level, because a chunk is counted as correct only if the chunk tag and the chunk boundaries are both correct.

6.2 Translation Results

For the translation experiments, we report the two accuracy measures BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) as well as the two error rates word error rate (WER) and position-independent word error rate (PER).

We perform translation experiments on the Basic Traveling Expression Corpus (BTEC) for the Chinese-English task. It is a speech translation task in the domain of tourism-related information. We report results on the IWSLT 2004, 2005 and 2006 evaluation test sets. There are 16 reference translations for the IWSLT 2004 and 2005 tasks and 7 reference translations for the IWSLT 2006 task.

Table 4 shows the corpus statistics of the task. A training corpus is used to train the translation model, the language model and to obtain the reordering rules. A development corpus is used to optimize the scaling factors for the BLEU score. The English text is processed using a tokenizer. The Chinese text processing uses word segmentation with the ICTCLAS segmenter (Zhang et al., 2003). The translation is evaluated case-insensitive and without punctuation marks.

Table 4: Statistics of training and test corpora for the IWSLT tasks.

		Chinese	English
Train	Sentences	40k	
	Words	308k	377k
Dev	Sentences	489	
	Words	5 478	6 008
Test IWSLT04	Sentences	500	
	Words	3 866	3 581
Test IWSLT05	Sentences	506	
	Words	3 652	3 579
Test IWSLT06	Sentences	500	
	Words	5 846	–

The translation results are presented in Table 5. The baseline system is a non-monotone translation system, in which the decoder does reordering on the target language side. Compared to the baseline system, the source reordering method improves the BLEU score by $0.5\% - 1.8\%$ absolute. It also achieves a better WER. Note that the used chunker here is out-of-domain [3]. An improvement is achieved even with a low F-measure for chunking. So, we could hope that larger improvement is possible using a high-accuracy chunker.

Though the input is a lattice, the source reordering is still faster than the reordering during decoding, e.g. for the IWSLT 2006 test set, the baseline system took 17.5 minutes and the source reordering system took 12.3 minutes. The result also indicates that the non-monotone decoding hurts the performance in a source reordering framework. A similar conclusion is also presented in (Xia and McCord, 2004).

Additional experiments we carried out to compare POS-level and chunk-level reorderings. We delete the chunk information and keep the POS tags. Then,

[3] The chunker is trained on newswire data, but the test data is from the tourism domain.

Table 5: Translation performance for the Chinese-English IWSLT task

		WER[%]	PER[%]	NIST	BLEU[%]
IWSLT04	baseline	47.3	38.2	7.78	39.1
	source reordering	46.3	37.2	7.70	40.9
IWSLT05	baseline	45.0	37.3	7.40	41.8
	source reordering	44.6	36.8	7.51	42.3
IWSLT06	baseline	67.4	50.0	6.65	22.4
	source reordering	65.6	50.4	6.46	23.3
	source reordering+non-monotone decoder	66.5	50.3	6.52	22.4

Table 6: Translation performance of reordering methods on IWSLT 2004 test set

	WER [%]	PER [%]	NIST	BLEU [%]
Baseline	47.3	38.2	7.78	39.1
POS	46.9	37.5	7.38	39.7
Chunk	46.3	37.2	7.70	40.9

Table 7: Lattice information for the Chinese-English IWSLT 2004 test data

	avg. density pro sent	used rules	translation time [min/sec]
POS	15.7	6 868	7:08
Chunk	8.2	3 685	3:47

we rerun the source reordering system on the IWSLT 2004 test set. The translation results are shown in Table 6. Though the accuracy of chunking is low, the chunk-level method gets better results than POS-level method. With POS tags, we get more reordering rules and more paths in the lattice, since the sentence length is longer than with chunks. The statistics are shown in Table 7.

7 Conclusions and Future Work

This paper presents a source-side reordering method which is based on syntactic chunks. The reordering rules are automatically learned from bilingual data. To avoid hard decision before decoding, a reordering lattice representing all possible reorderings is used instead of single source sentence for decoding. The experiments demonstrate that even with a very poor chunker, the chunk-level source reordering is still helpful for a state-of-the-art statistical translation system and it has better performance than the POS-level source reordering and target-side reordering.

There are some directions for future work. First, we would like to try this method on larger data sets and other language pairs. Second, we are going to improve the chunking accuracy. Third, we would reduce the number of rules and prune the lattice.

Acknowledgments

This material is partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023, and was partially funded by the Deutsche Forschungsgemeinschaft (DFG) under the project "Statistische Textübersetzung" (Ne572/5)

References

A. L. Berger, S. A. Della Pietra, and V. J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72, March.

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June.

B. Chen, M. Cettolo, and M. Federico. 2006. Reordering rules for phrase-based statistical machine translation. In *Int. Workshop on Spoken Language Translation Evaluation Campaign on Spoken Language Translation*, pages 1–15, Kyoto, Japan, November.

M. Collins, P. Koehn, and I. Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 531–540, Ann Arbor, Michigan, June.

M. R. Costa-jussà and J. A. R. Fonollosa. 2006. Statistical machine reordering. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*, pages 70–76, Sydney, Australia, July.

J. M. Crego and J. B. Mariño. 2006. Integration of postag-based source reordering into SMT decoding by an extended search graph. In *Proc. of AMTA06*, pages 29–36, Massachusetts, USA, August.

G. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proc. ARPA Workshop on Human Language Technology*.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 961–968, Sydney, Australia, July.

D. Gildea. 2003. Loosely tree-based alignment for machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 80–87, Sapporo, Japan, July.

J. Graehl and K. Knight. 2004. Training tree transducers. In *HLT-NAACL 2004: Main Proc.*, pages 105–112, Boston, Massachusetts, USA, May 2 - May 7.

S. Kanthak, D. Vilar, E. Matusov, R. Zens, and H. Ney. 2005. Novel reordering approaches in phrase-based statistical machine translation. In *43rd Annual Meeting of the Assoc. for Computational Linguistics: Proc. Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond*, pages 167–174, Ann Arbor, Michigan, June.

P. Koehn and K. Knight. 2003. Empirical methods for compound splitting. In *Proc. 10th Conf. of the Europ. Chapter of the Assoc. for Computational Linguistics (EACL)*, pages 347–354, Budapest, Hungary, April.

A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney. 2006. The RWTH Statistical Machine Translation System for the IWSLT 2006 Evaluation. In *Proc. of the Int. Workshop on Spoken Language Translation*, pages 103–110, Kyoto, Japan.

I. Melamed. 2004. Statistical machine translation by parsing. In *The Companion Volume to the Proc. of 42nd Annual Meeting of the Association for Computational Linguistics*, pages 653–660.

S. Nießen and H. Ney. 2001. Morpho-syntactic analysis for reordering in statistical machine translation. In *Proc. of MT Summit VIII*, pages 247–252.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.

F. J. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, March.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proc. 2004 Human Language Technology Conf. / North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT-NAACL)*, pages 161–168, Boston,MA.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the 41th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, July.

M. Popović and H. Ney. 2006. POS-based word reorderings for statistical machine translation. In *Proc. of the Fifth Int. Conf. on Language Resources and Evaluation (LREC)*.

L. Shen, A. Sarkar, and F. J. Och. 2004. Discriminative reranking for machine translation. In *HLT-NAACL 2004: Main Proc.*, pages 177–184, Boston, Massachusetts, USA, May 2 - May 7.

C. Tillmann, S. Vogel, H. Ney, and A. Zubiaga. 1997. A DP-based search using monotone alignments in statistical translation. In *Proc. 35th Annual Conf. of the Association for Computational Linguistics*, pages 289–296, Madrid, Spain, July.

D. Wu. 1996. A polynomial-time algorithm for statistical machine translation. In *Proc. 34th Annual Meeting of the Assoc. for Computational Linguistics*, pages 152–158, Santa Cruz, CA, June.

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

F. Xia and M. McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proc. of COLING04*, pages 508–514, Geneva, Switzerland, Aug 23–Aug 27.

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

R. Zens, F. J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In M. Jarke, J. Koehler, and G. Lakemeyer, editors, *25th German Conf. on Artificial Intelligence (KI2002)*, volume 2479 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 18–32, Aachen, Germany, September. Springer Verlag.

H. P. Zhang, Q. Liu, X. Q. Cheng, H. Zhang, and H. K. Yu. 2003. Chinese lexical analysis using hierarchical hidden markov model. In *Proc. of the second SIGHAN workshop on Chinese language processing*, pages 63–70, Morristown, NJ, USA.