A Beam Search Algorithm for ITG Word Alignment

Peng Li Yang Liu Maosong Sun
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
pengli09@gmail.com, {liuyang2011,sms}@tsinghua.edu.cn

ABSTRACT
Inversion transduction grammar (ITG) provides a syntactically motivated solution to modeling the distortion of words between two languages. Although the Viterbi ITG alignments can be found in polynomial time using a bilingual parsing algorithm, the computational complexity is still too high to handle real-world data, especially for long sentences. Alternatively, we propose a simple and effective beam search algorithm. The algorithm starts with an empty alignment and keeps adding single promising links as early as possible until the model probability does not increase. Experiments on Chinese-English data show that our algorithm is one order of magnitude faster than the bilingual parsing algorithm with bitext cell pruning without loss in alignment and translation quality.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE, CHINESE

一种ITG词语对齐的柱搜索算法

反向转录语法为两种语言间的词语调序提供了一种句法驱动的解决方案。虽然双语句法分析算法可以在多项式时间内搜索到Viterbi对齐，其计算复杂度依然太高，难以处理包含很多长句子的真实数据。为此，我们提出一种简单有效的柱搜索算法。该算法以空对齐为起点，优先选择最好的连线添加到对齐中，直至无法提高模型概率为止。在汉英数据上的实验结果表明，我们的算法比使用剪枝技术的双语分析算法快一个数量级，同时保持了对齐和翻译的质量。

KEYWORDS: word alignment, inversion transduction grammar, beam search.

KEYWORDS IN L_2: 词语对齐, 反向转录语法, 柱搜索.

1 Introduction

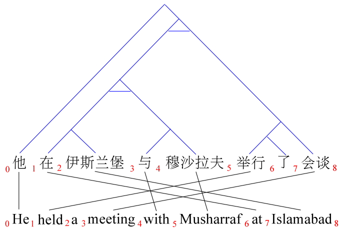
Word alignment plays an important role in statistical machine translation (SMT) as it indicates the correspondence between two languages. The parameter estimation of many SMT models rely heavily on word alignment. Och and Ney (2004) firstly introduce alignment consistency to identify equivalent phrase pairs. Simple and effective, rule extraction algorithms based on word alignment have also been extended to hierarchial phrase-based (Chiang, 2007) and syntax-based (Galley et al., 2004) SMT systems successfully. Studies reveal that word alignment has a profound effect on the performance of SMT systems (Ayan and Dorr, 2006; Fraser and Marcu, 2007).

One major challenge in word alignment is modeling the permutations of words between source and target sentences. Due to the diversity of natural languages, the word orders of source and target sentences are usually quite different, especially for distantly-related language pairs such as Chinese and English. While most word alignment approaches either use distortion models (Brown et al., 1993; Vogel and Ney, 1996) or features (Taskar et al., 2005; Moore, 2005; Moore et al., 2006; Liu et al., 2010c) to capture reordering of words, *inversion transduction grammar* (ITG) (Wu, 1997) provides a syntactically motivated solution. ITG is a synchronous grammar of which a derivation explains how a source sentence and a target sentence are generated synchronously. By recursively merging blocks (i.e., consecutive word sequences) either in a monotone order or an inverted order, ITG constrains the search space of distortion in a way that proves to be effective in both alignment (Zhang and Gildea, 2005, 2006; Haghighi et al., 2009; Liu et al., 2010a,b) and translation (Zens and Ney, 2003; Xiong et al., 2006) benchmark tests.

Although ITG only requires $O(n^6)$ time for finding Viterbi alignment, which is a significant improvement over the intractable search problem faced by most alignment models (Brown et al., 1993; Moore et al., 2006; Liu et al., 2010c), the degree of the polynomial is still too high for practical use. For example, the maximal sentence length of bilingual corpus is often set to 100 words in Moses (Koehn et al., 2007), a state-of-the-art SMT system. Synchronous parsing of such long sentences can be prohibitively slow, making ITG alignment methods hard to deal with large scale real-world data.

To alleviate this problem, many pruning methods have been proposed to reduce the computational complexity of synchronous parsing by pruning less promising cells. Zhang and Gildea (2005) introduce a tic-tac-toe pruning method based on IBM model 1 probabilities. Haghighi et al. (2009) use posterior predictions from simpler alignment models for identifying degenerate cells. Liu et al. (2010a) propose a discriminative framework to integrate all informative features to constrain the search space of ITG alignment.

Instead of using synchronous parsing to search for Viterbi ITG alignments, we propose a simple and effective search algorithm extended from the beam search algorithm proposed by Liu et al. (2010c). The algorithm starts with an empty alignment and keeps adding single links until the model probability does not increase. During the search process, a shift-reduce algorithm is used to verify the ITG constraint. As our algorithm runs in $O(bn^3)$ time, where b is the beam size, it is about 1000 times faster than the $O(n^6)$ time bilingual parsing algorithm empirically. More importantly, experiments on Chinese-English data show that our algorithm is 20 times faster than bilingual parsing with tic-tac-toe pruning (Zhang and Gildea, 2005) when achieving comparable alignment and translation quality.


0 他 1 在 2 伊斯兰堡 3 与 4 穆沙拉夫 5 举行 6 了 7 会谈 8
0 He 1 held 2 a 3 meeting 4 with 5 Musharraf 6 at 7 Islamabad 8

1) $X_{[0,8,0,8]} \rightarrow [X_{[0,1,0,1]}\ X_{[1,8,1,8]}]$
2) $X_{[0,1,0,1]} \rightarrow$ 他/he
3) $X_{[1,8,1,8]} \rightarrow \langle X_{[1,5,4,8]}\ X_{[5,8,1,4]}\rangle$
4) $X_{[1,5,4,8]} \rightarrow \langle X_{[1,3,6,8]}\ X_{[3,5,4,6]}\rangle$
5) $X_{[1,3,6,8]} \rightarrow [X_{[1,2,6,7]}\ X_{[2,3,7,8]}]$
6) $X_{[1,2,6,7]} \rightarrow$ 在/at
7) $X_{[2,3,7,8]} \rightarrow$ 伊斯兰堡/Islamabad
8) $X_{[3,5,4,6]} \rightarrow [X_{[3,4,4,5]}\ X_{[4,5,5,6]}]$
9) $X_{[3,4,4,5]} \rightarrow$ 与/with
10) $X_{[4,5,5,6]} \rightarrow$ 穆沙拉夫/Musharraf
11) $X_{[5,8,1,4]} \rightarrow [X_{[5,7,1,3]}\ X_{[7,8,3,4]}]$
12) $X_{[5,7,1,3]} \rightarrow [X_{[5,6,1,2]}\ X_{[6,7,2,3]}]$
13) $X_{[5,6,1,2]} \rightarrow$ 举行/held
14) $X_{[6,7,2,3]} \rightarrow [X_{[6,7,2,2]}\ X_{7,7,2,3}]$
15) $X_{[6,7,2,2]} \rightarrow$ 了/ϵ
16) $X_{[7,7,2,3]} \rightarrow \epsilon$/a
17) $X_{[7,8,3,4]} \rightarrow$ 会谈/meeting

Figure 1: An ITG derivation for a Chinese-English sentence pair.

2 Beam Search for ITG Word Alignment

Inversion transduction grammar (ITG) (Wu, 1997) is a synchronous grammar for synchronous parsing of source and target language sentences. It builds a synchronous parse tree that indicates the correspondence as well as permutation of blocks (i.e., consecutive word sequences) based on the following production rules:

$$(1)\ X \rightarrow [X\ X],\ (2)\ X \rightarrow \langle X\ X\rangle,\ (3)\ X \rightarrow f/e,\ (4)\ X \rightarrow f/\epsilon,\ (5)\ X \rightarrow \epsilon/e,$$

where X is a non-terminal, f is a source word, e is a target word, and ϵ is an empty word. While rule (1) merges two blocks in a monotone order, rule (2) merges in an inverted order. Rules (3) – (5) are responsible for aligning source and target words.

Figure 1 shows an ITG derivation for a Chinese-English sentence pair $\langle \mathbf{f}_0^J, \mathbf{e}_0^I\rangle$. The subscript of a non-terminal X denotes a bilingual span $[s,t,u,v]$ that corresponds to a block pair $\langle \mathbf{f}_s^t, \mathbf{e}_u^v\rangle$, where $\mathbf{f}_s^t = \mathbf{f}_{s+1}\ldots\mathbf{f}_t$ and $\mathbf{e}_u^v = \mathbf{e}_{u+1}\ldots\mathbf{e}_v$. An empty source word is represented as $\mathbf{f}_s^s$ and $\mathbf{e}_u^u$ for the target case.

The decision rule of finding the Viterbi alignment $\hat{\mathbf{a}}$ for a sentence pair $\langle \mathbf{f}_0^J, \mathbf{e}_0^I\rangle$ is given by [1]

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} \left\{ \prod_{(j,i)\in\mathbf{a}} p(\mathbf{f}_j, \mathbf{e}_i) \times \prod_{j\notin\mathbf{a}} p(\mathbf{f}_j, \epsilon) \times \prod_{i\notin\mathbf{a}} p(\epsilon, \mathbf{e}_i) \right\} \tag{1}$$

Traditionally, this can be done in $O(n^6)$ time using bilingual parsing (Wu, 1997).

In this paper, we extend a beam search algorithm (Liu et al., 2010c) to search for Viterbi ITG word alignment. Starting from an empty word alignment, the beam search algorithm

[1] For simplicity, we assume the distribution for the binary rules $X \rightarrow [X\ X]$ and $X \rightarrow \langle X\ X\rangle$ is uniform. Xiong et al. (2006) propose a maximal entropy model to distinguish between two merging options based on lexical evidence. We leave this for future work.

Algorithm 1 A beam search algorithm for ITG alignment.

```
1: procedure AlignItg($\mathbf{f}, \mathbf{e}$)
2:     $\hat{\mathbf{a}} \rightarrow \emptyset$                                  ▷ the alignment with highest probability
3:     $\mathscr{L} \rightarrow \{(j,i) : p(\mathbf{f}_j, \mathbf{e}_i) > p(\mathbf{f}, \epsilon) \times p(\epsilon, \mathbf{e})\}$     ▷ a set of promising links
4:     $open \leftarrow \emptyset$                                  ▷ a list of active alignments
5:     $\mathbf{a} \leftarrow \emptyset$                                     ▷ begin with an empty alignment
6:     Add($open, \mathbf{a}, \beta, b$)                            ▷ initialize the list
7:     while $open \neq \emptyset$ do
8:         $closed \leftarrow \emptyset$                            ▷ a list of expanded alignments
9:         for all $\mathbf{a} \in open$ do
10:            for all $l \in \mathscr{L} - \mathbf{a}$ do               ▷ enumerate all possible new links
11:                $\mathbf{a}' \leftarrow \mathbf{a} \cup \{l\}$                    ▷ produce a new alignment
12:                if Itg($\mathbf{a}'$) then                     ▷ ensure the ITG constraint
13:                    Add($closed, \mathbf{a}', \beta, b$)          ▷ update expanded alignments
14:                    if $\mathbf{a}' > \hat{\mathbf{a}}$ then
15:                        $\hat{\mathbf{a}} = \mathbf{a}'$                      ▷ update the best alignment
16:                    end if
17:                end if
18:            end for
19:        end for
20:        $open \leftarrow closed$                                ▷ update active alignments
21:    end while
22:    return $\hat{\mathbf{a}}$                             ▷ return the alignment with highest probability
23: end procedure
```

proposed by Liu et al. (2010c) keeps adding single links to current alignments until all expanded alignments do not have higher probabilities. From a graphical point of view, the search space is organized as a directed acyclic graph[2] that consists of $2^{J \times I}$ nodes and $J \times I \times 2^{J \times I - 1}$ edges. The nodes are divided into $J \times I + 1$ layers. The number of nodes in the kth layer ($k = 0, \ldots, J \times I$) is $\binom{J \times I}{k}$. The maximum of layer width is given by $\binom{J \times I}{\lfloor \frac{J \times I}{2} \rfloor}$. The goal of word alignment is to find a node that has the highest probability in the graph.

The major difference of our algorithm from (Liu et al., 2010c) is that we only consider ITG alignments. Wu (1997) shows that ITG alignments only account for 0.1% in the full search space. The percentage is even lower for long sentences. As the worst-case running time is $O(bn^4)$ (b is a beam size) for the beam search algorithm of Liu et al. (2010c), this can be reduced to $O(bn^3)$ for the beam search algorithm that searches for ITG word alignment. [3]

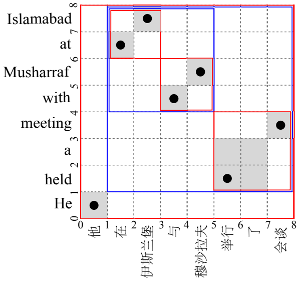
Algorithm 1 shows the beam search algorithm for ITG alignment. The best alignment is set to empty at the beginning (line 2). The algorithm collects *promising* links $\mathscr{L}$ before alignment expansion (line 3). By promising, we mean that adding a link will increase the probability of current alignment. The gains keep fixed during the search process: [4]

$$\forall \mathbf{a} \in \mathscr{A} : gain(\mathbf{a}, \mathbf{f}, \mathbf{e}, l) \equiv \frac{p(\mathbf{f}_j, \mathbf{e}_i)}{p(\mathbf{f}_j, \epsilon) \times p(\epsilon, \mathbf{e}_i)}, \tag{2}$$

[2] For space limitation, please refer to Figure 3 in (Liu et al., 2010c) for example.

[3] If the Viterbi alignment is a full alignment, i.e., there is a link between any pair of source and target words, and the beam size is 1, $\frac{(J \times I) \times (J \times I + 1)}{2}$ nodes will be explored. Apparently, this can hardly happen in practice. For ITG alignments, however, our algorithm can reach at most the $min(J, I)$-th layer because ITG only allows for one-to-one links.

[4] As ITG alignments are strictly one-to-one, the gain of adding a link $l = (j, i)$ only depends on the associated source word $\mathbf{f}_j$ and target word $\mathbf{e}_i$.



step	operation	stack
1	S	[0, 1, 0, 1]
2	S	[0, 1, 0, 1] [1, 2, 6, 7]
3	S	[0, 1, 0, 1] [1, 2, 6, 7] [2, 3, 7, 8]
4	R_M	[0, 1, 0, 1] [1, 3, 6, 8]
5	S	[0, 1, 0, 1] [1, 3, 6, 8] [3, 4, 4, 5]
6	S	[0, 1, 0, 1] [1, 3, 6, 8] [3, 4, 4, 5] [4, 5, 5, 6]
7	R_M	[0, 1, 0, 1] [1, 3, 6, 8] [3, 5, 4, 6]
8	R_I	[0, 1, 0, 1] [1, 5, 4, 8]
9	S	[0, 1, 0, 1] [1, 5, 4, 8] [5, 7, 1, 3]
10	S	[0, 1, 0, 1] [1, 5, 4, 8] [5, 7, 1, 3] [7, 8, 3, 4]
11	R_M	[0, 1, 0, 1] [1, 5, 4, 8] [5, 8, 1, 4]
12	R_I	[0, 1, 0, 1] [1, 8, 1, 8]
13	R_M	[0, 8, 0, 8]

Figure 2: A shift-reduce algorithm for judging ITG alignment.

where $\mathscr{A}$ is the set of all possible alignments. So our algorithm can safely take the computation of gains out of the loop (i.e., lines 7-21), which can not be done in (Liu et al., 2010c).

For each alignment, the algorithm calls a procedure $\textsc{Itg}(\mathbf{a})$ to verify whether it is an ITG alignment or not (line 12). We use a shift-reduce algorithm for ITG verification. As shown in Figure 2, the shift-reduce algorithm scans links from left to right on the source side. Each link (j, i) is treated as an atomic block $[j-1, j, i-1, i]$. The algorithm maintains a stack of blocks, on which three operators are defined:

1. S: shift a block into the stack;
2. R_M: merge two blocks in a monotone order;
3. R_I: merge two blocks in an inverted order.

The algorithm runs in a reduce-eager manner: merge blocks as soon as possible (e.g., [5, 7, 1, 3] in step 9). Unaligned words are attached to the left nearest aligned words deterministically. The alignment satisfies the ITG constraint if and only if the algorithm manages to find a block corresponding to the input sentence pair. The shift-reduce algorithm runs in linear time. [5]

At each level, the algorithm at most retains b alignments (line 13). As ITG only allows for one-to-one links, the beam search algorithm runs for at most $min(J,I)+1$ iterations (lines 7-21)[6]. Therefore, the running time of our beam search algorithm is $O(bn^3)$.

3 Experiments

We evaluated our algorithm on Chinese-English data for both alignment and translation. As Haghighi et al. (2009) has already compared ITG alignment with GIZA++ and discriminative methods, we only focus on comparing the search algorithms for ITG alignment. Our algorithm is compared with two baseline methods:

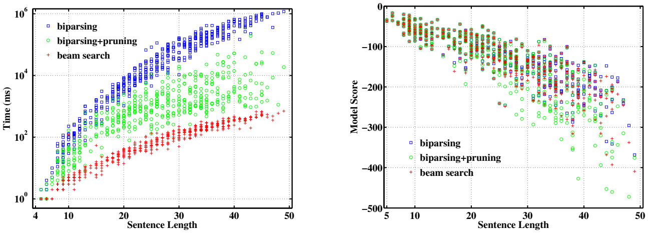
1. *biparsing*: the bilingual parsing algorithm as described in (Wu, 1997);

[5] In practice, the algorithm can be even more efficient by recording the sequence of blocks in each hypothesis without unaligned word attachment. Therefore, block merging needs not to start from scratch for each hypothesis.

[6] In the worst case, $min(J,I)$ links will be added in $min(J,I)$ iterations, in the $min(J,I)+1$ iteration, all the expanded alignments will validate the ITG constrain and the algorithm terminates.

algorithm	setting	average time (s)↓	average model score↑	AER↓
biparsing		126.164	**-127.17**	**29.13**
biparsing+pruning	$t = 10^{-3}$	2.404	-167.44	34.92
	$t = 10^{-4}$	3.002	-152.68	33.13
	$t = 10^{-5}$	3.571	-144.27	31.93
	$t = 10^{-6}$	5.427	-138.23	31.12
beam search	$b = 1$	**0.019**	-142.27	33.00
	$b = 10$	0.126	**-131.73**	**30.52**

Table 1: Comparison with bilingual parsing algorithms in terms of average time per sentence pair, average model score per sentence pair and AER (length ≤ 50 words on both sides). Note that t is the beam ratio in tic-tac-toe pruning (Zhang and Gildea, 2005).



(a) Comparison of aligning time.

(b) Comparison of model score.

Figure 3: Comparison of aligning time and model score over various sentence lengths.

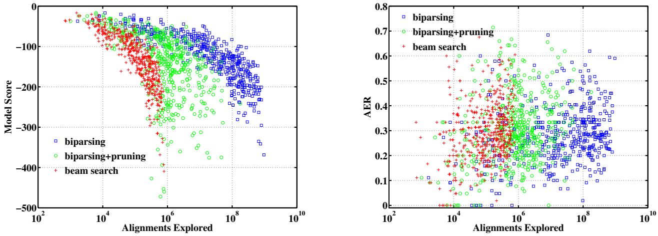
2. *biparsing + pruning*: the bilingual parsing algorithm with tic-tac-toe pruning (Zhang and Gildea, 2005).

For simplicity, we used the IBM model 4 translation probabilities trained on the FBIS corpus (6.5M+8.4M words) to approximate ITG lexical probabilities in the following experiments: $p(f,e) \approx p_{m4}(f|e) \times p_{m4}(e|f)/2$, $p(f,\epsilon) \approx p_{m4}(f|\epsilon)$, $p(\epsilon,e) \approx p_{m4}(e|\epsilon)$.

3.1 Alignment Evaluation

For the alignment evaluation, we selected 461 sentence pairs that contain at most 50 words on both sides from the hand-aligned dataset of (Liu et al., 2005). The three ITG alignment methods are compared in terms of average time per sentence pair, average model score per sentence pair, and AER. The results are shown in Table 1. Although achieving the best model score and AER, the *biparsing* algorithm runs too slow: 126.164 seconds per sentence pair on average. This is impractical for dealing with large scale real-world data that usually contains millions of sentence pairs. The tic-tac-toe pruning method (*biparsing + pruning*) does increase the speed by two orders of magnitude (3.571 seconds per sentence pair), which confirms the effectiveness of cell pruning (Zhang and Gildea, 2005; Haghighi et al., 2009; Liu et al., 2010a). Our beam search algorithm is one order of magnitude faster than the *biparsing+pruning* algorithm with significantly less search error.

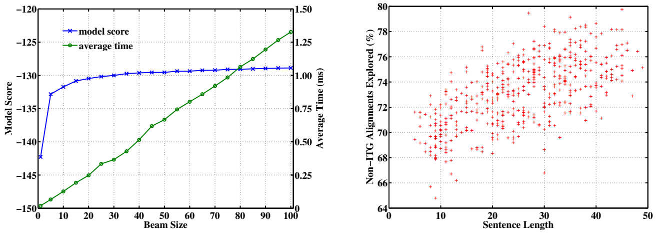
Figure 3 compares aligning time of the three algorithms over different sentence lengths ranging



(a) Comparison of model score over alignments explored.

(b) Comparison of AER over alignments explored.

Figure 4: The scatter diagram of model score and AER over alignments explored for the 461 evaluation sentence pairs.



(a) Effect of beam size on average time per sentence pair and model score per sentence pair.

(b) Percentages of non-ITG alignments explored.

Figure 5: Investigation on different properties of our algorithm.

from 5 to 50 words. Clearly, our beam search algorithm is faster than the *biparsing* and *biparsing + pruning* algorithms for all lengths. More importantly, the gap enlarges with the increase of sentence length. We observe that our search algorithm almost achieves the same model scores with *biparsing* and *biparsing + pruning* for short sentences. For long sentences, however, the differences are increasingly significant because it is hard to find Viterbi alignments for long sentences. In most cases, our algorithm achieves higher model scores than *biparsing+pruning*, which is consistent with Table 1.

Figure 4 shows the model score and AER over alignments explored. Generally, our beam search algorithm explores less alignments before reaching the same level of model score and AER than *biparsing* and *biparsing + pruning*. And the diversity between different sentences is much smaller than the other two algorithms. So our beam search algorithm is more efficient.

Figure 5(a) shows the effect of beam size on average time per sentence pair and average model score per sentence pair. While the theoretical running time is $O(bn^3)$, the empirical average time does increase linearly with the beam size. The model score also generally rises with the increase of beam size but grows insignificantly when $b > 20$.

Figure 5(b) shows the percentages of non-ITG alignments explored during the search process. We observe that generally over 68% alignments expanded are non-ITG alignments and the percentage increases for long sentences. This finding suggests that most of expanded alignments are verified as non-ITG, especially for long sentences. Our algorithm can be significantly improved if it manages to know which link will result in an ITG alignment before calling the ITG(**a**) procedure. We leave this for future work.

3.2 Translation Evaluation

For the translation evaluation, we used 138K sentence pairs that have at most 40 words from the FBIS corpus as the training set, NIST 2002 dataset as the development set, and NIST 2005 dataset as the test set. As the *biparsing* algorithm runs too slow on the training data, we only compared our algorithm with *biparsing+pruning* in terms of average time per sentence pair and BLEU. *Moses* (Koehn et al., 2007) (a state-of-the-art phrase-based SMT system) and *Joshua* (Li et al., 2009) (a state-of-the-art hierarchial phrase-based SMT system) are used in our experiments. Both of them are used with default settings, except that word alignments are produced by "*biparsing+pruning*" and "*beam search*" respectively rather than GIZA++. Table 2 shows the average aligning time as well as the BLEU scores obtained by Moses and Joshua. Our system runs 20 times faster than the baseline without significant loss in translation quality.

algorithm	setting	average time (s)	Moses	Joshua
biparsing+pruning	$t = 10^{-5}$	7.57	23.86	**23.77**
beam search	$b = 10$	**0.35**	**23.95**	23.38

Table 2: Comparison of average time per sentence pair and BLEU scores (trained on the sentence pairs with no more than 40 words of FBIS corpus). Our system runs 20 times faster than the baseline without significant loss in translation quality.

Conclusion

We have presented a simple and effective algorithm for finding Viterbi ITG alignments. With a time complexity of $O(bn^3)$, the algorithm starts with an empty alignment and keeps adding single links until the model probability does not increase. Our experiments on Chinese-English data show that the proposed beam search algorithm is one order of magnitude faster than the conventional bilingual parsing algorithm with tic-tac-toe pruning without loss in alignment and translation quality.

In the future, we plan to extend our algorithm to the block-based ITG with discriminative training (Haghighi et al., 2009; Liu et al., 2010a), which proves to deliver state-of-the-art alignment and translation performance. It is interesting to include maximum entropy reordering models (Xiong et al., 2006) to make better predictions for binary rules. In addition, adding an estimate of future cost will help reduce search error further.

Acknowledgments

This research is supported by the Boeing Tsinghua Joint Research Project "English-Chinese Bilingual Text Alignment and Study of English-Chinese Translation" and National High-tech R&D Program (863 Program) under the grant no. 2011AA01A207 and 2012AA011102. We are grateful to Dr. Ping Xue for his inspirational suggestions in early discussions.

References

Ayan, N. F. and Dorr, B. J. (2006). Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proceedings of COLING·ACL 2006*, pages 9–16.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Fraser, A. and Marcu, D. (2007). Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–302.

Galley, M., Hopkins, M., Knight, K., and Marcu, D. (2004). What's in a translation rule? In *Proceedings of NAACL 2004*, pages 273–280.

Haghighi, A., Blitzer, J., DeNero, J., and Klein, D. (2009). Better word alignments with supervised ITG models. In *Proceedings of ACL-IJCNLP 2009*, pages 923–931.

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 ACL 2007 (the Demo and Poster Sessions)*, pages 177–180.

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Khudanpur, S., Schwartz, L., Thornton, W. N. G., Weese, J., and Zaidan, O. F. (2009). Joshua: An open source toolkit for parsing-based machine translation. In *Proceedings of WMT 2009*, pages 135–139.

Liu, S., Li, C.-H., and Zhou, M. (2010a). Discriminative pruning for discriminative ITG alignment. In *Proceedings of ACL 2010*, pages 316–324.

Liu, S., Li, C.-H., and Zhou, M. (2010b). Improved discriminative ITG alignment using hierarchical phrase pairs and semi-supervised training. In *Proceedings of COLING 2010*, pages 730–738.

Liu, Y., Liu, Q., and Lin, S. (2005). Log-linear models for word alignment. In *Proceedings of ACL 2005*, pages 459–466.

Liu, Y., Liu, Q., and Lin, S. (2010c). Discriminative word alignment by linear modeling. *Computational Linguistics*, 36(3):303–339.

Moore, R. C. (2005). A discriminative framework for bilingual word alignment. In *Proceedings of EMNLP 2005*, pages 81–88.

Moore, R. C., Yih, W.-t., and Bode, A. (2006). Improved discriminative bilingual word alignment. In *Proceedings of COLING·ACL 2006*, pages 513–520.

Och, F. and Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Taskar, B., Lacoste-Julien, S., and Klein, D. (2005). A discriminative matching approach to word alignment. In *Proceedings of EMNLP 2005*, pages 73–80.

Vogel, S. and Ney, H. (1996). HMM-based word alignment in statistical translation. In *Proceedings of COLING 1996*, pages 836–841.

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

Xiong, D., Liu, Q., and Lin, S. (2006). Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of COLING·ACL 2006*, pages 521–528.

Zens, R. and Ney, H. (2003). A comparative study on reordering constraints in statistical machine translation. In *Proceedings of ACL 2003*, pages 144–151.

Zhang, H. and Gildea, D. (2005). Stochastic lexicalized inversion transduction grammar for alignment. In *Proceedings of ACL 2005*, pages 475–482.

Zhang, H. and Gildea, D. (2006). Efficient search for inversion transduction grammar. In *Proceedings of EMNLP 2006*, pages 224–231.