Discriminative Word Alignment by Linear Modeling

Yang Liu*
Institute of Computing Technology
Chinese Academy of Sciences

Qun Liu*
Institute of Computing Technology
Chinese Academy of Sciences

Shouxun Lin*
Institute of Computing Technology
Chinese Academy of Sciences

Word alignment plays an important role in many NLP tasks as it indicates the correspondence between words in a parallel text. Although widely used to align large bilingual corpora, generative models are hard to extend to incorporate arbitrary useful linguistic information. This article presents a discriminative framework for word alignment based on a linear model. Within this framework, all knowledge sources are treated as feature functions, which depend on a source language sentence, a target language sentence, and the alignment between them. We describe a number of features that could produce symmetric alignments. Our model is easy to extend and can be optimized with respect to evaluation metrics directly. The model achieves state-of-the-art alignment quality on three word alignment shared tasks for five language pairs with varying divergence and richness of resources. We further show that our approach improves translation performance for various statistical machine translation systems.

1. Introduction

Word alignment, which can be defined as an object for indicating the corresponding words in a parallel text, was first introduced as an intermediate result of statistical machine translation (Brown et al. 1993).

Consider the following Chinese sentence and its English translation:

中国 建筑业 对外开放 呈现 新 格局
Zhongguo jianzhuye duiwaikaifang chengxian xin geju

The opening of China's construction industry to the outside presents a new structure

* Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, No. 6 Kexueyuan South Road, Haidian District, P.O. Box 2704, Beijing 100190, China. E-mail: {yliu, liuqun, sxlin}@ict.ac.cn.

Submission received: 13 September 2007; revised submission received: 7 January 2010; accepted for publication: 21 February 2010.

The Chinese word *Zhongguo* is aligned to the English word *China* because they are translations of one another. Similarly, the Chinese word *xin* is aligned to the English word *new*. These connections are not necessarily one-to-one. For example, one Chinese word *jianzhuye* corresponds to two English words *construction industry*. In addition, the English words (e.g., *opening ... to the outside*) connected to a Chinese word (e.g., *duiwaikaifang*) could be discontinuous. Figure 1 shows an alignment for this sentence pair. The Chinese and English words are listed horizontally and vertically, respectively. They are numbered to facilitate identification. The dark points indicate the correspondence between the words in two languages. The goal of word alignment is to identify such correspondences in a parallel text.

Word alignment plays an important role in many NLP tasks. In statistical machine translation, word-aligned corpora serve as an excellent source for translation-related knowledge. The estimation of translation model parameters usually relies heavily on word-aligned corpora, not only for phrase-based and hierarchical phrase-based models (Koehn, Och, and Marcu 2003; Och and Ney 2004; Chiang 2005, 2007), but also for syntax-based models (Quirk, Menezes, and Cherry 2005; Galley et al. 2006; Liu, Liu, and Lin 2006; Marcu et al. 2006). Besides machine translation, many applications for word-aligned corpora have been suggested, including machine-assisted translation,

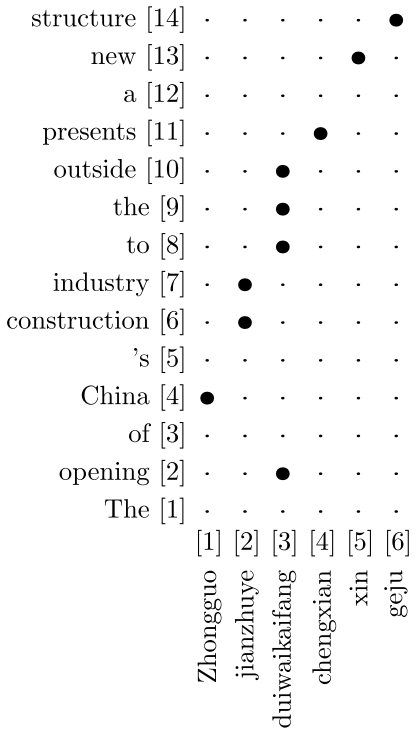

Figure 1
Example of a word alignment between a Chinese–English sentence pair. The Chinese and English words are listed horizontally and vertically, respectively. They are numbered to facilitate identification. The dark points indicate the correspondences between the words in the two languages.

translation assessment and critiquing tools, text generation, bilingual lexigraphy, and word sense disambiguation.

Various methods have been proposed for finding word alignments between parallel texts. Among them, generative alignment models (Brown et al. 1993; Vogel and Ney 1996) have been widely used to produce word alignments for large bilingual corpora. Describing the relationship of a bilingual sentence pair, a generative model treats word alignment as a hidden process and maximizes the likelihood of a training corpus using the expectation maximization (EM) algorithm. After the maximization process is complete, the unknown model parameters are determined and the word alignments are set to the maximum posterior predictions of the model.

However, one drawback of generative models is that they are hard to extend. Generative models usually impose strong independence assumptions between sub-models, making it very difficult to incorporate arbitrary features explicitly. For example, when considering whether to align two words, generative models cannot include information about lexical and syntactic features such as part of speech and orthographic similarity in an easy way. Such features would allow for more effective use of sparse data and result in a model that is more robust in the presence of unseen words. Extending a generative model requires that the interdependence of information sources be modeled explicitly, which often makes the resulting system quite complex.

In this article, we introduce a discriminative framework for word alignment based on the linear modeling approach. Within this framework, we treat all knowledge sources as feature functions that depend on a source sentence, a target sentence, and the alignment between them. Each feature function is associated with a feature weight. The linear combination of features gives an overall score to each candidate alignment. The best alignment is the one with the highest overall score. A linear model not only allows for easy integration of new features, but also admits optimizing feature weights directly with respect to evaluation metrics. Experimental results show that our approach improves both alignment quality and translation performance significantly.

This article is organized as follows. Section 2 gives a formal description of our model. We show how to train feature weights by taking evaluation metrics into account and how to find the most probable alignment in an exponential search space efficiently. Section 3 describes a number of features used in our experiments, focusing on the features that produce symmetric alignments. In Section 4, we evaluate our model in both alignment and translation tasks. Section 5 reviews previous work related to our approach and the article closes with a conclusion in Section 6.

2. Approach

2.1 The Model

Given a source language sentence $\mathbf{f} = f_1, \ldots, f_j, \ldots, f_J$ and a target language sentence $\mathbf{e} = e_1, \ldots, e_i, \ldots, e_I$, we define a **link** $l = (j, i)$ to exist if f_j and e_i are translations (or part of a translation) of one another. Then, an **alignment** is defined as a subset of the Cartesian product of the word positions:

$$\mathbf{a} \subseteq \{(j, i) : j = 1, \ldots, J; i = 1, \ldots, I\} \quad (1)$$

We propose a linear alignment model:

$$score(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{2}$$

where $h_m(\mathbf{f}, \mathbf{e}, \mathbf{a})$ is a **feature function** and λ_m is its associated **feature weight**. The linear combination of features gives an overall score $score(\mathbf{f}, \mathbf{e}, \mathbf{a})$ to each candidate alignment $\mathbf{a}$ for a given sentence pair $\langle \mathbf{f}, \mathbf{e} \rangle$.

2.2 Training

To achieve good alignment quality, it is essential to find a good set of feature weights λ_1^M. Before discussing how to train λ_1^M, we first describe two evaluation metrics that measure alignment quality, because we will optimize λ_1^M with respect to them directly.

2.2.1 Evaluation Metrics. The first metric is **alignment error rate** (AER), proposed by Och and Ney (2003). AER has been used as official evaluation criterion in most word alignment shared tasks. Och and Ney define two kinds of links in hand-aligned alignments: sure links for alignments that are unambiguous and possible links for ambiguous alignments. Sure links usually connect content words such as *Zhongguo* and *China*. In contrast, possible links often align words within idiomatic expressions and free translations.

An AER score is given by

$$\text{AER}(S, P, A) = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|} \tag{3}$$

where S is a set of sure links in a reference alignment that is hand-aligned by human experts, P is a set of possible links in the reference alignment, and A is a candidate alignment. Note that S is a subset of P: $S \subseteq P$. The lower the AER score is, the better the alignment quality is.

Although widely used, AER has been criticized for correlating poorly with translation quality (Ayan and Dorr 2006a; Fraser and Marcu 2007b). In other words, lower AER scores do not necessarily lead to better translation quality.[1] Fraser and Marcu (2007b) argue that reference alignments should consist of only sure links. They propose a new measure called the **balanced F-measure**:

$$\text{precision}(S, A) = \frac{|A \cap S|}{|S|} \tag{4}$$

$$\text{recall}(S, A) = \frac{|A \cap S|}{|A|} \tag{5}$$

$$\text{F-measure}(S, \alpha, A) = \frac{1}{\frac{\alpha}{\text{precision}(S,A)} + \frac{1-\alpha}{\text{recall}(S,A)}} \tag{6}$$

1 It has not yet been uniformly accepted that better word alignments yield better translations. Ayan and Dorr (2006a) present a detailed discussion of the impact of word alignment on statistical machine translation.

where α is a parameter that sets the trade-off between precision and recall. Higher F-measure means better alignment quality. Obviously, α less than 0.5 weights recall higher, whereas α greater than 0.5 weights precision higher.

We use both AER and F-measure in our experiments. AER is used in experiments evaluating alignment quality (Section 4.1) and F-measure is used in experiments evaluating translation performance (Section 4.2).

2.2.2 Minimum Error Rate Training. Suppose we have three candidate alignments: $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3$. Their AER scores are 0.21, 0.20, and 0.22, respectively. Therefore, $\mathbf{a}_2$ is the best candidate alignment, $\mathbf{a}_1$ is the second best, and $\mathbf{a}_3$ is the third best. We use three features to score each candidate. Table 1 lists the feature values for each candidate.

If the set of feature weights is $\{1.0, 1.0, 1.0\}$, the model scores (see Equation (2)) of the three candidates are -71, -74, and -76, respectively. Whereas reference alignment considers $\mathbf{a}_2$ as the best candidate, $\mathbf{a}_1$ has the maximal model score. This is undesirable because the model fails to agree with the reference. If we change the feature weights to $\{1.0, -2.0, 2.0\}$, the model scores become -73, -71, and -83, respectively. Now, the model chooses $\mathbf{a}_2$ as the best candidate correctly.

If a set of feature weights manages to make model predictions agree with reference alignments in training examples, we would expect the model to achieve good alignment quality on unseen data as well. To do this, we adopt the minimum error rate training (MERT) algorithm proposed by Och (2003) to find feature weights that minimize AER or maximize F-measure on a representative hand-aligned training corpus.

Given a reference alignment $\mathbf{r}$ and a candidate alignment $\mathbf{a}$, we use a loss function $E(\mathbf{r}, \mathbf{a})$ to measure alignment performance. Note that $E(\mathbf{r}, \mathbf{a})$ can be either AER or $1 -$ F-measure. Given a bilingual corpus $\langle \mathbf{f}_1^S, \mathbf{e}_1^S \rangle$ with a reference alignment $\mathbf{r}_s$ and a set of K different candidate alignments $\mathbf{C}_s = \{\mathbf{a}_{s,1} \ldots \mathbf{a}_{s,K}\}$ for each sentence pair $\langle \mathbf{f}_s, \mathbf{e}_s \rangle$, our goal is to find a set of feature weights $\hat{\lambda}_1^M$ that minimizes the overall loss on the training corpus:

$$\hat{\lambda}_1^M = \operatorname*{argmin}_{\lambda_1^M} \left\{ \sum_{s=1}^{S} E(\mathbf{r}_s, \hat{\mathbf{a}}(\mathbf{f}_s, \mathbf{e}_s; \lambda_1^M)) \right\} \tag{7}$$

$$= \operatorname*{argmin}_{\lambda_1^M} \left\{ \sum_{s=1}^{S} \sum_{k=1}^{K} E(\mathbf{r}_s, \mathbf{a}_{s,k}) \delta(\hat{\mathbf{a}}(\mathbf{f}_s, \mathbf{e}_s; \lambda_1^M), \mathbf{a}_{s,k}) \right\} \tag{8}$$

Table 1
Example feature values and alignment error rates.

	feature values			
candidate	h_1	h_2	h_3	AER
$\mathbf{a}_1$	–85	4	10	0.21
$\mathbf{a}_2$	–89	3	12	0.20
$\mathbf{a}_3$	–93	6	11	0.22

where $\hat{\mathbf{a}}(\mathbf{f}_s, \mathbf{e}_s; \lambda_1^M)$ is the best candidate alignment produced by the linear model:

$$\hat{\mathbf{a}}(\mathbf{f}_s, \mathbf{e}_s; \lambda_1^M) = \operatorname*{argmax}_{\mathbf{a}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}_s, \mathbf{e}_s, \mathbf{a}) \right\} \tag{9}$$

The basic idea of MERT is to optimize only one parameter (i.e., feature weight) each time and keep all other parameters fixed. This process runs iteratively over M parameters until the overall loss on the training corpus does not decrease.

Formally, suppose we tune a parameter and keep the other $M - 1$ parameters fixed; each candidate alignment corresponds to a line in the plane with γ as the independent variable:

$$\gamma \cdot \mu(\mathbf{f}, \mathbf{e}, \mathbf{a}) + \sigma(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{10}$$

where γ denotes the parameter being tuned (i.e., λ_m) and $\mu(\mathbf{f}, \mathbf{e}, \mathbf{a})$ and $\sigma(\mathbf{f}, \mathbf{e}, \mathbf{a})$ are constants with respect to γ:

$$\mu(\mathbf{f}, \mathbf{e}, \mathbf{a}) = h_m(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{11}$$

$$\sigma(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{m'=1, m' \neq m}^{M} \lambda_{m'} h_{m'}(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{12}$$

The set of candidates in $\mathbf{C}_s$ defines a set of lines. For example, given the candidate alignments in Table 1, suppose we only tune λ_2 and keep λ_1 and λ_3 fixed with an initial set of parameters $\{1.0, 1.0, 1.0\}$. According to Equation (10), $\mathbf{a}_1$ corresponds to a line $4\gamma - 75$, $\mathbf{a}_2$ corresponds to a line $3\gamma - 77$, and $\mathbf{a}_3$ corresponds to a line $6\gamma - 82$.

The decision rule in Equation (9) states that $\hat{\mathbf{a}}$ is the line with the highest model score for a given γ. The selection of γ for each sentence pair ultimately determines the loss at γ. How do we find values of γ that could generate different loss values?

As the loss can only change if we move to a γ where the highest line is different than before, Och (2003) suggests only evaluating the loss at values in between the intersections that line the top surface of the cluster of lines. Figure 2 demonstrates eight

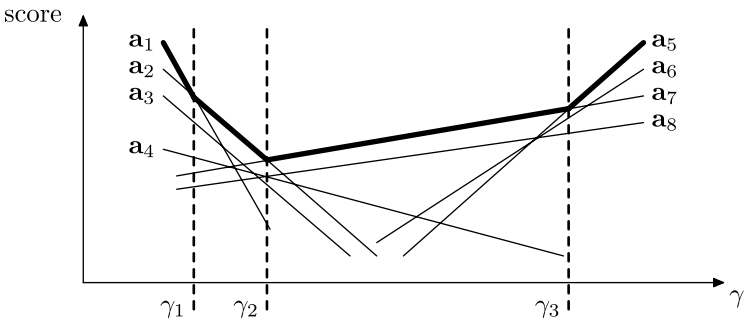

Figure 2
Candidate alignments in dimension γ and the critical intersections. Each candidate alignment is represented as a line. γ_1,γ_2, and γ_3 are critical intersections where the best candidate $\hat{\mathbf{a}}$ (highlighted in **bold**) will change: $\hat{\mathbf{a}}$ is $\mathbf{a}_1$ in $(-\infty, \gamma_1]$, $\mathbf{a}_2$ in $(\gamma_1, \gamma_2]$, $\mathbf{a}_7$ in $(\gamma_2, \gamma_3]$, and $\mathbf{a}_5$ in $(\gamma_3, +\infty)$.

candidate alignments. The sequence of the topmost line segments highlighted in bold constitutes an upper envelope, which indicates the best candidate alignments the model predicts with various values of γ. Instead of computing all possible K^2 intersections between the lines in $\mathbf{C}_s$, we just need to find the **critical intersections** where the topmost line changes. In Figure 2, γ_1, γ_2, and γ_3 are critical intersections. In the interval $(-\infty, \gamma_1]$, $\mathbf{a}_1$ has the highest score. Similarly, the best candidates are $\mathbf{a}_2$ for $(\gamma_1, \gamma_2]$, $\mathbf{a}_7$ for $(\gamma_2, \gamma_3]$, and $\mathbf{a}_5$ for $(\gamma_3, +\infty)$, respectively. The optimal $\hat{\gamma}$ can be found by collecting all critical intersections on the training corpus and choosing one γ that results in the minimal loss value. Please refer to Och (2003) for more details.

2.3 Search

Given a source language sentence $\mathbf{f}$ and a target language sentence $\mathbf{e}$, we try to find the best candidate alignment with the highest model score:

$$\hat{\mathbf{a}} = \operatorname*{argmax}_{\mathbf{a}} \left\{ score(\mathbf{f}, \mathbf{e}, \mathbf{a}) \right\} \tag{13}$$

$$= \operatorname*{argmax}_{\mathbf{a}} \left\{ \sum_{m=1}^{M} \lambda_m h_m(\mathbf{f}, \mathbf{e}, \mathbf{a}) \right\} \tag{14}$$

To do this, we begin with an empty alignment and keep adding new links until the model score of the current alignment does not increase. Figure 3 illustrates this search process. Given a source language sentence $f_1 f_2$ and a target language sentence $e_1 e_2$, the initial alignment $\mathbf{a}_1$ is empty (i.e., all words are unaligned). Then, we obtain a new alignment $\mathbf{a}_2$ by adding a link $(1, 1)$ to $\mathbf{a}_1$. Similarly, the addition of $(1, 2)$ to $\mathbf{a}_1$ leads to $\mathbf{a}_3$. $\mathbf{a}_2$ and $\mathbf{a}_3$ can be further extended to produce more alignments.

Graphically speaking, the search space of a sentence pair can be organized as a directed acyclic graph. Each node in the graph is a candidate alignment and each edge corresponds to a link. We say that alignments that have the same number of links constitute a **level**. There are $2^{J \times I}$ possible nodes and $J \times I + 1$ levels in a graph. In Figure 3, $\mathbf{a}_2$, $\mathbf{a}_3$, $\mathbf{a}_4$, and $\mathbf{a}_5$ belong to the same level because they all contain one link. The maximum level width is given by $\binom{J \times I}{\lfloor \frac{J \times I}{2} \rfloor}$. In Figure 3, the maximal level width is $\binom{4}{2} = 6$. Our goal is to find the node with the highest model score in a search graph.

As the search space of word alignment is exponential (although enumerable), it is computationally prohibitive to explore all the graph. Instead, we can search efficiently in a greedy way. In Figure 3, starting from $\mathbf{a}_1$, we add single links to $\mathbf{a}_1$ and obtain four new alignments: $\mathbf{a}_2$, $\mathbf{a}_3$, $\mathbf{a}_4$, and $\mathbf{a}_5$. We retain the best new alignment that has a higher score than $\mathbf{a}_1$, say $\mathbf{a}_3$, and discard the others. Then, we add single links to $\mathbf{a}_3$ and obtain three new alignments: $\mathbf{a}_7$, $\mathbf{a}_9$, and $\mathbf{a}_{11}$. After choosing $\mathbf{a}_9$ as the current best alignment, the next candidates are $\mathbf{a}_{12}$ and $\mathbf{a}_{14}$. Suppose the model scores of both $\mathbf{a}_{12}$ and $\mathbf{a}_{14}$ are lower than that of $\mathbf{a}_9$. We terminate the search process and choose $\mathbf{a}_9$ as the best candidate alignment.

During this search process, we expect that the addition of a single link l to the current best alignment $\mathbf{a}$ will result in a new alignment $\mathbf{a} \cup \{l\}$ with a higher score:

$$score(\mathbf{f}, \mathbf{e}, \mathbf{a} \cup \{l\}) > score(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{15}$$

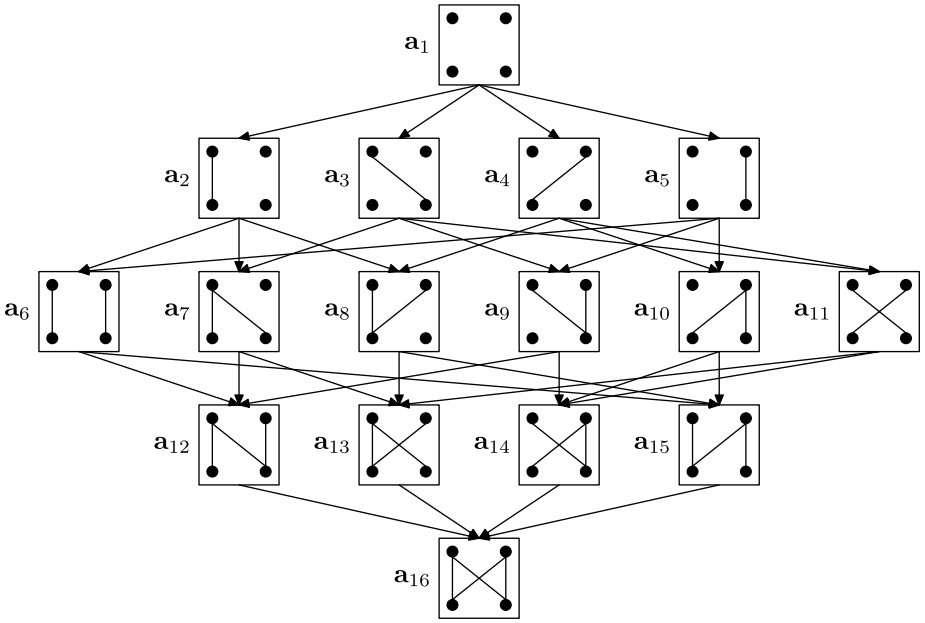

Figure 3
Search space of a sentence pair: $f_1 f_2$ and $e_1 e_2$. Each node in the directed graph is a candidate alignment and each edge denotes a transition between two nodes by adding a link.

that is

$$\sum_{m=1}^{M} \lambda_m \big(h_m(\mathbf{f}, \mathbf{e}, \mathbf{a} \cup \{l\}) - h_m(\mathbf{f}, \mathbf{e}, \mathbf{a}) \big) > 0 \tag{16}$$

As a result, we can remove most of the computational overhead by calculating only the difference of scores instead of the scores themselves. The difference of alignment scores with the addition of a link, which we refer to as a **link gain**, is defined as

$$\mathcal{G}(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) = \sum_{m=1}^{M} \lambda_m g_m(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) \tag{17}$$

where $g_m(\mathbf{f}, \mathbf{e}, \mathbf{a}, l)$ is a **feature gain**, which is the incremental feature value after adding a link l to the current alignment $\mathbf{a}$:

$$g_m(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) = h_m(\mathbf{f}, \mathbf{e}, \mathbf{a} \cup \{l\}) - h_m(\mathbf{f}, \mathbf{e}, \mathbf{a}) \tag{18}$$

In our experiments, we use a beam search algorithm that is more general than the above greedy algorithm. In the greedy algorithm, we retain at most one candidate in each level of the space graph while traversing top-down. In the beam search algorithm, we retain at most b candidates at each level.

Algorithm 1 shows the beam search algorithm. The input is a source language sentence $\mathbf{f}$ and a target language sentence $\mathbf{e}$ (line 1). The algorithm maintains a list of

Algorithm 1 A beam search algorithm for word alignment

```
 1: procedure ALIGN(f, e)
 2:     open ← ∅                                   ▷ a list of active alignments
 3:     N ← ∅                                      ▷ n-best list
 4:     a ← ∅                                      ▷ begin with an empty alignment
 5:     ADD(open, a, β, b)                         ▷ initialize the list
 6:     while open ≠ ∅ do
 7:         closed ← ∅                             ▷ a list of promising alignments
 8:         for all a ∈ open do
 9:             for all l ∈ J × I − a do           ▷ enumerate all possible new links
10:                 a' ← a ∪ {l}                   ▷ produce a new alignment
11:                 g ← GAIN(f, e, a, l)           ▷ compute the link gain
12:                 if g > 0 then                  ▷ ensure that the score will increase
13:                     ADD(closed, a', β, b)      ▷ update promising alignments
14:                 end if
15:                 ADD(N, a', 0, n)               ▷ update n-best list
16:             end for
17:         end for
18:         open ← closed                          ▷ update active alignments
19:     end while
20:     return N                                   ▷ return n-best list
21: end procedure
```

active alignments *open* (line 2) and an n-best list $\mathcal{N}$ (line 3). The aligning process begins with an empty alignment $\mathbf{a}$ (line 4) and the procedure ADD($open$, $\mathbf{a}$, β, b) adds $\mathbf{a}$ to *open*. The procedure prunes the search space by discarding any alignment that has a score worse than:

1. β multiplied with the best score in the list, or
2. the score of b-th best alignment in the list.

For each iteration (line 6), we use a list *closed* to store promising alignments that have higher scores than the current alignment. For every possible link l (line 9), we produce a new alignment $\mathbf{a}'$ (line 10) and calculate the link gain $\mathcal{G}$ by calling the procedure GAIN($\mathbf{f}$, $\mathbf{e}$, $\mathbf{a}$, l). If $\mathbf{a}'$ has a higher score (line 12), it is added to *closed* (line 13). We also update $\mathcal{N}$ to keep the top n alignments explored during the search (line 15). The n-best list will be used in training feature weights by MERT. This process iterates until there are no promising alignments. The theoretical running time of this algorithm is $\mathcal{O}(bJ^2I^2)$.

3. Feature Functions

The primary art in discriminative modeling is to define useful features that capture various characteristics of word alignments. Intuitively, we can include generative models such as the IBM Models 1–5 (Brown et al. 1993) as features in a discriminative model. A straightforward way is to use a generative model itself as a feature directly (Liu, Liu,

and Lin 2005). Another way is to treat each sub-model of a generative model as a feature (Fraser and Marcu 2006). In either case, a generative model can be regarded as a special case of a discriminative model where all feature weights are one. A detailed discussion of the treatment of the IBM models as features can be found in Appendix B.

One major drawback of the IBM models is asymmetry. They are restricted such that each source word is assigned to exactly one target word. This is not the case for many language pairs. For example, in our running example, one Chinese word *jianzhuye* corresponds to two English words *construction industry*. As a result, our linear model will produce only one-to-one alignments if the IBM models in two translation directions (i.e., source-to-target and target-to-source) are both used. Although some authors would use the one-to-one assumption to simplify the modeling problem (Melamed 2000; Taskar, Lacoste-Julien, and Klein 2005), many translation phenomena cannot be handled and the recall cannot reach 100% in principle.

A more general way is to model alignment as an arbitrary relation between source and target language word positions. As our linear model is capable of including many overlapping features regardless of their interdependencies, it is easy to add features that characterize symmetric alignments. In the following subsections, we will introduce a number of symmetric features used in our experiments.

3.1 Translation Probability Product

To determine the correspondence of words in two languages, word-to-word translation probabilities are always the most important knowledge source. To model a symmetric alignment, a straightforward way is to compute the product of the translation probabilities of each link in two directions.

For example, suppose that there is an alignment $\{(1,2)\}$ for a source language sentence $f_1 f_2$ and a target language sentence $e_1 e_2$; the translation probability product is

$$t(e_2|f_1) \times t(f_1|e_2)$$

where $t(e|f)$ is the probability that f is translated to e and $t(f|e)$ is the probability that e is translated to f, respectively.

Unfortunately, the underlying model is biased: The more links added, the smaller the product will be. For example, if we add a link $(2,2)$ to the current alignment and obtain a new alignment $\{(1,2),(2,2)\}$, the resulting product will decrease after being multiplied with $t(e_2|f_2) \times t(f_2|e_2)$:

$$t(e_2|f_1) \times t(f_1|e_2) \times t(e_2|f_2) \times t(f_2|e_2)$$

The problem results from the absence of empty **cepts**. Following Brown et al. (1993), a cept in an alignment is either a single source word or it is empty. They assign cepts to positions in the source sentence and reserve position zero for the empty cept. All unaligned target words are assumed to be "aligned" to the empty cept. For example, in the current example alignment $\{(1,2)\}$, the unaligned target word e_1 is said to be "aligned" to the empty cept f_0. As our model is symmetric, we use f_0 to denote the empty cept on the source side and use e_0 to denote the empty cept on the target side, respectively.

If we take empty cepts into account, the product for $\{(1,2)\}$ can be rewritten as

$$t(e_2|f_1) \times t(f_1|e_2) \times t(e_1|f_0) \times t(f_2|e_0)$$

Similarly, the product for $\{(1,2),(2,2)\}$ now becomes

$$t(e_2|f_1) \times t(f_1|e_2) \times t(e_2|f_2) \times t(f_2|e_2) \times t(e_1|f_0)$$

Note that after adding the link $(2,2)$, the new product still has more factors than the old product. However, the new product is not necessarily always smaller than the old one. In this case, the new product divided by the old product is

$$\frac{t(e_2|f_2) \times t(f_2|e_2)}{t(f_2|e_0)}$$

Whether a new product increases or not depends on actual translation probabilities.[2]

Depending on whether they are aligned or not, we divide the words in a sentence pair into two categories: **aligned** and **unaligned**. For each aligned word, we use translation probabilities conditioned on its counterpart in two directions (i.e., $t(e_i|f_j)$ and $t(f_j|e_i)$). For each unaligned word, we use translation probabilities conditioned on empty cepts on the other side in two directions (i.e., $t(e_i|f_0)$ and $t(f_j|e_0)$).

Formally, the feature function for translation probability product is given by[3]

$$\begin{aligned} h_{tpp}(\mathbf{f},\mathbf{e},\mathbf{a}) = & \sum_{(j,i)\in\mathbf{a}} \big(\log\big(t(e_i|f_j)\big) + \log\big(t(f_j|e_i)\big)\big) + \\ & \sum_{j=1}^{J} \log\big(\delta(\psi_j,0) \times t(f_j|e_0) + 1 - \delta(\psi_j,0)\big) + \\ & \sum_{i=1}^{I} \log\big(\delta(\phi_i,0) \times t(e_i|f_0) + 1 - \delta(\phi_i,0)\big) \end{aligned} \tag{19}$$

where $\delta(x,y)$ is the Kronecker function, which is 1 if $x = y$ and 0 otherwise. We define the **fertility** of a source word f_j as the number of aligned target words:

$$\psi_j = \sum_{(j',i)\in\mathbf{a}} \delta(j',j) \tag{20}$$

2 Even though we take empty cepts into account, the bias problem still exists because the product will decrease by adding new links if there are no unaligned words. For example, the product will go down if we further add a link $(1,1)$ to $\{(1,2),(2,2)\}$ as all source words are aligned. This might not be a bad bias because reference alignments usually do not have all words aligned and contain too many links. Although translation probability product is degenerate as a generative model, the bias problem can be alleviated when this feature is combined with other features such as link count (see Section 3.8).

3 We use the logarithmic form of translation probability product to avoid manipulating very small numbers (e.g., $4.3 \times e^{-100}$) just for practical reasons.

Table 2
Calculating feature values of translation probability product for a source sentence $f_1 f_2$ and a target sentence $e_1 e_2$.

alignment	feature value
$\{\}$	$\log\big(t(e_1\|f_0)\cdot t(e_2\|f_0)\cdot t(f_1\|e_0)\cdot t(f_2\|e_0)\big)$
$\{(1,2)\}$	$\log\big(t(e_1\|f_0)\cdot t(e_2\|f_1)\cdot t(f_1\|e_2)\cdot t(f_2\|e_0)\big)$
$\{(1,2),(2,2)\}$	$\log\big(t(e_1\|f_0)\cdot t(e_2\|f_1)\cdot t(e_2\|f_2)\cdot t(f_1\|e_2)\cdot t(f_2\|e_2)\big)$

Similarly, the fertility of a target word e_i is the number of aligned source words:

$$\phi_i = \sum_{(j,i')\in\mathbf{a}} \delta(i', i) \tag{21}$$

For example, as only one English word *China* is aligned to the first Chinese word *Zhongguo* in Figure 1, the fertility of *Zhongguo* is $\psi_1 = 1$. Similarly, the fertility of the third Chinese word *duiwaikaifang* is $\psi_3 = 4$ because there are four aligned English words. The fertility of the first English word *The* is $\phi_1 = 0$. Obviously, the words with zero fertilities (e.g., *The*, *'s*, and *a* in Figure 1) are unaligned.

In Equation (19), the first term calculates the product of aligned words, the second term deals with unaligned source words, and the third term deals with unaligned target words. Table 2 shows the feature values for some word alignments.

For efficiency, we need to calculate the difference of feature values instead of the values themselves, which we call feature gain (see Equation (18)). The feature gain for translation probability product is[4]

$$\begin{aligned} g_{tpp}(\mathbf{f},\mathbf{e},\mathbf{a},j,i) = {} & \log\big(t(e_i|f_j)\big) + \log\big(t(f_j|e_i)\big) - \\ & \log\big(\delta(\psi_j,0)\times t(f_j|e_0) + 1 - \delta(\psi_j,0)\big) - \\ & \log\big(\delta(\phi_i,0)\times t(e_i|f_0) + 1 - \delta(\phi_i,0)\big) \end{aligned} \tag{22}$$

where ψ_j and ϕ_i are the fertilities before adding the link (j,i).

Although this feature is symmetric, we obtain the translation probabilities $t(f|e)$ and $t(e|f)$ by training the IBM models using GIZA++ (Och and Ney 2003).

3.2 Exact Match

Motivated by the fact that proper names (e.g., *IBM*) or specialized terms (e.g., *DNA*) are often the same in both languages, Taskar, Lacoste-Julien, and Klein (2005) use a feature that sums up the number of words linked to identical words. We adopt this exact match feature in our model:

$$h_{em}(\mathbf{f},\mathbf{e},\mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} \delta(f_j, e_i) \tag{23}$$

4 For clarity, we use $g_{tpp}(\mathbf{f},\mathbf{e},\mathbf{a},j,i)$ instead of $g_{tpp}(\mathbf{f},\mathbf{e},\mathbf{a},l)$ because j and i appear in the equation.

$$g_{em}(\mathbf{f}, \mathbf{e}, \mathbf{a}, j, i) = \delta(f_j, e_i) \tag{24}$$

3.3 Cross Count

Due to the diversity of natural languages, word orders between two languages are usually different. For example, subject-verb-object (SVO) languages such as Chinese and English often put an object after a verb while subject-object-verb (SOV) languages such as Japanese and Turkish often put an object before a verb. Even between SVO languages such as Chinese and English, word orders could be quite different too. In Figure 1, while *Zhongguo* is the first Chinese word, its counterpart *China* is the fourth English word. Meanwhile, the third Chinese word *duiwaikaifang* after *Zhongguo* is aligned to the second English word *opening* before *China*. We say that there is a **cross** between the two links $(1, 4)$ and $(3, 2)$ because $(1 - 3) \times (4 - 2) < 0$. In Figure 1, there is only one cross. As a result, we could use the number of crosses in alignments to capture the divergence of word orders between two languages.

Formally, the cross count feature function is given by

$$h_{cc}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} \sum_{(j',i')\in\mathbf{a}} [\![(j - j') \times (i - i') < 0]\!] \tag{25}$$

$$g_{cc}(\mathbf{f}, \mathbf{e}, \mathbf{a}, j, i) = \sum_{(j',i')\in\mathbf{a}} [\![(j - j') \times (i - i') < 0]\!] \tag{26}$$

where $[\![expr]\!]$ is an indicator function that takes a boolean expression *expr* as the argument:

$$[\![expr]\!] = \begin{cases} 1 \text{ if } expr \text{ is true} \\ 0 \text{ otherwise} \end{cases} \tag{27}$$

3.4 Neighbor Count

Moore (2005) finds that word alignments between closely related languages tend to be approximately monotonic. Even for distantly related languages, the number of crossing links is far less than chance since phrases tend to be translated as contiguous chunks. In Figure 1, the dark points are positioned approximately in parallel with the diagonal line, indicating that the alignment is approximately monotonic.

To capture such monotonicity, we follow Lacoste-Julien et al. (2006) to encourage strictly monotonic alignments by adding a bonus for any pair of links (j, i) and (j', i') such that

$$j - j' = 1 \wedge i - i' = 1$$

In Figure 1, there is one such link pair: $(3, 10)$ and $(4, 11)$. We call these links **neighbors**. Similarly, $(5, 13)$ and $(6, 14)$ are also neighbors.

Formally, the neighbor count feature function is given by

$$h_{nc}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} \sum_{(j',i')\in\mathbf{a}} [\![j - j' = 1 \wedge i - i' = 1]\!] \tag{28}$$

$$g_{nc}(\mathbf{f}, \mathbf{e}, \mathbf{a}, j, i) = \sum_{(j',i') \in \mathbf{a}} [\![j - j' = 1 \wedge i - i' = 1]\!] \tag{29}$$

3.5 Fertility Probability Product

Casual inspection of some word alignments quickly establishes that some Chinese words such as *Zhongguo* and *chengxian* are often aligned to one English word whereas other Chinese words such as *duiwaikaifang* tend to be translated into multiple English words. Brown et al. (1993) call the number of target words to which a source word f is connected the fertility of f. Recall that we have given the formal definition of fertility in the symmetric scenario in Equation (20) and Equation (21).

Besides word association (Sections 3.1 and 3.2) and word distortion (Sections 3.3 and 3.4), fertility also proves to be very important in modeling alignment because sophisticated generative models such as the IBM Models 3–5 parameterize fertilities directly. As our goal is to produce symmetric alignments, we calculate the product of fertility probabilities in two directions.

Given an alignment $\{(1,2)\}$ for a source sentence $f_1 f_2$ and a target sentence $e_1 e_2$, the fertility probability product is

$$n(1|f_0) \times n(1|f_1) \times n(0|f_2) \times n(1|e_0) \times n(0|e_1) \times n(1|e_2)$$

where $n(\psi_j|f_j)$ is the probability that f_j has a fertility of ψ_j and $n(\phi_i|e_i)$ is the probability that e_i has a fertility of ϕ_i, respectively.[5] For example, $n(1|f_0)$ denotes the probability that one target word is "aligned" to the source empty cept f_0 and $n(1|e_2)$ denotes the probability that one source word is aligned to e_2.

If we add a link $(2, 2)$ to the current alignment and obtain a new alignment $\{(1, 2), (2, 2)\}$, the resulting product will be

$$n(1|f_0) \times n(1|f_1) \times n(1|f_2) \times n(0|e_0) \times n(0|e_1) \times n(2|e_2)$$

The new product divided by the old product is

$$\frac{n(1|f_2) \times n(0|e_0) \times n(2|e_2)}{n(0|f_2) \times n(1|e_0) \times n(1|e_2)}$$

Formally, the feature function for fertility probability product is given by

$$h_{fpp}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{j=0}^{J} \log(n(\psi_j|f_j)) + \sum_{i=0}^{I} \log(n(\phi_i|e_i)) \tag{30}$$

5 Brown et al. (1993) treat the empty cept in a different way. They assume that at most half of the source words in an alignment are not aligned (i.e., $\phi_0 \leq J/2$) and define a binomial distribution relying on an auxiliary parameter p_0. Here, we use $n(\phi_0|e_0)$ instead of the original form $n_0(\phi_0|\sum_{i=1}^{I} \phi_i)$ just for simplicity. See Appendix B for more details.

The corresponding feature gain is

$$\begin{aligned} g_{fpp}(\mathbf{f},\mathbf{e},\mathbf{a},j,i) = & \log(n(\psi_0 - \delta(\phi_i,0)|f_0)) - \log(n(\psi_0|f_0)) + \\ & \log(n(\psi_j+1|f_j) - \log(n(\psi_j|f_j)) + \\ & \log(n(\phi_0 - \delta(\psi_j,0)|e_0)) - \log(n(\phi_0|e_0)) + \\ & \log(n(\phi_i+1|e_i)) - \log(n(\phi_i|e_i)) \end{aligned} \tag{31}$$

where ψ_j and ϕ_i are the fertilities before adding the link (j,i).

Table 3 gives the feature values for some word alignments. In practice, we also obtain all fertility probabilities $n(\psi_j|f_j)$ and $n(\phi_i|e_i)$ by using the output of GIZA++ directly.

3.6 Linked Word Count

We observe that there should not be too many unaligned words in good alignments. For example, there are only three unaligned words on the target side in Figure 1: *The*, *'s*, and *a*. Unaligned words are usually function words that have little lexical meaning but instead serve to express grammatical relationships with other words or specify the attitude or mood of the speaker. To control the number of unaligned words, we follow Moore, Yih, and Bode (2006) to introduce a linked word count feature that simply counts the number of aligned words:

$$h_{lwc}(\mathbf{f},\mathbf{e},\mathbf{a}) = \sum_{j=1}^{J} [\![\psi_j > 0]\!] + \sum_{i=1}^{I} [\![\phi_i > 0]\!] \tag{32}$$

$$g_{lwc}(\mathbf{f},\mathbf{e},\mathbf{a},j,i) = \delta(\psi_j,0) + \delta(\phi_i,0) \tag{33}$$

In Equation (33), ψ_j and ϕ_i are the fertilities before adding l.

3.7 Sibling Distance

In word alignments, there are usually several words connected to the same word on the other side. For example, in Figure 1, two English words *construction* and *industry* are aligned to one Chinese word *jianzhuye*. We call the words aligned to the same word on the other side **siblings**. In Figure 1, *opening*, *to*, *the*, and *outside* are also siblings because they are aligned to *duiwaikaifang*. A word (e.g., *jianzhuye*) often tends to produce a series of words in another language that belong together, whereas others (e.g., *duiwaikaifang*)

Table 3
Calculating feature values of fertility probability product for a source sentence $f_1 f_2$ and a target sentence $e_1 e_2$.

alignment	feature value
$\{\}$	$\log(n(2\|f_0)\cdot n(0\|f_1)\cdot n(0\|f_2)\cdot n(2\|e_0)\cdot n(0\|e_1)\cdot n(0\|e_2))$
$\{(1,2)\}$	$\log(n(1\|f_0)\cdot n(1\|f_1)\cdot n(0\|f_2)\cdot n(1\|e_0)\cdot n(0\|e_1)\cdot n(1\|e_2))$
$\{(1,2),(2,2)\}$	$\log(n(1\|f_0)\cdot n(1\|f_1)\cdot n(1\|f_2)\cdot n(0\|e_0)\cdot n(0\|e_1)\cdot n(2\|e_2))$

tend to produce a series of words that should be separate. To model this tendency, we introduce a feature that sums up the distances between siblings.

Formally, we use $\omega_{j,k}$ to denote the position of the k-th target word aligned to a source word f_j and use $\pi_{i,k}$ to denote the position of the k-th source word aligned to a target word e_i. For example, *jianzhuye* is the second source word (i.e., f_2) in Figure 1. As the first target word aligned to f_2 is *construction* (i.e., e_6), therefore we say that $\omega_{2,1} = 6$. Similarly, $\omega_{2,2} = 7$ because *industry* (i.e., e_7) is the second target word aligned to *jianzhuye*. Obviously, $\omega_{j,k+1}$ is always greater than $\omega_{j,k}$ by definition.

As *construction* and *industry* are siblings, we define the distance between them as $\omega_{2,2} - \omega_{2,1} - 1 = 0$. Note that we give no penalty to siblings that belong closely together. In Figure 1, there are four siblings *opening*, *to*, *the*, and *outside* aligned to the source word *duiwaikaifang*. The sum of distances between them is calculated as

$$\begin{aligned} & \omega_{3,2} - \omega_{3,1} - 1 + \omega_{3,3} - \omega_{3,2} - 1 + \omega_{3,4} - \omega_{3,3} - 1 \\ = \; & \omega_{3,4} - \omega_{3,1} - 3 \\ = \; & 10 - 2 - 3 \\ = \; & 5 \end{aligned}$$

Therefore, the distance sum of f_j can be efficiently calculated as

$$\Delta(j, \psi_j) = \begin{cases} \omega_{j,\psi_j} - \omega_{j,1} - \psi_j + 1 & \text{if } \psi_j > 1 \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

Accordingly, the distance sum of e_i is

$$\nabla(i, \phi_i) = \begin{cases} \pi_{i,\phi_i} - \pi_{i,1} - \phi_i + 1 & \text{if } \phi_i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Formally, the feature function for sibling distance is given by

$$h_{sd}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{j=1}^{J} \Delta(j, \psi_j) + \sum_{i=1}^{I} \nabla(i, \phi_i) \tag{36}$$

The corresponding feature gain is

$$\begin{aligned} g_{sd}(\mathbf{f}, \mathbf{e}, \mathbf{a}, j, i) = \; & \Delta(j, \psi_j + 1) - \Delta(j, \psi_j) + \\ & \nabla(i, \phi_i + 1) - \nabla(i, \phi_i) \end{aligned} \tag{37}$$

where ψ_j and ϕ_i are the fertilities before adding the link (j, i).

3.8 Link Count

Given a source sentence with J words and a target sentence with I words, there are $J \times I$ possible links. However, the actual number of links in a reference alignment is usually far less. For example, there are only 10 links in Figure 1 although the maximum is $6 \times 14 = 84$. The number of links has an important effect on alignment quality because

more links result in higher recall while fewer links result in higher precision. A good trade-off between recall and precision usually results from a reasonable number of links. Using the number of links as a feature could also alleviate the bias problem posed by the translation probability product feature (see Section 3.1). A negative weight of the link count feature often leads to fewer links while a positive weight favors more links.

Formally, the feature function for link count is

$$h_{lc}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = |\mathbf{a}| \tag{38}$$

$$g_{lc}(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) = 1 \tag{39}$$

where $|\mathbf{a}|$ is the cardinality of $\mathbf{a}$ (i.e., the number of links in $\mathbf{a}$).

3.9 Link Type Count

Due to the different fertilities of words, there are different types of links. For instance, one-to-one links indicate that one source word (e.g., *Zhongguo*) is translated into exactly one target word (e.g., *China*) while many-to-many links exist for phrase-to-phrase translation. The distribution of link types differs for different language pairs. For example, one-to-one links occur more frequently in closely related language pairs (e.g., French–English) and one-to-many links are more common in distantly related language pairs (e.g., Chinese–English). To capture the distribution of link types independent of languages, we use features to count different types of links.

Following Moore (2005), we divide links in an alignment into four categories:

1. **one-to-one** links, in which neither the source nor the target word participates in other links;
2. **one-to-many** links, in which only the source word participates in other links;
3. **many-to-one** links, in which only the target word participates in other links;
4. **many-to-many** links, in which both the source and target words participate in other links.

In Figure 1, $(1, 4)$, $(4, 11)$, $(5, 13)$, and $(6, 14)$ are one-to-one links and the others are one-to-many links.

As a result, we introduce four features:

$$h_{o2o}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} [\![\psi_j = 1 \wedge \phi_i = 1]\!] \tag{40}$$

$$h_{o2m}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} [\![\psi_j > 1 \wedge \phi_i = 1]\!] \tag{41}$$

$$h_{m2o}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} [\![\psi_j = 1 \wedge \phi_i > 1]\!] \tag{42}$$

$$h_{m2m}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \sum_{(j,i)\in\mathbf{a}} [\![\psi_j > 1 \wedge \phi_i > 1]\!] \tag{43}$$

Their feature gains cannot be calculated in a straightforward way because the addition of a link might change the link types of its siblings on both the source and target sides. For example, if we align the Chinese word *chengxian* and the English word *industry*, the newly added link $(4, 7)$ is a many-to-many link. Its source sibling $(2, 7)$, which was a one-to-many link, now becomes a many-to-many link. Meanwhile, its target sibling $(4, 11)$, which was a one-to-one link, now becomes a one-to-many link.

Algorithm 2 shows how to calculate the four feature gains. After initialization (line 2), we first decide the type of l (lines 3–11). Then, we consider the siblings of l on the target side (lines 12–24) and those on the source side (lines 25–38), respectively. Note that the feature gains of siblings will not change if $\psi_i \neq 1$ or $\phi_j \neq 1$.

3.10 Bilingual Dictionary

A conventional bilingual dictionary can be considered an additional knowledge source. The intuition is that a dictionary is expected to be more reliable than an automatically trained lexicon. For example, if *Zhongguo* and *China* appear in an entry of a dictionary, they should be more likely to be aligned. Thus, we use a single indicator feature to encourage linking word pairs that occur in a dictionary $\mathbf{D}$:

$$h_{bd}(\mathbf{f}, \mathbf{e}, \mathbf{a}, \mathbf{D}) = \sum_{(j,i) \in \mathbf{a}} [\![(f_j, e_i) \in \mathbf{D}]\!] \tag{44}$$

$$g_{bd}(\mathbf{f}, \mathbf{e}, \mathbf{a}, \mathbf{D}, j, i) = [\![(f_j, e_i) \in \mathbf{D}]\!] \tag{45}$$

3.11 Link Co-Occurrence Count

The system combination technique that integrates predictions from multiple systems proves to be effective in machine translation (Rosti, Matsoukas, and Schwartz 2007; He et al. 2008). In word alignment, a link should be aligned if it appears in most system predictions. Taskar, Lacoste-Julien, and Klein (2005) include the IBM Model 4 predictions as features and obtain substantial improvements.

To enable system combination, we design a feature to favor links voted by most systems. Given an alignment $\mathbf{a}'$ produced by another system, we use the number of links of the intersection of $\mathbf{a}$ and $\mathbf{a}'$ as a feature:

$$h_{lcc}(\mathbf{f}, \mathbf{e}, \mathbf{a}, \mathbf{a}') = |\mathbf{a} \cap \mathbf{a}'| \tag{46}$$

$$g_{lcc}(\mathbf{f}, \mathbf{e}, \mathbf{a}, \mathbf{a}', j, i) = [\![l \in \mathbf{a} \cap \mathbf{a}']\!] \tag{47}$$

4. Experiments

In this section, we try to answer two questions:

1. Does the proposed approach achieve higher alignment quality than generative alignment models?
2. Do statistical machine translation systems produce better translations if we replace generative alignment models with the proposed approach?

In Section 4.1, we evaluate our approach on three word alignment shared tasks for five language pairs with varying divergence and richness of resources. Experimental

Algorithm 2 Calculating gains for the link type count features

procedure GAINLINKTYPECOUNT($\mathbf{f}, \mathbf{e}, \mathbf{a}, j, i$)
  $\{g_{o2o}, g_{o2m}, g_{m2o}, g_{m2m}\} \leftarrow \{0, 0, 0, 0\}$ ▷ initialize the feature gains
  if $\psi_j = 0 \wedge \phi_i = 0$ **then** ▷ consider (j, i) first
    $g_{o2o} \leftarrow g_{o2o} + 1$
  else if $\psi_j > 0 \wedge \phi_i = 0$ **then**
    $g_{o2m} \leftarrow g_{o2m} + 1$
  else if $\psi_j = 0 \wedge \phi_i > 0$ **then**
    $g_{m2o} \leftarrow g_{m2o} + 1$
  else
    $g_{m2m} \leftarrow g_{m2m} + 1$
  end if
  if $\psi_j = 1$ **then** ▷ consider the siblings of (j, i) on the target side
    for $i' = 1 \ldots I$ **do**
      if $(j, i') \in \mathbf{a} \wedge i' \neq i$ **then** ▷ (j, i') is a sibling of (j, i) on the target side
        if $\phi_{i'} = 1$ **then** ▷ (j, i') was a one-to-one link
          $g_{o2o} \leftarrow g_{o2o} - 1$
          $g_{o2m} \leftarrow g_{o2m} + 1$ ▷ (j, i') now becomes a one-to-many link
        else ▷ (j, i') was a many-to-one link
          $g_{m2o} \leftarrow g_{m2o} - 1$
          $g_{m2m} \leftarrow g_{m2m} + 1$ ▷ (j, i') now becomes a many-to-many link
        end if
      end if
    end for
  end if
  if $\phi_i = 1$ **then** ▷ consider the siblings of (j, i) on the source side
    for $j' = 1 \ldots J$ **do**
      if $(j', i) \in \mathbf{a} \wedge j' \neq j$ **then** ▷ (j', i) is a sibling of (j, i) on the source side
        if $\psi_{j'} = 1$ **then** ▷ (j', i) was a one-to-one link
          $g_{o2o} \leftarrow g_{o2o} - 1$
          $g_{m2o} \leftarrow g_{m2o} + 1$ ▷ (j', i) now becomes a many-to-one link
        else ▷ (j', i) was a one-to-many link
          $g_{o2m} \leftarrow g_{o2m} - 1$
          $g_{m2m} \leftarrow g_{m2m} + 1$ ▷ (j', i) now becomes a many-to-many link
        end if
      end if
    end for
  end if
  return $\{g_{o2o}, g_{o2m}, g_{m2o}, g_{m2m}\}$ ▷ return the four feature gains
end procedure

results show that our system outperforms systems participating in the three shared tasks significantly and achieves comparable results with other state-of-the-art discriminative alignment models.

In Section 4.2, we investigate the effect of our model on translation quality. By training feature weights with respect to F-measure instead of AER, our model results in superior translation quality over generative methods for phrase-based, hierarchical phrase-based, and tree-to-string SMT systems.

4.1 Evaluation of Alignment Quality

In this section, we present results of experiments on three word alignment shared tasks:

1. HLT/NAACL 2003 shared task (Mihalcea and Pedersen 2003). As part of the HLT/NAACL 2003 workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond," this shared task includes two language pairs: English–French and Romanian–English. Participants can use both limited and unlimited resources.
2. ACL 2005 shared task (Martin, Mihalcea, and Pedersen 2005). As part of the ACL 2005 workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond," this shared task includes three language pairs to cover different language and data characteristics: English–Inuktitut, Romanian–English, and English–Hindi. Participants can use both limited and unlimited resources.
3. HTRDP 2005 shared task. As part of the 2005 HTRDP (National High Technology Research and Development Program of China, also called "863" Program) Evaluation on Chinese Information Processing and Intelligent Human-Machine Interface Technology, this shared task included only one language pair: Chinese–English. Participants can use unlimited resources.

Among these, we choose two tasks, English–French and Chinese–English, to report detailed experimental results. Results for the other tasks can also be found in Table 11.

Corpus statistics for the English–French and Chinese–English tasks are shown in Tables 4 and 5. The English–French data from the HLT/NAACL 2003 shared task consist of a training corpus of 1,130,104 sentence pairs, a development corpus of 37 sentence pairs, and a test corpus of 447 sentence pairs. The development and test sets are manually aligned and marked with both sure and possible labels. Although the Canadian Hansard bilingual corpus is widely used in the community, direct comparisons are difficult due to the differences in splitting of training data, development data, and test data. To make our results more comparable to previous work, we followed Lacoste-

Table 4
Corpus characteristics of the English–French task.

		English	French
Training corpus	Sentences	1,130,104	
	Words	20.01M	23.61M
	Vocabulary	68,019	86,591
Development corpus	Sentences	37	
	Words	661	721
	Vocabulary	322	344
Test corpus	Sentences	447	
	Words	7,020	7,761
	Vocabulary	1,732	1,943

Table 5
Corpus characteristics of the Chinese–English task.

		Chinese	English
Training corpus	Sentences	837,594	
	Words	10.32M	10.71M
	Vocabulary	93,532	134,143
Development corpus	Sentences	502	
	Words	9,338	9,364
	Vocabulary	2,608	2,587
Test corpus	Sentences	505	
	Words	9,088	10,224
	Vocabulary	2,319	2,651

Julien et al. (2006) splitting the original test set into two parts: the first 200 sentences as the development set and the remaining 247 sentences as the test set. To compare with systems participating in the 2003 NAACL shared task, we also used the small development set of 37 sentences to optimize feature weights, and ran our system on the original test set of 447 sentences. The results are shown in Table 11.

The Chinese–English data from the HTRDP 2005 shared task contains a development corpus of 502 sentence pairs and a test corpus of 505 sentence pairs. We use a training corpus of 837,594 sentence pairs available from Chinese Linguistic Data Consortium and a bilingual dictionary containing 415,753 entries.

4.1.1 Comparison of the Search Algorithm with GIZA++. We develop a word alignment system named *Vigne* based on the linear modeling approach. As we mentioned before, our model can include the IBM models as features (see Appendix B). To investigate the effectiveness of our search algorithm, we compare Vigne with GIZA++ by using the same models.

Table 6 shows the alignment error rate percentages for various IBM models in GIZA++ and Vigne. To make the results comparable, we ensured that Vigne shared

Table 6
Comparison of AER scores for various IBM models in GIZA++ and Vigne. These models are trained only on development and test sets. The pruning setting for Vigne is $\beta = 0$ and $b = 1$. All differences are not statistically significant.

			English–French		Chinese–English	
Model	Training Scheme	Direction	GIZA++	Vigne	GIZA++	Vigne
Model 1	1^5	$S \rightarrow T$	50.6	50.6	58.0	58.0
		$T \rightarrow S$	46.2	46.2	56.1	56.1
Model 2	$1^5 2^5$	$S \rightarrow T$	47.8	47.8	59.3	59.3
		$T \rightarrow S$	43.6	43.6	57.4	57.4
Model 3	$1^5 H^5 3^3$	$S \rightarrow T$	31.6	31.4	45.0	44.5
		$T \rightarrow S$	27.9	27.9	47.4	46.5
Model 4	$1^5 H^5 3^3 4^3$	$S \rightarrow T$	34.5	34.2	44.9	44.6
		$T \rightarrow S$	30.8	30.6	46.7	46.4

the same parameters with GIZA++.[6] Table 6 also gives the training schemes used for GIZA++. For example, the training scheme for Model 4 is $1^5H^53^34^3$. This notation indicates that five iterations of Model 1, five iterations of HMM, three iterations of Model 3, and three iterations of Model 4 are performed. As the two systems use the same model parameters, the amount of training data will have no effect on the comparison. Therefore, we trained the IBM Models only on the development and test sets. As a result, the AER scores in Table 6 look quite high.

In GIZA++, there exist simple polynomial algorithms to find the Viterbi alignments for Models 1 and 2. We observe that the greedy search algorithm ($\beta = 0$ and $b = 1$) used by Vigne can also find the optimal alignments. Note that the two systems achieve identical AER scores because there are no search errors.

For Models 3 and 4, maximization over all alignments cannot be efficiently carried out as the corresponding search problem is NP-complete. To alleviate the problem, GIZA++ resorts to a greedy search algorithm. The basic idea is to compute a Viterbi alignment of a simple model such as Model 2 or HMM. This alignment (an intermediate node in the search space) is then iteratively improved with respect to the alignment probability of the refined model by moving or swapping links. In contrast, our search algorithm starts from an empty alignment and has only one operation: adding a link. In addition, we treat the fertility probability of an empty cept in a different way (see Equation B.7). Interestingly, Vigne achieves slightly better results than GIZA++ for both models. All differences are not statistically significant.

4.1.2 Comparison to Generative Models Using Asymmetric Features. Table 7 compares the AER scores achieved by GIZA++, Cross-EM (Liang, Taskar, and Klein 2006), and Vigne. On both tasks, we lowercased all English words in the training, development, and test sets as a preprocessing step. For GIZA++, we used the default training scheme of $1^5H^53^54^5$. We used the three symmetrization heuristics proposed by Och and Ney (2003): intersection, union, and refined method. For Cross-EM, we also used the default configuration and jointly trained Model 1 and HMM for five iterations. For Vigne, we used a greedy search strategy by setting $\beta = 0$ and $b = 1$. Note that both GIZA++ and Cross-EM are unsupervised alignment methods.

On the English–French task, the refined combination of Model 4 alignments produced by GIZA++ in both translation directions yields an AER of 5.9%. Cross-EM outperforms GIZA++ significantly by achieving 5.1%. For Vigne, we use Model 4 as the primary feature. The linear combination of Model 4 in both directions achieves a lower AER than either one separately. The link count feature controls the number of links in the resulting alignments and leads to an absolute improvement of 0.1%. With the addition of cross count and neighbor count features, the AER score drops to 5.4%. We attribute this to the fact that the two features are capable of capturing the locality and monotonicity properties of natural languages, especially for closely related language pairs such as English–French. After adding the linked word count feature, our model achieves an AER of 5.2%. Finally, Vigne achieves an AER of 4.0% by combining predictions from refined Model 4 and jointly trained HMM.

On the Chinese–English task, one-to-many and many-to-one relationships occur more frequently in the reference alignments than the English–French task. As Cross-EM

6 In GIZA++ training, the final parameters are estimated on the final alignments, which are computed using the parameters obtained in the previous iteration. As a result, Vigne made use of the parameters generated by the iteration before the final iteration. In other experiments, Vigne used the final parameters.

Table 7
Comparison of GIZA++, Cross-EM, and Vigne on both tasks. Note that Vigne yields only one-to-one alignments if both "Model 4 s2t" and "Model 4 t2s" features are used. The pruning setting for Vigne is $\beta = 0$ and $b = 1$. While the final results of our system are better than the best baseline generative models significantly at $p < 0.01$, adding a single feature will not always produce a significant improvement, especially for English–French.

System	Setting	English–French	Chinese–English
GIZA++	Model 4 s2t	7.7	20.9
	Model 4 t2s	9.2	30.3
	Intersection	6.8	21.8
	Union	9.6	28.1
	Refined method	5.9	18.4
Cross-EM	HMM, joint	5.1	18.9
Vigne	Model 4 s2t	7.8	20.5
	+Model 4 t2s	5.6	18.3
	+link count	5.5	17.7
	+cross count	5.4	17.6
	+neighbor count	5.2	17.4
	+exact match	5.3	-
	+linked word count	5.2	17.3
	+bilingual dictionary	-	17.1
	+link co-occurrence count (GIZA++)	5.1	16.3
	+link co-occurrence count (Cross-EM)	4.0	15.7

is prone to produce one-to-one alignments by encouraging agreement, symmetrizing Model 4 by refined method yields better results than Cross-EM. We observe that the advantages of adding features such as link count, cross count, neighbor count, and linked word count to our linear model continue to hold, resulting in a much lower AER than both GIZA++ and Cross-EM. The addition of the bilingual dictionary is beneficial and yields an AER of 17.1%. Further improvements were obtained by including predictions from GIZA++ and Cross-EM.

As the IBM models do not allow a source word to be aligned with more than one target word, the activation of the IBM models in both directions always yields one-to-one alignments and thus has a loss in recall. To alleviate this problem, we use a heuristic postprocessing step to produce many-to-one or one-to-many alignments. First, we collect links that have higher translation probabilities than corresponding null links in both directions. Then, these candidate links are sorted according to their translation probabilities. Finally, they are added to the alignments under structural constraints similar to those of Och and Ney (2003).

On the English–French task, this symmetrization method achieves relatively small but very consistent improvements ranging from 0.1% to 0.2%. On the Chinese–English task, the improvements are more significant, ranging from 0.1% to 0.8%. This difference also results from the fact that the reference alignments of the Chinese–English task contain more one-to-many and many-to-one relationships than the English–French task. After symmetrization, the final AER scores for the two tasks are 3.8% and 15.1%, respectively.

4.1.3 Resulting Feature Weights. Table 8 shows the resulting feature weights of minimum error rate training. We observe that adding new features has an effect on the weights

Table 8
Resulting feature weights of minimum error rate training on the Chinese–English task (M4ST: Model 4 s2t; M4TS: Model 4 t2s; LC: link count; CC: cross count; NC: neighbor count; LWC: linked word count; BD: bilingual dictionary; LCCG: link co-occurrence count (GIZA++); LCCC: link co-occurrence count (Cross-EM)).

	M4ST	M4TS	LC	CC	NC	LWC	BD	LCCG	LCCC
M4ST	1.00	-	-	-	-	-	-	-	-
+M4TS	0.63	0.37	-	-	-	-	-	-	-
+LC	0.18	0.07	−0.75	-	-	-	-	-	-
+CC	0.19	0.07	−0.56	−0.18	-	-	-	-	-
+NC	0.12	0.06	−0.55	−0.08	0.17	-	-	-	-
+LWC	0.14	0.08	−0.22	−0.08	0.25	−0.26	-	-	-
+BD	0.07	0.02	−0.35	−0.05	0.16	0.01	0.34	-	-
+LCCG	0.03	0.04	−0.13	−0.05	0.20	−0.16	0.28	0.11	-
+LCCC	0.02	0.02	0.14	−0.03	0.10	−0.26	0.30	0.04	0.09

of other features. The weights of the cross count feature are consistently negative, suggesting that crossing links are always discouraged for Chinese–English. Also, the positive weights of the neighbor count feature indicate that monotonic alignments are encouraged. When the bilingual dictionary was included, the weights of Model 4 features in both directions dramatically decreased.

4.1.4 Results of the Symmetric Alignment Model. As we mentioned before, the linear model can model many-to-many alignments directly without any postprocessing symmetrization heuristics.

Table 9 demonstrates the results of the symmetric alignment model on both tasks. As the activation of translation and fertility probability products allows for arbitrary relationships, the addition of the link count feature excludes most loosely related links

Table 9
AER scores achieved by the symmetric alignment model on both tasks. The pruning setting for Vigne is $\beta = 0$ and $b = 1$. Although the final model obviously outperforms the initial model significantly at $p < 0.01$, adding a single feature will not always result in a significant improvement, especially for English–French.

Features	English–French	Chinese–English
translation probability product	17.3	23.6
+fertility probability product	14.6	22.6
+link count	14.5	21.6
+cross count	5.8	18.5
+neighbor count	5.2	17.2
+exact match	5.1	-
+linked word count	5.2	17.0
+link types	5.0	16.9
+sibling distance	4.9	16.2
+bilingual dictionary	-	15.9
+link co-occurrence count (GIZA++)	4.5	15.1
+link co-occurrence count (Cross-EM)	3.7	14.5

and results in more significant improvements than for asymmetric IBM models. One interesting finding is that the cross count feature is very useful, leading to dramatic absolute reduction of 8.7% on the English–French task and 3.1% on the Chinese–English task, respectively. We find that the advantages of adding neighbor count and linked word count still hold. By further including predictions from GIZA++ and Cross-EM, our linear model achieves the best result: 3.7% on the English–French task and 14.5% on the Chinese–English task.

We find that the symmetric linear model outperforms the asymmetric one, especially on the Chinese–English task. This suggests that although the asymmetric model can produce symmetric alignments via symmetrization heuristics, the "genuine" symmetric model produces many-to-many alignment in a more natural way.

4.1.5 Effect of Beam Search. Table 10 shows the effect of varying beam widths. The aligning speed (words per second) decreases almost linearly with the increase of beam width b. For simple alignment models such as using only the translation probability product feature, enlarging the beam size fails to bring improvements due to modeling errors. When more features are added, the model becomes more expressive. Therefore, our system benefits from larger beam size consistently, although some benefits are not significant statistically. When we set $b = 10$, the final AER scores for the English–French and Chinese–English tasks are 3.6% and 14.3%, respectively.

4.1.6 Effect of Training Corpus Size. One disadvantage of our approach is that we need a hand-aligned training corpus for training feature weights. However, compared with building a treebank, manual alignment is far less expensive because one annotator only needs to answer yes–no questions: Should this pair of words be aligned or not? If well trained, even a non-linguist who is familiar with both source and target languages could

Table 10
Comparison of aligning speed (words per second) and AER score with varying beam widths for the Chinese–English task. We fix $\beta = 0.01$. **Bold** numbers refer to the results that are better than the baseline but not significantly so. We use "+" to denote the results that outperform the best baseline ($b = 1$) and are statistically significant at $p < 0.05$. Similarly, we use "++" to denote significantly better than baseline at $p < 0.01$.

	b=1		b=5		b=10	
Features	w/sec	AER	w/sec	AER	w/sec	AER
translation probability product	3,941	23.6	843	23.6	426	23.7
+fertility probability product	1,418	22.6	300	22.7	150	22.9
+link count	1,557	21.6	330	21.7	166	21.9
+cross count	1,696	18.5	359	18.6	180	18.6
+neighbor count	1,648	17.2	355	**16.8**$^{+}$	178	**16.7**$^{+}$
+linked word count	1,627	17.0	351	**16.4**$^{+}$	176	**16.5**$^{+}$
+sibling distance	1,531	16.9	326	**16.5**$^{+}$	165	**16.4**$^{+}$
+link types	899	16.2	187	**15.6**$^{+}$	96	**15.5**$^{++}$
+bilingual dictionary	890	15.9	187	**15.6**	94	**15.5**$^{+}$
+link co-occurrence count (GIZA++)	877	15.1	182	**15.0**	92	**14.9**
+link co-occurrence count (Cross-EM)	867	14.5	183	**14.4**	92	**14.3**

produce high-quality alignments. We estimate that aligning a sentence pair usually takes only two minutes on average.

An interesting question is: How many training examples are needed to train a good discriminative model? Figure 4 shows the learning curves with different numbers of features on the Chinese–English task. We choose four feature groups with varying numbers of features: 3, 6, 10, and 14. There are eight fractions of the training corpus: 10, 20, 50, 100, 200, 300, 400, and 502. Generally, the more features a model uses, the more training examples are needed to train feature weights. Surprisingly, even when we use 14 features, 50 sentences seem to be good enough for minimum error rate training. This finding suggests that our approach could work well even with a quite small training corpus.

4.1.7 Summary of Results. Table 11 summarizes the results on all three shared tasks. Vigne used the same configuration for all tasks. We used the symmetric linear model and activated all features. The pruning setting is $\beta = 0.01$ and $b = 10$. Our system outperforms the systems participating in all the three shared tasks significantly.

Note that for the English–French task we used the small development set of 37 sentences to optimize feature weights, and ran our system on the original test set of 447 sentences. For the Romanian–English language pair, we follow Fraser and Marcu (2006) in reducing the vocabulary by stemming Romanian and English words down to their first four characters. For the other language pairs, English–Inuktitut and English–Hindi, the symmetric linear model maintains its superiority over the asymmetric linear model and yields better results than the other participants.

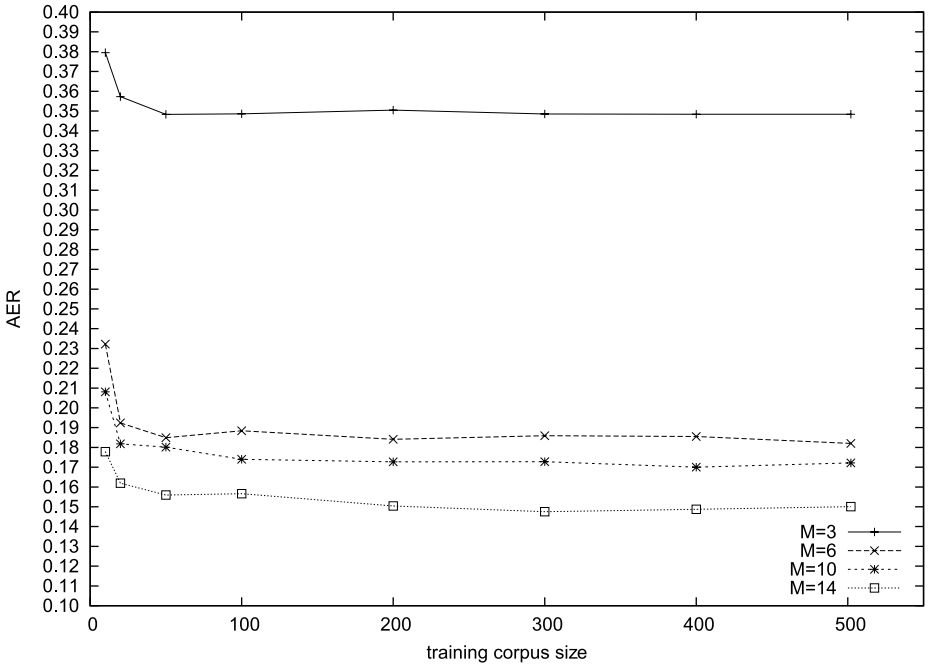

Figure 4
Effect of training corpus size on the Chinese–English task. We choose four feature groups with varying numbers of features: 3, 6, 10, and 14. There are eight training corpora with varying numbers of sentence pairs: 10, 20, 50, 100, 200, 300, 400, and 502.

Table 11
Comparison with the systems participating in the three shared tasks. "non-null" denotes that the reference alignments have no null links, "null" denotes that the reference alignments have null links, "limited" denotes only limited resources can be used, and "unlimited" denotes that there are no restrictions on resources used.

Shared Task	Task	Participants	Vigne
HLT-NAACL 2003	Romanian–English, non-null, limited	28.9–52.7	23.5
	Romanian–English, null, limited	37.4–59.8	26.9
	English–French, non-null, limited	8.5–29.4	4.0
	English–French, null, limited	18.5–51.7	4.6
ACL 2005	English–Inuktitut, limited	9.5–71.3	8.9
	Romanian–English, limited	26.6–44.5	24.7
	English–Hindi, limited	51.4	44.8
HTRDP 2005	Chinese–English, unlimited	23.5–49.2	14.3

4.1.8 Comparison to Other Work. In the word alignment literature, the Canadian Hansard bilingual corpus is the most widely used data set. Table 12 lists alignment error rates achieved by previous work and our system. Note that direct comparisons are problematic due to the different configurations of training data, development data, and test data. Our result matches the state-of-the-art performance on the Hansard data (Lacoste-Julien et al. 2006; Moore, Yih, and Bode 2006).

4.2 Evaluation of Translation Quality

In this section, we report on experiments with Chinese-to-English translation. To investigate the effect of our discriminative model on translation performance, we used three translation systems:

1. *Moses* (Koehn and Hoang 2007), a state-of-the-art phrase-based SMT system;
2. *Hiero* (Chiang 2007), a state-of-the-art hierarchical phrase-based system;
3. *Lynx* (Liu, Liu, and Lin 2006), a linguistically syntax-based system that makes use of tree-to-string rules.

Table 12
Comparison of some word alignment systems on the Canadian Hansard data.

System	Training	Test	AER
Och and Ney (2003)	1.5M	500	5.2
Moore (2005)	500K	223	7.5
Taskar, Lacoste-Julien, and Klein (2005)	1.1M	347	5.4
Liang, Taskar, and Klein (2006)	1.1M	347	4.9
Lacoste-Julien et al. (2006)	1.1M	247	3.8
Blunsom and Cohn (2006)	1.1M	347	5.2
Moore, Yih, and Bode (2006)	1.1M	223	3.7
This work	1.1M	247	3.6

For all three systems we trained the translation models on the FBIS corpus (7.2M+9.2M words). For the language model, we used the SRI Language Modeling Toolkit (Stolcke 2002) to train a trigram model with modified Kneser-Ney smoothing on the Xinhua portion of the Gigaword corpus. We used the 2002 NIST MT evaluation test set as the development set for training feature weights of translation systems, the 2005 test set as the devtest set for choosing optimal values of α for different translation systems, and the 2008 test set as the final test set. Our evaluation metric is case-sensitive BLEU-4, as defined by NIST, that is, using the shortest (as opposed to closest) reference length for brevity penalty.

We annotated the first 200 sentences of the FBIS corpus using the Blinker guidelines (Melamed 1998). All links are sure ones. These hand-aligned sentences served as the training corpus for Vigne. To train the feature weights in our discriminative model using minimum-error-rate training (Och 2003), we adopt balanced F-measure (Fraser and Marcu 2007b) as the optimization criterion.

The pipeline begins by running GIZA++ and Cross-EM on the FBIS corpus. We used seven generative alignment methods based on IBM Model 4 and HMM as baseline systems: (1) C→E, (2) E→C, (3) intersection, (4) union, (5) refined method (Och and Ney 2003), (6) grow-diag-final (Koehn, Och, and Marcu 2003), and (7) Cross-EM (Liang, Taskar, and Klein 2006). Instead of exploring the entire search space, our linear model only searches within the union of baseline predictions, which enables our system to align large bilingual corpus at a very fast speed of 3,000 words per second. In other words, our system is able to annotate the FBIS corpus in about 1.5 hours. Then, we train the feature weights of the linear model on the training corpus with respect to F-measure under different settings of α. After that, our system runs on the FBIS corpus to produce word alignments using the optimized weights. Finally, the three SMT systems train their models on the word-aligned FBIS corpus.

Can our approach achieve higher F-measure scores than generative methods with different values of α (the weighting factor in F-measure)? Table 13 shows the results of all the systems on the development set. To estimate the loss from restricting the search

Table 13
Maximization of F-measure with different settings of α (the weighting factor in the balanced F-measure). We use IBM Model 4 and HMM as baseline systems. Our system restricts the search space by exploring only the union of baseline predictions. We compute the "oracle" alignments by intersecting the union with reference alignments. We use "+" to denote the result that outperforms the best baseline result with statistical significance at $p < 0.05$. Similarly, we use "++" to denote significantly better than baseline at $p < 0.01$.

	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
IBM Model 4 C→E	82.6	81.3	80.1	79.0	77.8
IBM Model 4 E→C	68.2	70.5	73.0	75.7	78.5
IBM Model 4 intersection	63.6	68.9	75.2	82.8	92.1
IBM Model 4 union	86.6	82.0	77.9	74.2	70.8
IBM Model 4 refined method	75.4	78.5	81.8	85.4	89.4
IBM Model 4 grow-diag-final	82.4	82.1	81.7	81.4	81.1
Cross-EM HMM	70.4	73.7	77.3	81.2	85.5
oracle	91.9	93.6	95.3	97.1	99.0
Vigne	**87.8**$^{++}$	**85.8**$^{++}$	**86.4**$^{++}$	**88.6**$^{++}$	**93.3**$^{++}$

space, we compute **oracle** alignments by intersecting the union of baseline predictions with reference alignments. The F-measures achieved by oracle alignments range from 91.9 to 99.0, indicating that the union of baseline predictions is good enough to approximate the true search space. We observe that C→E, union, and grow-diag-final weight recall higher because F-measure decreases when α increases. On the other hand, E→C, intersection, refined method, and Cross-EM weight precision higher. In particular, α has a weak effect on grow-diag-final as its F-measure always keeps above 0.8 when α is varied. For each α, we trained a set of feature weights to maximize the F-measure on the development set. We find that our discriminative model outperforms the baseline systems significantly at all values of α.

Table 14 shows the BLEU scores of the three systems on the devtest set. For Moses and Hiero, we used the default setting. For Lynx, we used the phrase pairs learned by Moses to improve rule coverage (Liu, Liu, and Lin 2006). The best generative alignment method is grow-diag-final, which is widely used in SMT. For all the three SMT systems, our system outperforms the baseline systems statistically significantly. For Moses, the best value of α is 0.5. For Hiero and Lynx, the best α is 0.3, suggesting that recall-oriented alignments yield better translation performance.

Table 15 gives the BLEU scores of the three systems on the final test set. We used the parameters optimized on the dev and devtest sets. More specifically, Moses used grow-diag-final and $\alpha = 0.5$, Hiero used grow-diag-final and $\alpha = 0.3$, and Lynx used union and $\alpha = 0.3$. We find that our discriminative alignment model improves the three systems significantly.

5. Related Work

The first generative alignment models were the IBM Models 1–5 proposed by Brown et al. (1993). Vogel and Ney (1996) propose a first-order Hidden Markov model (HMM) for word alignment. They show that it is beneficial to make the alignment probabilities dependent on differences in position rather than on the absolute positions. Och and

Table 14
BLEU scores on the devtest set. We use "+" to denote the result that outperforms the best baseline result (highlighted in **bold**) statistically significantly at $p < 0.05$. Similarly, we use "++" to denote significantly better than baseline at $p < 0.01$.

		Moses	Hiero	Lynx
IBM Model 4 C→E		24.7	25.7	24.8
IBM Model 4 E→C		20.6	23.5	21.6
IBM Model 4 intersection		20.1	23.2	21.2
IBM Model 4 union		24.3	24.1	**25.1**
IBM Model 4 refined method		24.2	24.0	24.2
IBM Model 4 grow-diag-final		**25.0**	**25.8**	24.3
Cross-EM HMM		23.6	24.9	24.8
Vigne	$\alpha = 0.1$ tuned	23.9	25.3	**26.0**$^{++}$
	$\alpha = 0.3$ tuned	24.9	**26.8**$^{++}$	**26.1**$^{++}$
	$\alpha = 0.5$ tuned	**25.7**$^{+}$	**26.6**$^{++}$	24.3
	$\alpha = 0.7$ tuned	23.7	25.4	24.7
	$\alpha = 0.9$ tuned	21.9	24.7	23.9

Table 15
BLEU scores on the final test set. We use the parameters optimized on the dev and devtest sets. We use "+" to denote the result that outperforms the best baseline result (indicated in **bold**) statistically significantly at $p < 0.05$. Similarly, we use "++" to denote significantly better than baseline at $p < 0.01$.

	Moses	Hiero	Lynx
generative	20.1	20.7	19.9
discriminative	**20.8**$^{+}$	**21.6**$^{+}$	**21.0**$^{++}$

Ney (2003) re-implement the IBM models and the HMM model and compare them with heuristic approaches systematically. The resulting toolkit GIZA++ developed by Franz J. Och is the most popular alignment system nowadays. Liang, Taskar, and Klein (2006) present an unsupervised way to produce symmetric alignments by training two simple asymmetric models (e.g., IBM Model 1 and the HMM model) jointly to maximize a combination of data likelihood and agreement between the models. Fraser and Marcu (2007a) introduce a new generative model called LEAF that directly models many-to-many non-consecutive word alignments. Their model can be trained using both unsupervised and semi-supervised training methods.

Recent years have witnessed the rapid development of discriminative alignment methods. As a first attempt, Och and Ney (2003) proposed the Model 6, which is a log-linear combination of the IBM models and the HMM model. Cherry and Lin (2003) develop a statistical model to find word alignments, which allows for easy integration of context-specific features. Liu, Liu, and Lin (2005) apply the log-linear model used in SMT (Och and Ney 2002) to word alignment and report significant improvements over the IBM models. Moore (2005) presents a discriminative framework for word alignment and uses averaged perceptron for parameter optimization. Taskar, Lacoste-Julien, and Klein (2005) treat the alignment prediction task as a maximum weight bipartite matching problem and use the large-margin method to train feature weights. Neural networks and transformation-based learning have also been introduced to word alignment (Ayan, Dorr, and Monz 2005a, 2005b). Blunsom and Cohn (2006) propose a new discriminative model based on conditional random fields (CRF). Fraser and Marcu (2006) use sub-models of IBM Model 4 as features and train feature weights using a semi-supervised algorithm. Ayan and Dorr (2006b) use a maximum entropy model to combine word alignments. Cherry and Lin (2006) show that introducing soft syntactic constraints through discriminative training can improve alignment quality. Lacoste-Julien et al. (2006) extend the bipartite matching model of Taskar, Lacoste-Julien, and Klein (2005) by including fertility and first-order interactions. Recently, max-product belief propagation has been successfully applied to discriminative word alignment (Niehues and Vogel 2008; Cromierès and Kurohashi 2009). Haghighi et al. (2009) investigate supervised word alignment methods that exploit inversion transduction grammar (ITG) constraints.

Our work can be seen as an application of the linear model (Och 2003) in word alignment. While aiming at producing symmetric word alignments in a discriminative way, our approach uses asymmetric generative models (Brown et al. 1993) as the major information sources. Our linear model is similar to that of Moore, Yih, and Bode (2006). They train two linear models called stage 1 and stage 2. The feature values are extracted from word-aligned sentence pairs. After the stage 1 model aligns the entire training corpus automatically, the stage 2 model uses features based not only on the parallel

sentences themselves but also on statistics of the alignments produced by the stage 1 model. They use average perceptron and support vector machine (SVM) to train feature weights and use a beam search algorithm to find the most probable alignments. Table 12 shows that the two methods achieve comparable results on the Hansard data, confirming Moore, Yih, and Bode's (2006) claim that model structure and feature selection are more important than discriminative training method.

6. Conclusions and Future Work

We have presented a discriminative framework for word alignment based on the linear modeling approach. This framework is easy to extend by including features that characterize the aligning process. In addition, our approach supports symmetric alignment modeling that allows for an arbitrary relationship between source and target language positions. As the linear model offers excellent flexibility in using a large variety of features and in combining information from various sources, it is able to produce good predictions on language pairs that are either closely related (e.g., English–French) or distantly related (e.g., English–Inuktitut), either with rich resources (e.g., Chinese–English) or with scarce resources (e.g., English–Hindi). We further show that our approach can benefit different types of SMT systems: phrase-based, hierarchical phrase-based, and syntax-based.

The real benefit of our model does not stem from the use of the linear model, but rather from the discriminative training that optimizes feature weights with respect to evaluation metrics on the gold-standard word alignments. One disadvantage of our approach is the need for annotated training data. Although we have shown that a very small number of training examples would be enough for parameter estimation (Section 4.1.6), it is difficult to select such a representative training corpus to ensure that the model will work well on unseen data, especially when the bilingual corpus to be aligned consists of parallel texts from different domains.

Another problem is that it is hard to find an evaluation metric for word alignment that correlates well with translation quality because the relationship between alignment and translation is still not quite clear. Without a good loss function, discriminative models cannot outperform generative models in large-scale applications. Therefore, it is important to investigate how to select training examples and how to choose optimization criterion.

The design of feature functions is most important for a discriminative alignment model. Often, we need to try various feature groups manually on the development set to determine the optimal feature group. Furthermore, a feature group optimized for one language pair may not have the same effect on another one. In the future, we plan to investigate an algorithm for automatic feature selection.

Appendix A: Table of Notation

$\mathbf{f}$	source sentence
$\mathbf{f}_1^S$	sequence of source sentences: $\mathbf{f}_1, \ldots, \mathbf{f}_s, \ldots, \mathbf{f}_S$
f	source word
J	length of $\mathbf{f}$
j	position in $\mathbf{f}$, $j = 1, 2, \ldots, J$
f_j	the j-th word in $\mathbf{f}$
f_0	empty cept on the source side

$\mathbf{e}$	target sentence
$\mathbf{e}_1^S$	sequence of target sentences: $\mathbf{e}_1, \ldots, \mathbf{e}_s, \ldots, \mathbf{e}_S$
e	target word
I	length of $\mathbf{e}$
i	position in $\mathbf{e}$, $i = 1, 2, \ldots, I$
e_i	the i-th word in $\mathbf{e}$
e_0	empty cept on the target side
$\mathbf{a}$	alignment
l	a link (j, i) in $\mathbf{a}$
ψ_j	number of positions of $\mathbf{e}$ connected to position j of $\mathbf{f}$
ϕ_i	number of positions of $\mathbf{f}$ connected to position i of $\mathbf{e}$
$\omega_{j,k}$	position of the k-th target word aligned to f_j
$\pi_{i,k}$	position of the k-th source word aligned to e_i
$\Delta(j, \psi_j)$	sum of sibling distances for f_j
$\nabla(i, \phi_i)$	sum of sibling distances for e_i
$score(\mathbf{f}, \mathbf{e}, \mathbf{a})$	a score that indicates how well $\mathbf{a}$ is the alignment between $\mathbf{f}$ and $\mathbf{e}$
$\hat{\mathbf{a}}$	the best candidate alignment
λ	feature weight
γ	the feature weight being optimized
$h(\mathbf{f}, \mathbf{e}, \mathbf{a})$	feature function
$\mathcal{G}(\mathbf{f}, \mathbf{e}, \mathbf{a}, l)$	link gain after adding l to $\mathbf{a}$
$g(\mathbf{f}, \mathbf{e}, \mathbf{a}, l)$	feature gain after adding l to $\mathbf{a}$
$\mu(\mathbf{f}, \mathbf{e}, \mathbf{a})$	value of the feature being optimized
$\sigma(\mathbf{f}, \mathbf{e}, \mathbf{a})$	dot-product of fixed features
$t(e\|f)$	the probability that f is translated to e
$t(f\|e)$	the probability that e is translated to f
$n(\psi\|f)$	the probability that f has a fertility of ψ
$n(\phi\|e)$	the probability that e has a fertility of ϕ
$\mathbf{r}$	reference alignment
$\mathbf{C}_s$	set of candidate alignments for the s-th training example
$\mathbf{a}_{s,k}$	the k-th candidate alignment for the s-th training example
$E(\mathbf{r}, \mathbf{a})$	loss function that measures alignment quality
α	the precision/recall weighting factor in balanced F-measure
β	pruning threshold in the beam search algorithm
b	beam size in the beam search algorithm
$\delta(x, y)$	the Kronecker function, which is 1 if $x = y$ and 0 otherwise
$[\![expr]\!]$	an indicator function taking a boolean expression $expr$ as the argument

Appendix B: Using the IBM Models as Feature Functions

In this article, we use IBM Models 1–4 as feature functions by taking the logarithm of the models themselves rather than the sub-models just for simplicity. It is easy to separate each sub-model as a feature as suggested by Fraser and Marcu (2006). We distinguish

between two translation directions (i.e., source-to-target and target-to-source) to use the IBM models as feature functions. All model parameters are estimated by GIZA++ (Och and Ney 2003).

The feature function for the IBM Model 1 is

$$h_{m_1}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \log\left(\frac{\epsilon(J|I)}{(I+1)^J}\prod_{j=1}^{J} t(f_j|e_{a_j})\right) \tag{B.1}$$

where $\epsilon(J|I)$ predicts the length of the source sentence conditioned on that of the target sentence, $(I+1)^{-J}$ defines a uniform distribution of the alignment between source and target words, and $t(f_j|e_i)$ is a translation sub-model. Note that $a_j = i$, which means that f_j is connected to e_i.

The corresponding feature gain is

$$g_{m_1}(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) = \log\big(t(f_j|e_i)\big) - \log\big(t(f_j|e_0)\big) \tag{B.2}$$

where f_j and e_i are linked by l and e_0 is the empty cept to which all unaligned source words are "aligned."

Based on a similar generative story to Model 1, Model 2 replaces the uniform alignment probability distribution with an alignment sub-model $a(i|j, I, J)$. This sub-model assumes that the position of e_i depends on the position of its translation f_j and sentence lengths I and J.

The feature function for Model 2 is

$$h_{m_2}(\mathbf{f}, \mathbf{e}, \mathbf{a}) = \log\left(\epsilon(J|I)\prod_{j=1}^{J} t(f_j|e_{a_j})a(a_j|j, I, J)\right) \tag{B.3}$$

The corresponding feature gain is

$$\begin{aligned} g_{m_2}(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) = {} & \log(t(f_j|e_i)) - \log(t(f_j|e_0)) + \\ & \log(a(i|j, I, J)) - \log(a(0|j, I, J)) \end{aligned} \tag{B.4}$$

where f_j and e_i are linked by l and 0 is the index of the empty cept e_0.

Model 3 is a fertility-based model that parameterizes fertility of words. Unlike Model 2, Model 3 uses a fertility sub-model $n(\phi_i|e_i)$ and a distortion sub-model $d(j|i, I, J)$. Formally, the feature function of Model 3 is given by

$$\begin{aligned} h_{m_3}(\mathbf{a}, \mathbf{f}, \mathbf{e}) = \log\Bigg(& n_0\Big(\phi_0|\sum_{i=1}^{I}\phi_i\Big)\prod_{i=1}^{I} n(\phi_i|e_i)\phi_i!\prod_{j=1}^{J} t(f_j|e_{a_j}) \\ & \times \prod_{j:a_j\neq 0} d(j|i, I, J)\Bigg) \end{aligned} \tag{B.5}$$

Brown et al. (1993) treat $n_0(\phi_0|\sum_{i=1}^{I}\phi_i)$, the fertility probability of e_0, in a different way. They assume that at most half of the source words in an alignment are not

aligned (i.e., $\phi_0 \leq \frac{J}{2}$) and define a binomial distribution relying on an auxiliary parameter p_0:

$$n_0\left(\phi_0|\sum_{i=1}^{I}\phi_i\right) = \begin{cases} \binom{J-\phi_0}{\phi_0} p_0^{J-2\phi_0}(1-p_0)^{\phi_0} & \text{if } \phi_0 \leq \frac{J}{2} \\ 0 & \text{otherwise} \end{cases} \tag{B.6}$$

Note that we follow Brown et al. (1993) in replacing $\sum_{i=1}^{I}\phi_i$ with $J-\phi_0$ for simplicity. The original form should be $\binom{\sum_{i=1}^{I}\phi_i}{\phi_0} p_0^{\sum_{i=1}^{I}\phi_i-\phi_0}(1-p_0)^{\phi_0}$.

However, this assumption results in a problem for our search algorithm that begins with an empty alignment (see Algorithm 1), for which ϕ_0 is J and the feature value $h_{m_3}(\mathbf{f},\mathbf{e},\mathbf{a})$ is negative infinity. To alleviate this problem, we modify Equation B.6 slightly by adding a smoothing parameter $p_n \in (0,1)$:

$$n_0\left(\phi_0|\sum_{i=1}^{I}\phi_i\right) = \begin{cases} \binom{J-\phi_0}{\phi_0} p_0^{J-2\phi_0}(1-p_0)^{\phi_0} p_n & \text{if } \phi_0 \leq \frac{J}{2} \\ \frac{1-p_n}{\lfloor \frac{J}{2} \rfloor} & \text{otherwise} \end{cases} \tag{B.7}$$

Therefore, the feature gain for Model 3 is

$$\begin{aligned} g_{m_3}(\mathbf{f},\mathbf{e},\mathbf{a},l) = {} & \log(g_{n_0}(J,\phi_0)) + \\ & \log(n(\phi_i+1|e_i)) - \log(n(\phi_i|e_i)) + \\ & \log(\phi_i+1) + \\ & \log(t(f_j|e_i)) - \log(t(f_j|e_0)) + \\ & \log(d(j|i,I,J)) \end{aligned} \tag{B.8}$$

where f_j and e_i are linked by l, ϕ_i is the fertility before adding l, and $g_{n_0}(J,\phi_0)$ is the gain for $n_0(\phi_0|\sum_{i=1}^{I}\phi_i)$:

$$g_{n_0}(J,\phi_0) = \begin{cases} \frac{\phi_0\times(J-\phi_0+1)}{(J-2\phi_0+1)\times(J-2\phi_0+2)} & \text{if } \phi_0 \leq \frac{J}{2} \\ \frac{1-p_n}{\binom{J-\phi_0+1}{\phi_0-1}\times p_0^{J-2\phi_0+2}\times(1-p_0)^{\phi_0-1}\times p_n\times\lfloor\frac{J}{2}\rfloor} & \frac{J}{2} < \phi_0 \leq \frac{J}{2}+1 \\ 1 & \text{otherwise} \end{cases} \tag{B.9}$$

Model 4 defines a new distortion sub-model $\mathcal{D}(\mathbf{a})$ that relies on word classes $\mathcal{A}$ and $\mathcal{B}$ to capture movement of phrases. The feature function for Model 4 is

$$h_{m_4}(\mathbf{a},\mathbf{f},\mathbf{e}) = \log\left(n_0(\phi_0|\sum_{i=1}^{I}\phi_i)\prod_{i=1}^{I}n(\phi_i|e_i)\prod_{j=1}^{J}t(f_j|e_{a_j})\frac{1}{\phi_0!}\mathcal{D}(\mathbf{a})\right) \tag{B.10}$$

where

$$\mathcal{D}(\mathbf{a}) = \prod_{i=1}^{I}\prod_{k=1}^{\phi_i} p_{ik}(\pi_{ik}) \tag{B.11}$$

$$p_{ik}(j) = \begin{cases} d_1(j - c_{\rho_i} | \mathcal{A}(e_{\rho_i}), \mathcal{B}(\tau_{i1})) & \text{if } k = 1 \\ d_{>1}(j - \pi_{i,k-1} | \mathcal{B}(\tau_{ik})) & \text{otherwise} \end{cases} \tag{B.12}$$

Brown et al. (1993) propose two distortion models for Model 4: $d_1(\cdot)$ for the first word of a **tablet** τ and $d_{>1}(\cdot)$ for the other words of the tablet. In Equation B.12, ρ_i is the first position to the left of i for which $\phi_i > 0$, c_{ρ_i} is the ceiling of the average position of the words in τ_ρ, τ_{ik} denotes the k-th French word aligned to e_k, $\pi_{i,k-1}$ denotes the position of the $k-1$-th French word aligned to e_i, and $\mathcal{A}(\cdot)$ and $\mathcal{B}(\cdot)$ are word classes for the source and target languages, respectively. Please refer to Brown et al. (1993) for more details.

The corresponding feature gain is

$$\begin{aligned} g_{m_4}(\mathbf{f}, \mathbf{e}, \mathbf{a}, l) = & \log(g_{n_0}(J, \phi_0)) + \\ & \log(n(\phi_i + 1 | e_i)) - \log(n(\phi_i | e_i)) + \\ & \log(t(f_j | e_i)) - \log(t(f_j | e_0)) + \\ & \log(\phi_0) + \\ & \log(\mathcal{D}(\mathbf{a} \cup \{l\})) - \log(\mathcal{D}(\mathbf{a})) \end{aligned} \tag{B.13}$$

where f_j and e_i are linked by l and ϕ_i is the fertility before adding l.

In Model 4, the addition of a single link might change the distortion probabilities $p_{ik}(j)$ of other links. As a result, we have to compute the overall distortion probabilities $\mathcal{D}(\mathbf{a})$ every time.

Acknowledgments

This work was supported by National Natural Science Foundation of China, Contract No. 60603095 and 60573188. Thanks to the three anonymous reviewers for their insightful and constructive comments and suggestions. We are grateful to Rada Mihalcea for giving us the Romanian–English training data and David Chiang for allowing us to use Hiero. Stephan Vogel, Vamshi Ambati, and Kelly Widmaier offered valuable feedback on an earlier version of this article.

References

Ayan, Necip Fazil and Bonnie J. Dorr. 2006a. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proceedings of COLING-ACL 2006*, pages 9–16, Sydney.

Ayan, Necip Fazil and Bonnie J. Dorr. 2006b. A maximum entropy approach to combining word alignments. In *Proceedings of HLT-NAACL 2006*, pages 96–103, New York, NY.

Ayan, Necip Fazil, Bonnie J. Dorr, and Christof Monz. 2005a. Alignment link projection using transformation-based learning. In *Proceedings of HLT-EMNLP 2005*, pages 185–192, Vancouver.

Ayan, Necip Fazil, Bonnie J. Dorr, and Christof Monz. 2005b. Neuralign: Combining word alignments using neural networks. In *Proceedings of HLT-EMNLP 2005*, pages 65–72, Vancouver.

Blunsom, Phil and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of COLING-ACL 2006*, pages 65–72, Sydney.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Cherry, Colin and Dekang Lin. 2003. A probability model to improve word alignment. In *Proceedings of ACL 2003*, pages 88–95, Sapporo.

Cherry, Colin and Dekang Lin. 2006. Soft syntactic constraints for word alignment through discriminative training. In *Proceedings of COLING-ACL 2006 (poster)*, pages 105–112, Sydney.

Chiang, David. 2005. A hierarchical phrase-based model for statistical machine

translation. In *Proceedings of ACL 2005*, pages 263–270, Ann Arbor, MI.
Chiang, David. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.
Cromierès, Fabien and Sadao Kurohashi. 2009. An alignment algorithm using belief propagation and a structure-based distortion model. In *Proceedings of EACL 2009*, pages 166–174, Athens.
Fraser, Alexander and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of COLING-ACL 2006*, pages 769–776, Sydney.
Fraser, Alexander and Daniel Marcu. 2007a. Getting the structure right for word alignment: LEAF. In *Proceedings of EMNLP-CoNLL 2007*, pages 51–60, Prague.
Fraser, Alexander and Daniel Marcu. 2007b. Measuring word alignment quality for statistical machine translation. *Computational Linguistics*, 33(3):293–303.
Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of COLING-ACL 2006*, pages 961–968, Sydney.
Haghighi, Aria, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised ITG models. In *Proceedings of ACL-IJCNLP 2009*, pages 923–931, Suntec.
He, Xiaodong, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. 2008. Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of EMNLP 2008*, pages 98–107, Honolulu, HI.
Koehn, Philipp and Hieu Hoang. 2007. Factored translation models. In *Proceedings of EMNLP-CoNLL 2007*, pages 868–876, Prague.
Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133, Edmonton.
Lacoste-Julien, Simon, Ben Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of HLT-NAACL 2007*, pages 112–119, New York, NY.
Liang, Percy, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of HLT-NAACL 2006*, pages 104–111, New York, NY.
Liu, Yang, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of ACL 2005*, pages 459–466, Ann Arbor, MI.
Liu, Yang, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of COLING-ACL 2006*, pages 609–616, Sydney.
Marcu, Daniel, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of EMNLP 2006*, pages 44–52, Sydney.
Martin, Joel, Rada Mihalcea, and Ted Pedersen. 2005. Word alignment for languages with scarce resources. In *Proceedings of the ACL 2005 Workshop on Building and Using Parallel Texts*, pages 65–74, Ann Arbor, MI.
Melamed, I. Dan. 1998. Annotation style guide for the blinker project. Technical report No. 98-06, University of Pennsylvania, Philadelphia.
Melamed, I. Dan. 2000. Models for translational equivalence among words. *Computational Linguistics*, 26(2):221–249.
Mihalcea, Rada and Ted Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of HLT-NAACL 2003 Workshop on Building and Using Parallel Texts*, pages 1–10, Edmonton.
Moore, Robert C. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of HLT-EMNLP 2005*, pages 81–88, Vancouver.
Moore, Robert C., Wen-tau Yih, and Andreas Bode. 2006. Improved discriminative bilingual word alignment. In *Proceedings of COLING-ACL 2006*, pages 513–520, Sydney.
Niehues, Jan and Stephan Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25, Columbus, OH.
Och, Franz J. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL 2003*, pages 160–167, Sapporo.
Och, Franz J. and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of ACL 2002*, pages 295–302, Philadephia, PA.
Och, Franz J. and Hermann Ney. 2003. A systematic comparison of various

statistical alignment models. *Computational Linguistics*, 29(1):19–51.
Och, Franz J. and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.
Quirk, Chris, Arul Menezes, and Colin Cherry. 2005. Dependency treelet translation: Syntactically informed phrasal SMT. In *Proceedings of ACL 2005*, pages 271–279, Ann Arbor, MI.
Rosti, Antti-Veikko, Spyros Matsoukas, and Richard Schwartz. 2007. Improved word-level system combination for machine translation. In *Proceedings of ACL 2007*, pages 312–319, Prague.
Stolcke, Andreas. 2002. SRILM—an extensible language modeling toolkit. In *Proceedings of ICSLP 2002*, pages 901–904, Denver, CO.
Taskar, Ben, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of HLT-EMNLP 2005*, pages 73–80, Vancouver.
Vogel, Stephan and Hermann Ney. 1996. HMM-based word alignment in statistical translation. In *Proceedings of COLING 1996*, pages 836–841, Copenhagen.