Weighted Alignment Matrices for Statistical Machine Translation

Yang Liu , Tian Xia , Xinyan Xiao and **Qun Liu**
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology
Chinese Academy of Sciences
P.O. Box 2704, Beijing 100190, China
{yliu,xiatian,xiaoxinyan,liuqun}@ict.ac.cn

Abstract

Current statistical machine translation systems usually extract rules from bilingual corpora annotated with 1-best alignments. They are prone to learn noisy rules due to alignment mistakes. We propose a new structure called *weighted alignment matrix* to encode all possible alignments for a parallel text compactly. The key idea is to assign a probability to each word pair to indicate how well they are aligned. We design new algorithms for extracting phrase pairs from weighted alignment matrices and estimating their probabilities. Our experiments on multiple language pairs show that using weighted matrices achieves consistent improvements over using n-best lists in significant less extraction time.

1 Introduction

Statistical machine translation (SMT) relies heavily on annotated bilingual corpora. Word alignment, which indicates the correspondence between the words in a parallel text, is one of the most important annotations in SMT. Word-aligned corpora have been found to be an excellent source for translation-related knowledge, not only for phrase-based models (Och and Ney, 2004; Koehn et al., 2003), but also for syntax-based models (e.g., (Chiang, 2007; Galley et al., 2006; Shen et al., 2008; Liu et al., 2006)). Och and Ney (2003) indicate that the quality of machine translation output depends directly on the quality of initial word alignment.

Modern alignment methods can be divided into two major categories: *generative* methods and *discriminative* methods. Generative methods (Brown et al., 1993; Vogel and Ney, 1996) treat word alignment as a hidden process and maximize the likelihood of bilingual training corpus using the expectation maximization (EM) algorithm. In contrast, discriminative methods (e.g., (Moore et al., 2006; Taskar et al., 2005; Liu et al., 2005; Blunsom and Cohn, 2006)) have the freedom to define arbitrary feature functions that describe various characteristics of an alignment. They usually optimize feature weights on manually-aligned data. While discriminative methods show superior alignment accuracy in benchmarks, generative methods are still widely used to produce word alignments for large sentence-aligned corpora.

However, neither generative nor discriminative alignment methods are reliable enough to yield high quality alignments for SMT, especially for distantly-related language pairs such as Chinese-English and Arabic-English. The F-measures for Chinese-English and Arabic-English are usually around 80% (Liu et al., 2005) and 70% (Fraser and Marcu, 2007), respectively. As most current SMT systems only use 1-best alignments for extracting rules, alignment errors might impair translation quality.

Recently, several studies have shown that offering more alternatives of annotations to SMT systems will result in significant improvements, such as replacing 1-best trees with packed forests (Mi et al., 2008) and replacing 1-best word segmentations with word lattices (Dyer et al., 2008). Similarly, Venugopal et al. (2008) use n-best alignments instead of 1-best alignments for translation rule extraction. While they achieve significant improvements on the IWSLT data, extracting rules from n-best alignments might be computationally expensive.

In this paper, we propose a new structure named *weighted alignment matrix* to represent the alignment distribution for a sentence pair compactly. In a weighted matrix, each element that corresponds to a word pair is assigned a probability to measure the confidence of aligning the two words. Therefore, a weighted matrix is capable of using a lin-

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1017–1026,
Singapore, 6-7 August 2009. ©2009 ACL and AFNLP

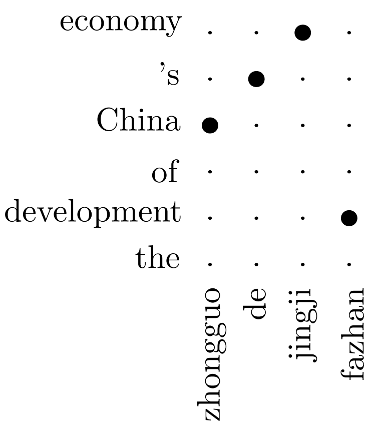


Figure 1: An example of word alignment between a pair of Chinese and English sentences.

ear space to encode the probabilities of exponentially many alignments. We develop a new algorithm for extracting phrase pairs from weighted matrices and show how to estimate their relative frequencies and lexical weights. Experimental results show that using weighted matrices achieves consistent improvements in translation quality and significant reduction in extraction time over using n-best lists.

2 Background

Figure 1 shows an example of word alignment between a pair of Chinese and English sentences. The Chinese and English words are listed horizontally and vertically, respectively. The dark points indicate the correspondence between the words in two languages. For example, the first Chinese word "*zhongguo*" is aligned to the fourth English word "*China*".

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

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

Usually, SMT systems only use the 1-best alignments for extracting translation rules. For example, given a source phrase $\tilde{f}$ and a target phrase $\tilde{e}$, the phrase pair $(\tilde{f}, \tilde{e})$ is said to be *consistent* (Och and Ney, 2004) with the alignment if and only if: (1) there must be at least one word inside one phrase aligned to a word inside the other phrase and (2) no words inside one phrase can be aligned to a word outside the other phrase.

After all phrase pairs are extracted from the training corpus, their translation probabilities can be estimated as *relative frequencies* (Och and Ney, 2004):

$$\phi(\tilde{e}|\tilde{f}) = \frac{count(\tilde{f}, \tilde{e})}{\sum_{\tilde{e}'} count(\tilde{f}, \tilde{e}')} \quad (2)$$

where $count(\tilde{f}, \tilde{e})$ indicates how often the phrase pair $(\tilde{f}, \tilde{e})$ occurs in the training corpus.

Besides relative frequencies, *lexical weights* (Koehn et al., 2003) are widely used to estimate how well the words in $\tilde{f}$ translate the words in $\tilde{e}$. To do this, one needs first to estimate a lexical translation probability distribution $w(e|f)$ by relative frequency from the same word alignments in the training corpus:

$$w(e|f) = \frac{count(f, e)}{\sum_{e'} count(f, e')} \quad (3)$$

Note that a special source NULL token is added to each source sentence and aligned to each unaligned target word.

As the alignment $\tilde{a}$ between a phrase pair $(\tilde{f}, \tilde{e})$ is retained during extraction, the lexical weight can be calculated as

$$p_w(\tilde{e}|\tilde{f}, \tilde{a}) = \prod_{i=1}^{|\tilde{e}|} \frac{1}{|\{j|(j,i) \in \tilde{a}\}|} \sum w(e_i|f_j) \quad (4)$$

If there are multiple alignments $\tilde{a}$ for a phrase pair $(\tilde{f}, \tilde{e})$, Koehn et al. (2003) choose the one with the highest lexical weight:

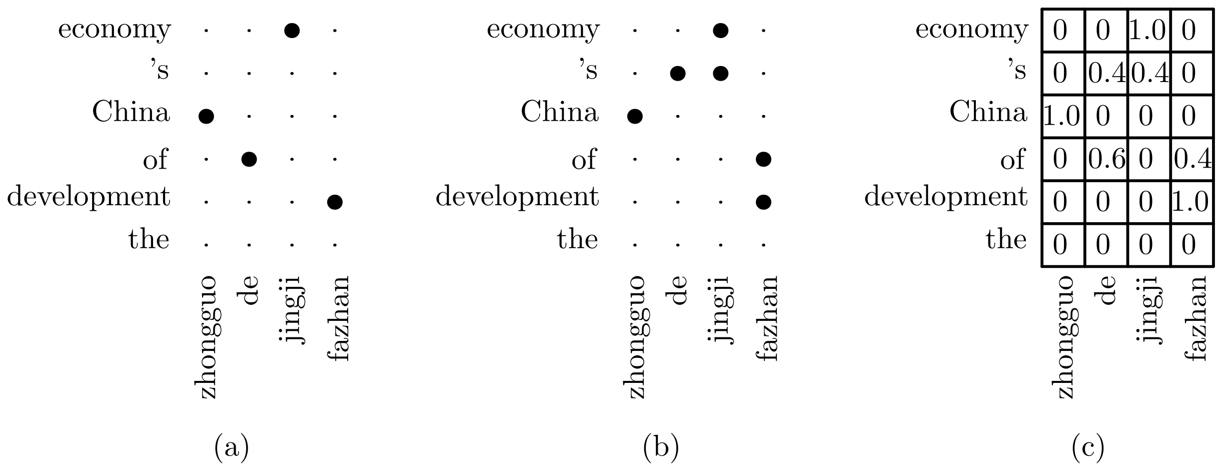
$$p_w(\tilde{e}|\tilde{f}) = \max_{\tilde{a}} \left\{ p_w(\tilde{e}|\tilde{f}, \tilde{a}) \right\} \quad (5)$$

Simple and effective, relative frequencies and lexical weights have become the standard features in modern discriminative SMT systems.

3 Weighted Alignment Matrix

We believe that offering more candidate alignments to extracting translation rules might help improve translation quality. Instead of using n-best lists (Venugopal et al., 2008), we propose a new structure called *weighted alignment matrix*.

We use an example to illustrate our idea. Figure 2(a) and Figure 2(b) show two alignments of a Chinese-English sentence pair. We observe that some links (e.g., (1,4) corresponding to the word



	zhongguo	de	jingji	fazhan
economy	0	0	1.0	0
's	0	0.4	0.4	0
China	1.0	0	0	0
of	0	0.6	0	0.4
development	0	0	0	1.0
the	0	0	0	0

(a) (b) (c)

Figure 2: (a) One alignment of a sentence pair; (b) another alignment of the same sentence pair; (c) the resulting weighted alignment matrix that takes the two alignments as samples, of which the initial probabilities are 0.6 and 0.4, respectively.

pair (“*zhongguo*”, “*China*”)) occur in both alignments, some links (e.g., (2,3) corresponding to the word pair (“*de*”,“*of*”)) occur only in one alignment, and some links (e.g., (1,1) corresponding to the word pair (“*zhongguo*”, “*the*”)) do not occur. Intuitively, we can estimate how well two words are aligned by calculating its relative frequency, which is the probability sum of alignments in which the link occurs divided by the probability sum of all possible alignments. Suppose that the probabilities of the two alignments in Figures 2(a) and 2(b) are 0.6 and 0.4, respectively. We can estimate the relative frequencies for every word pair and obtain a weighted matrix shown in Figure 2(c). Therefore, each word pair is associated with a probability to indicate how well they are aligned. For example, in Figure 2(c), we say that the word pair (“*zhongguo*”, “*China*”) is definitely aligned, (“*zhongguo*”, “the”) is definitely unaligned, and (“*de*”, “*of*”) has a 60% chance to get aligned.

Formally, a weighted alignment matrix m is a $J \times I$ matrix, in which each element stores a *link probability* $p_m(j,i)$ to indicate how well f_j and e_i are aligned. Currently, we estimate link probabilities from an n-best list by calculating relative frequencies:

$$\begin{aligned} p_m(j,i) &= \frac{\sum_{a \in \mathcal{N}} p(a) \times \delta(a,j,i)}{\sum_{a \in \mathcal{N}} p(a)} & (6) \\ &= \sum_{a \in \mathcal{N}} p(a) \times \delta(a,j,i) & (7) \end{aligned}$$

where

$$\delta(a,j,i) = \begin{cases} 1 & (j,i) \in a \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Note that $\mathcal{N}$ is an n-best list, $p(a)$ is the probability of an alignment a in the n-best list, $\delta(a,j,i)$ indicates whether a link (j,i) occurs in the alignment a or not. We assign 0 to any unseen alignment. As $p(a)$ is usually normalized (i.e., $\sum_{a \in \mathcal{N}} p(a) \equiv 1$), we remove the denominator in Eq. (6).

Accordingly, the probability that the two words f_j and e_i are not aligned is

$$\bar{p}_m(j,i) = 1.0 - p_m(j,i) \quad (9)$$

For example, as shown in Figure 2(c), the probability for the two words “*de*” and “*of*” being aligned is 0.6 and the probability that they are not aligned is 0.4.

Intuitively, the probability of an alignment a is the product of link probabilities. If a link (j,i) occurs in a, we use $p_m(j,i)$; otherwise we use $\bar{p}_m(j,i)$. Formally, given a weighted alignment matrix m, the probability of an alignment a can be calculated as

$$\begin{aligned} p_m(a) = \prod_{j=1}^{J} \prod_{i=1}^{I} (p_m(j,i) \times \delta(a,j,i) + \\ \bar{p}_m(j,i) \times (1 - \delta(a,j,i))) \quad (10) \end{aligned}$$

It proves that the sum of all alignment probabilities is always 1: $\sum_{a \in \mathcal{A}} p_m(a) \equiv 1$, where $\mathcal{A}$

```
 1: procedure PHRASEEXTRACT($f_1^J$, $e_1^I$, $m$, $l$)
 2:     $\mathcal{R} \leftarrow \emptyset$
 3:     for $j_1 \leftarrow 1 \ldots J$ do
 4:         $j_2 \leftarrow j_1$
 5:         while $j_2 < J \wedge j_2 - j_1 < l$ do
 6:             $T \leftarrow \{i | \exists j : j_1 \leq j \leq j_2 \wedge p_m(j, i) > 0\}$
 7:             $i_l \leftarrow \textsc{min}(T)$
 8:             $i_u \leftarrow \textsc{max}(T)$
 9:             for $n \leftarrow 1 \ldots l$ do
10:                 for $i_1 \leftarrow i_l - n + 1 \ldots i_u$ do
11:                     $i_2 \leftarrow i_1 + n - 1$
12:                     $\mathcal{R} \leftarrow \mathcal{R} \cup \{(f_{j_1}^{j_2}, e_{i_1}^{i_2})\}$
13:                 end for
14:             end for
15:             $j_2 \leftarrow j_2 + 1$
16:         end while
17:     end for
18:     return $\mathcal{R}$
19: end procedure
```

Figure 3: Algorithm for extracting phrase pairs from a sentence pair $\langle f_1^J, e_1^I \rangle$ annotated with a weighted alignment matrix m.

is the set of all possible alignments. Therefore, a weighted alignment matrix is capable of encoding the probabilities of $2^{J \times I}$ alignments using only a $J \times I$ space.

Note that $p_m(a)$ is not necessarily equal to $p(a)$ because the encoding of a weighted alignment matrix changes the alignment probability distribution. For example, while the initial probability of the alignment in Figure 2(a) (i.e., $p(a)$) is 0.6, the probability of the same alignment encoded in the matrix shown in Figure 2(c) (i.e., $p_m(a)$) becomes 0.1296 according to Eq. (10). It should be emphasized that a weighted matrix encodes all possible alignments rather than the input n-best list, although the link probabilities are estimated from the n-best list.

4 Phrase Pair Extraction

In this section, we describe how to extract phrase pairs from the training corpus annotated with weighted alignment matrices (Section 4.1) and how to estimate their relative frequencies (Section 4.2) and lexical weights (Section 4.3).

4.1 Extraction Algorithm

Och and Ney (2004) describe a "phrase-extract" algorithm for extracting phrase pairs from a sentence pair annotated with a 1-best alignment. Given a source phrase, they first identify the target phrase that is consistent with the alignment. Then, they expand the boundaries of the target phrase if the boundary words are unaligned.

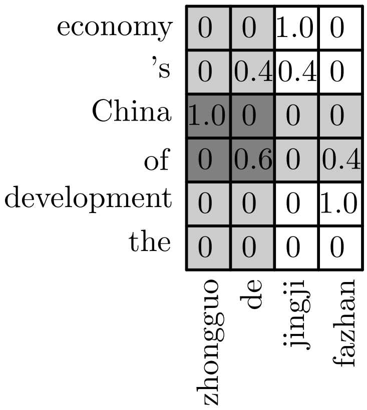
Unfortunately, this algorithm cannot be directly used to manipulate a weighted alignment matrix, which is a compact representation of all possible alignments. The major difference is that the "tight" phrase that has both boundary words aligned is not necessarily the smallest candidate in a weighted matrix. For example, in Figure 2(a), the "tight" target phrase corresponding to the source phrase "*zhongguo de*" is "*of China*". According to Och's algorithm, the target phrase "*China*" breaks the alignment consistency and therefore is not valid candidate. However, this is not true for using the weighted matrix shown in Figure 2(c). The target phrase "*China*" is treated as a "potential" candidate [1], although it might be assigned only a small fractional count (see Table 1).

Therefore, we enumerate all potential phrase pairs and calculate their fractional counts for eliminating less promising candidates. Figure 3 shows the algorithm for extracting phrases from a weighted matrix. The input of the algorithm is a source sentence f_1^J, a target sentence e_1^I, a weighted alignment matrix m, and a phrase length limit l (line 1). After initializing $\mathcal{R}$ that stores collected phrase pairs (line 2), we identify the corresponding target phrases for all possible source phrases (lines 3-5). Given a source phrase $f_{j_1}^{j_2}$, we find the lower and upper bounds of target positions (i.e., i_l and i_u) that have positive link probabilities (lines 6-8). For example, the lower bound is 3 and the upper bound is 5 for the source phrase "*zhongguo de*" in Figure 2(c). Finally, we enumerate all target phrases that allow for unaligned boundary words with varying phrase lengths (lines 9-14). Note that we need to ensure that $1 \leq i_1 \leq I$ and $1 \leq i_2 \leq I$ in lines 10-11, which are omitted for simplicity.

4.2 Calculating Relative Frequencies

To estimate the relative frequency of a phrase pair, we need to estimate how often it occurs in the training corpus. Given an n-best list, the fractional count of a phrase pair is the probability sum of the alignments with which the phrase pair is consistent. Obviously, it is unrealistic for a weighted alignment matrix to enumerate all possible alignments explicitly to calculate fractional counts. Instead, we resort to link probabilities to calculate

[1] By potential, we mean that the fractional count of a phrase pair is positive. Section 4.2 describes how to calculate fractional counts.



	zhongguo	de	jingji	fazhan
economy	0	0	1.0	0
's	0	0.4	0.4	0
China	1.0	0	0	0
of	0	0.6	0	0.4
development	0	0	0	1.0
the	0	0	0	0

Figure 4: An example of calculating fractional count. Given the phrase pair (“*zhongguo de*”, “*of China*”), we divide the matrix into three areas: inside (heavy shading), outside (light shading), and irrelevant (no shading).

counts efficiently. Equivalent to explicit enumeration, we interpret the fractional count of a phrase pair as the probability that it satisfies the two alignment consistency conditions (see Section 2).

Given a phrase pair, we divide the elements of a weighted alignment matrix into three categories: (1) *inside* elements that fall inside the phrase pair, (2) *outside* elements that fall outside the phrase pair while fall in the same row or the same column, and (3) *irrelevant* elements that fall outside the phrase pair while fall in neither the same row nor the same column. Figure 4 shows an example. Given the phrase pair (“*zhongguo de*”, “*of China*”), we divide the matrix into three areas: inside (heavy shading), outside (light shading), and irrelevant (no shading).

To what extent a phrase pair satisfies the alignment consistency is measured by calculating *inside* and *outside* probabilities. Although there are the same terms in the parsing literature, they have different meanings here. The inside probability indicates the chance that there is at least one word inside one phrase aligned to a word inside the other phrase. The outside probability indicates the chance that no words inside one phrase are aligned to a word outside the other phrase.

Given a phrase pair $(f_{j_1}^{j_2}, e_{i_1}^{i_2})$, we denote the inside area as $in(j_1, j_2, i_1, i_2)$ and the outside area as $out(j_1, j_2, i_1, i_2)$. Therefore, the inside probability of a phrase pair is calculated as

$$\alpha(j_1, j_2, i_1, i_2) = 1 - \prod_{(j,i) \in in(j_1,j_2,i_1,i_2)} \bar{p}_m(j, i) \quad (11)$$

target phrase	α	β	*count*
of China	1.0	0.36	0.36
of China ’s	1.0	0.36	0.36
China ’s	1.0	0.24	0.24
China	1.0	0.24	0.24
’s economy	0.4	0	0

Table 1: Some candidate target phrases of the source phrase “*zhongguo de*” in Figure 4, where α is inside probability, β is outside probability, and *count* is fractional count.

For example, the inside probability for (“*zhongguo de*”, “*of China*”) in Figure 4 is 1.0, which means that there always exists at least one aligned word pair inside.

Accordingly, the outside probability of a phrase pair is calculated as

$$\beta(j_1, j_2, i_1, i_2) = \prod_{(j,i) \in out(j_1,j_2,i_1,i_2)} \bar{p}_m(j, i) \quad (12)$$

For example, the outside probability for (“*zhongguo de*”, “*of China*”) in Figure 4 is 0.36, which means the probability that there are no aligned word pairs outside is 0.36.

Finally, we use the product of inside and outside probabilities as the fractional count of a phrase pair:

$$\begin{aligned} count(f_{j_1}^{j_2}, e_{i_1}^{i_2}) &= \alpha(j_1, j_2, i_1, i_2) \times \\ & \quad \beta(j_1, j_2, i_1, i_2) \end{aligned} \quad (13)$$

Table 1 lists some candidate target phrases of the source phrase “*zhongguo de*” in Figure 4. We also give their inside probabilities, outside probabilities, and fractional counts.

After collecting the fractional counts from the training corpus, we then use Eq. (2) to calculate relative frequencies in two translation directions.

Often, our approach extracts a large amount of phrase pairs from training corpus as we soften the alignment consistency constraint. To maintain a reasonable phrase table size, we discard any phrase pair that has a fractional count lower than a threshold t. During extraction, we first obtain a list of candidate target phrases for each source phrase, as shown in Table 1. Then, we prune the list according to the threshold t. For example, we only retain the top two candidates in Table 1 if $t = 0.3$. Note that we perform the pruning locally. Although it is more reasonable to prune a phrase table after accumulating all fractional counts from training corpus, such global pruning strategy usually leads to very large disk and memory requirements.

4.3 Calculating Lexical Weights

Recall that we need to obtain two translation probability tables $w(e|f)$ and $w(f|e)$ before calculating lexical weights (see Section 2). Following Koehn et al. (2003), we estimate the two distributions by relative frequencies from the training corpus annotated with weighted alignment matrices. In other words, we still use Eq. (3) but the way of calculating fractional counts is different now.

Given a source word f_j, a target word e_i, and a weighted alignment matrix, the fractional count $count(f_j, e_i)$ is $p_m(j, i)$. For NULL words, the fractional counts can be calculated as

$$count(f_j, e_0) = \prod_{i=1}^{I} \bar{p}_m(j, i) \quad (14)$$

$$count(f_0, e_i) = \prod_{j=1}^{J} \bar{p}_m(j, i) \quad (15)$$

For example, in Figure 4, $count$(*de*, *of*) is 0.6, $count$(*de*,*NULL*) is 0.24, and $count$(*NULL*,*of*) is 0.24.

Then, we adapt Eq. (4) to calculate lexical weight:

$$p_w(\tilde{e}|\tilde{f}, m) = \prod_{i=1}^{|\tilde{e}|} \Bigg(\bigg(\frac{1}{\{j|p_m(j,i) > 0\}} \times \sum_{\forall j: p_m(j,i)>0} p(e_i|f_j) \times p_m(j,i) \bigg) + p(e_i|f_0) \times \prod_{j=1}^{|\tilde{f}|} \bar{p}_m(j,i) \Bigg) \quad (16)$$

For example, for the target word "*of*" in Figure 4, the sum of aligned and unaligned probabilities is

$$\frac{1}{2} \times (p(\textit{of}|\textit{de}) \times 0.6 + p(\textit{of}|\textit{fazhan}) \times 0.4) + p(\textit{of}|\text{NULL}) \times 0.24$$

Note that we take link probabilities into account and calculate the probability that a target word translates a source NULL token explicitly.

5 Experiments

5.1 Data Preparation

We evaluated our approach on Chinese-to-English translation. We used the FBIS corpus (6.9M + 8.9M words) as the training data. For language model, we used the SRI Language Modeling Toolkit (Stolcke, 2002) to train a 4-gram model on the Xinhua portion of GIGAWORD corpus. We used the NIST 2002 MT evaluation test set as our development set, and used the NIST 2005 test set as our test set. We evaluated the translation quality using *case-insensitive* BLEU metric (Papineni et al., 2002).

To obtain weighted alignment matrices, we followed Venugopal et al. (2008) to produce n-best lists via GIZA++. We first ran GIZA++ to produce 50-best lists in two translation directions. Then, we used the refinement technique "grow-diag-final-and" (Koehn et al., 2003) to all 50×50 bidirectional alignment pairs. Suppose that p_{s2t} and p_{t2s} are the probabilities of an alignment pair assigned by GIZA++, respectively. We used $p_{s2t} \times p_{t2s}$ as the probability of the resulting symmetric alignment. As different alignment pairs might produce the same symmetric alignments, we followed Venugopal et al. (2008) to remove duplicate alignments and retain only the alignment with the highest probability. Therefore, there were 550 candidate alignments on average for each sentence pair in the training data. We obtained n-best lists by selecting the top n alignments from the 550-best lists. The probability of each alignment in the n-best list was re-estimated by re-normalization (Venugopal et al., 2008). Finally, these n-best alignments served as samples for constructing weighted alignment matrices.

After extracting phrase pairs from n-best lists and weighted alignment matrices, we ran Moses (Koehn et al., 2007) to translate the development and test sets. We used the simple distance-based reordering model to remove the dependency of lexicalization on word alignments for Moses.

5.2 Effect of Pruning Threshold

Our first experiment investigated the effect of pruning threshold on translation quality (BLEU scores on the test set) and the phrase table size (filtered for the test set), as shown in Figure 5. To save time, we extracted phrase pairs just from the first 10K sentence pairs of the FBIS corpus. We used 12 different thresholds: 0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, and 0.9. Obviously, the lower the threshold is, the more phrase pairs are extracted. When $t = 0.0001$, the number of phrase pairs used on the test set was 460,284

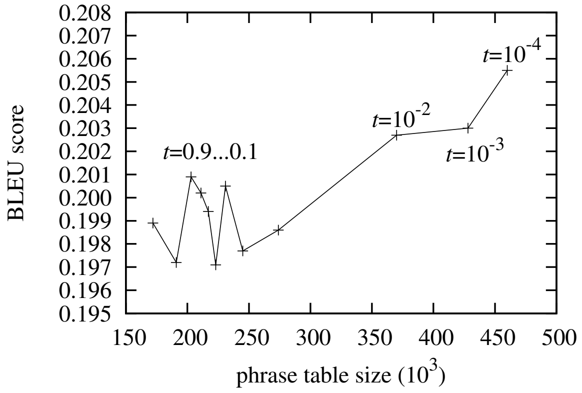


Figure 5: Effect of pruning threshold on translation quality and phrase table size.

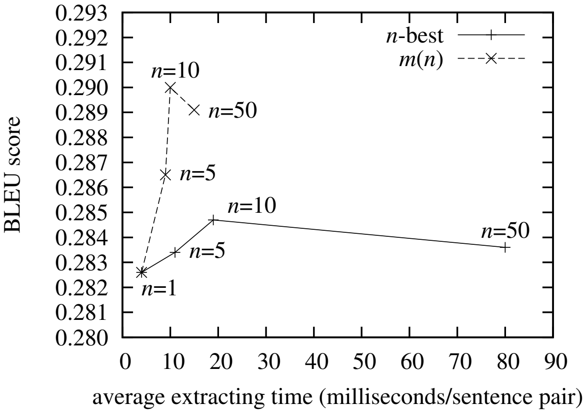


Figure 6: Comparison of n-best alignments and weighted alignment matrices. We use $m(n)$ to denote the matrices that take n-best lists as samples.

and the BLEU score was 20.55. Generally, both the number of phrase pairs and the BLEU score went down with the increase of t. However, this trend did not hold within the range [0.1, 0.9]. To achieve a good tradeoff between translation quality and phrase table size, we set $t = 0.01$ for the following experiments.

5.3 N-best lists Vs. Weighted Matrices

Figure 6 shows the BLEU scores and average extraction time using n-best alignments and weighted matrices, respectively. We used the entire training data for phrase extraction. When using 1-best alignments, Moses achieved a BLEU score of 0.2826 and the average extraction time was 4.19 milliseconds per sentence pair (see point $n = 1$). The BLEU scores rose with the increase of n for using n-best alignments. However, the score went down slightly when $n = 50$. This suggests that including more noisy alignments might be harmful. These improvements over 1-best alignments are not statistically significant. This finding failed to echo the promising results reported by Venogopal et al. (2008). We think that there are two possible reasons. First, they evaluated their approach on the IWSLT data while we used the NIST data. It might be easier to obtain significant improvements on the IWSLT data in which the sentences are shorter. Second, they used the hierarchical phrase-based system while we used the phrase-based system, which might be less sensitive to word alignments because the alignments inside the phrase pairs hardly have an effect.

When using weighted alignment matrices, we obtained higher BLEU scores than using n-best lists with much less extraction time. We achieved a BLEU score of 0.2901 when using the weighted matrices estimated from 10-best lists. The absolute improvement of 0.75 over using 1-best alignments (from 0.2826 to 0.2901) is statistically significant at $p < 0.05$ by using *sign-test* (Collins et al., 2005). Although the improvements over n-best lists are not always statistically significant, weighted alignment matrices maintain consistent superiority in both translation quality and extraction speed.

5.4 Comparison of Parameter Estimation

In theory, the set of phrase pairs extracted from n-best alignments is the subset of the set extracted from the corresponding weighted matrices. In practice, however, this is not true because we use the pruning threshold t to maintain a reasonable table size. Even so, the phrase tables produced by n-best lists and weighted matrices still share many phrase pairs.

Table 2 gives some statistics. We use $m(10)$ to represent the weighted matrices estimated from 10-best lists. "all" denotes the full phrase table, "shared" denotes the intersection of two tables, and "non-shared" denotes the complement. Note that the probabilities of "shared" phrase pairs are different for the two approaches. We obtained 6.13M and 6.34M phrase pairs for the test set by using 10-best lists and the corresponding matrices, respectively. There were 4.58M phrase pairs included by both tables. Note that the relative frequencies and lexical weights for the same phrase

method	shared		non-shared		all	
	phrases	BLEU	phrases	BLEU	phrases	BLEU
10-best	4.58M	**28.35**	1.55M	12.32	6.13M	28.47
$m(10)$	4.58M	**28.90**	1.76M	13.21	6.34M	29.01

Table 2: Comparison of phrase tables learned from n-best lists and weighted matrices. We use $m(10)$ to represent the weighted matrices estimated from 10-best lists. “all” denotes the full phrase table, “shared” denotes the intersection of two tables, and “non-shared” denotes the complement. Note that the probabilities of “shared” phrase pairs are different for the two approaches.

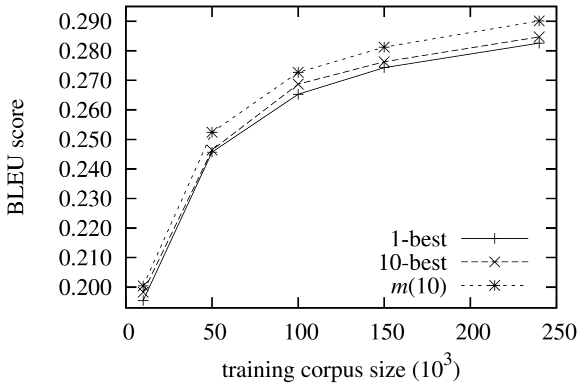


Figure 7: Comparison of n-best alignments and weighted alignment matrices with varying training corpus sizes.

pairs might be different in two tables. We found that using matrices outperformed using n-best lists even with the same phrase pairs. This suggests that our methods for parameter estimation make better use of noisy data. Another interesting finding was that using the shared phrase pairs achieved almost the same results with using full phrase tables.

5.5 Effect of Training Corpus Size

To investigate the effect of training corpus size on our approach, we extracted phrase pairs from n-best lists and weighted matrices trained on five training corpora with varying sizes: 10K, 50K, 100K, 150K, and 239K sentence pairs. As shown in Figure 7, our approach outperformed both 1-best and n-best lists consistently. More importantly, the gains seem increase when more training data are used.

	S↔E	F↔E	G↔E
Sentences	1.26M	1.29M	1.26M
Foreign words	33.16M	33.18M	29.58M
English words	31.81M	32.62M	31.93M

Table 3: Statistics of the Europarl training data. “S” denotes Spanish, “E” denotes English, “F” denotes French, “G” denotes German.

	1-best	10-best	$m(10)$
S→E	30.90	30.97	31.03
E→S	31.16	31.25	31.34
F→E	30.69	30.76	30.82
E→F	26.42	26.65	26.54
G→E	24.46	24.58	24.66
E→G	18.03	18.30	18.20

Table 4: BLEU scores (case-insensitive) on the Europarl data. “S” denotes Spanish, “E” denotes English, “F” denotes French, “G” denotes German.

5.6 Results on Other Language Pairs

To further examine the efficacy of the proposed approach, we scaled our experiments to large data with multiple language pairs. We used the Europarl training corpus from the WMT07 shared task. [2] Table 3 shows the statistics of the training data. There are four languages (Spanish, French, German, and English) and six translation directions (Foreign-to-English and English-to-Foreign). We used the “dev2006” data in the “dev” directory as the development set and the “test2006” data in the “devtest” directory as the test set. Both the development and test sets contain 2,000 sentences with single reference translations.

We tokenized and lowercased all the training, development, and test data. We trained a 4-gram language model using SRI Language Modeling Toolkit on the target side of the training corpus for each task. We ran GIZA++ on the entire training data to obtain n-best alignments and weighted matrices. To save time, we just used the first 100K sentences of each aligned training corpus to extract phrase pairs.

[2] http://www.statmt.org/wmt07/shared-task.html

Table 4 lists the case-insensitive BLEU scores of 1-best, 10-best, and m(10) on the Europarl data. Using weighted packed matrices continued to show advantage over using 1-best alignments on multiple language pairs. However, these improvements were very small and not significant. We attribute this to the fact that GIZA++ usually produces high quality 1-best alignments for closely-related European language pairs, especially when trained on millions of sentences.

6 Related Work

Recent studies has shown that SMT systems can benefit from making the annotation pipeline wider: using packed forests instead of 1-best trees (Mi et al., 2008), word lattices instead of 1-best segmentations (Dyer et al., 2008), and n-best alignments instead of 1-best alignments (Venugopal et al., 2008). We propose a compact representation of multiple word alignments that enables SMT systems to make a better use of noisy alignments.

Matusov et al. (2004) propose "cost matrices" for producing symmetric alignments. Kumar et al. (2007) describe how to use "posterior probability matrices" to improve alignment accuracy via a bridge language. Although not using the term "weighted matrices" directly, they both assign a probability to each word pair.

We follow Och and Ney (2004) to develop a new phrase extraction algorithm for weighted alignment matrices. The methods for calculating relative frequencies (Och and Ney, 2004) and lexical weights (Koehn et al., 2003) are also adapted for the weighted matrix case.

Many researchers (e.g., (Venugopal et al., 2003; Deng et al., 2008)) observe that softening the alignment consistency constraint help improve translation quality. For example, Deng et al. (2008) define a feature named "within phrase pair consistency ratio" to measure the degree of consistency. As each link is associated with a probability in a weighted matrix, we use these probabilities to evaluate the validity of a phrase pair.

We estimate the link probabilities by calculating relative frequencies over n-best lists. Niehues and Vogel (2008) propose a discriminative approach to modeling the alignment matrix directly. The difference is that they assign a boolean value instead of a probability to each word pair.

7 Conclusion and Future Work

We have presented a new structure called weighted alignment matrix that encodes the alignment distribution for a sentence pair. Accordingly, we develop new methods for extracting phrase pairs and estimating their probabilities. Our experiments show that the proposed approach achieves better translation quality over using n-best lists in less extraction time. An interesting finding is that our approach performs better than the baseline even they use the same phrase pairs.

Although our approach consistently outperforms using 1-best alignments for varying language pairs, the improvements are comparatively small. One possible reason is that taking n-best lists as samples sometimes might change alignment probability distributions inappropriately. A more principled solution is to directly model the weighted alignment matrices, either in a generative or a discriminative way. We believe that better estimation of alignment distributions will result in more significant improvements.

Another interesting direction is applying our approach to extracting translation rules with hierarchical structures such as hierarchical phrases (Chiang, 2007) and tree-to-string rules (Galley et al., 2006; Liu et al., 2006). We expect that these syntax-based systems could benefit more from our approach.

Acknowledgement

The authors were supported by Microsoft Research Asia Natural Language Processing Theme Program grant (2009-2010), High-Technology R&D Program (863) Project No. 2006AA010108, and National Natural Science Foundation of China Contract 60736014. Part of this work was done while Yang Liu was visiting the SMT group led by Stephan Vogel at CMU. We thank the anonymous reviewers for their insightful comments. We are also grateful to Stephan Vogel, Alon Lavie, Francisco Guzman, Nguyen Bach, Andreas Zollmann, Vamshi Ambati, and Kevin Gimpel for their helpful feedback.

References

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

Peter F. Brown, 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.

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

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL 2005*, pages 531–540, Ann Arbor, USA, June.

Yonggang Deng, Jia Xu, and Yuqing Gao. 2008. Phrase table training for precision and recall: What makes a good phrase and a good phrase pair? In *Proceedings of ACL/HLT 2008*, pages 81–88, Columbus, Ohio, USA, June.

Christopher Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL/HLT 2008*, pages 1012–1020, Columbus, Ohio, June.

Alexander Fraser and Daniel Marcu. 2007. Measuring word alignment quality for statistical machine translation. *Computational Linguistics, Squibs and Discussions*, 33(3):293–303.

Michel Galley, 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, Australia, July.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT/NAACL 2003*, pages 127–133, Edmonton, Canada, May.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL 2007 (poster)*, pages 77–80, Prague, Czech Republic, June.

Shankar Kumar, Franz J. Och, and Wolfgang Macherey. 2007. Improving word alignment with bridge languages. In *Proceedings of EMNLP 2007*, pages 42–50, Prague, Czech Republic, June.

Yang Liu, Qun Liu, and Shouxun Lin. 2005. Log-linear models for word alignment. In *Proceedings of ACL 2005*, pages 459–466, Ann Arbor, Michigan, June.

Yang Liu, 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, Australia, July.

Evgeny Matusov, Richard Zens, and Hermann Ney. 2004. Symmetric word alignments for statistical machine translation. In *Proceedings of COLING 2004*, pages 219–225, Geneva, Switzerland, August.

Haitao Mi, Liang Huang, and Qun Liu. 2008. Forest-based translation. In *Proceedings of ACL/HLT 2008*, pages 192–199, Columbus, Ohio, June.

Robert C. Moore, Wen-tau Yih, and Andreas Bode. 2006. Improved discriminative bilingual word alignment. In *Proceedings of COLING/ACL 2006*, pages 513–520, Sydney, Australia, July.

Jan Niehues and Stephan Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proceedings of WMT-3*, pages 18–25, Columbus, Ohio, USA, June.

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

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

Libin Shen, Jinxi Xu, and Ralph Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *Proceedings of ACL/HLT 2008*, pages 577–585, Columbus, Ohio, June.

Andreas Stolcke. 2002. Srilm - an extension language model modeling toolkit. In *Proceedings of ICSLP 2002*, pages 901–904, Denver, Colorado, September.

Ben Taskar, Simon Lacoste-Julien, and Dan Klein. 2005. A discriminative matching approach to word alignment. In *Proceedings of HLT/EMNLP 2005*, pages 73–80, Vancouver, British Columbia, Canada, October.

Ashish Venugopal, Stephan Vogel, and Alex Waibel. 2003. Effective phrase translation extraction from alignment models. In *Proceedings of ACL 2003*, pages 319–326, Sapporo, Japan, July.

Ashish Venugopal, Andreas Zollmann, Noah A. Smith, and Stephan Vogel. 2008. Wider pipelines: n-best alignments and parses in mt training. In *Proceedings of AMTA 2008*, pages 192–201, Waikiki, Hawaii, October.

Stephan Vogel and Hermann Ney. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of COLING 1996*, pages 836–841, Copenhagen, Danmark, August.