

# Alignment Model Adaptation for Domain-Specific Word Alignment

WU Hua, WANG Haifeng, LIU Zhanyi

Toshiba (China) Research and Development Center

5/F., Tower W2, Oriental Plaza

No.1, East Chang An Ave., Dong Cheng District

Beijing, 100738, China

{wuhua, wanghaifeng, liuzhanyi}@rdc.toshiba.com.cn

## Abstract

This paper proposes an alignment adaptation approach to improve domain-specific (in-domain) word alignment. The basic idea of alignment adaptation is to use out-of-domain corpus to improve in-domain word alignment results. In this paper, we first train two statistical word alignment models with the large-scale out-of-domain corpus and the small-scale in-domain corpus respectively, and then interpolate these two models to improve the domain-specific word alignment. Experimental results show that our approach improves domain-specific word alignment in terms of both precision and recall, achieving a relative error rate reduction of 6.56% as compared with the state-of-the-art technologies.

## 1 Introduction

Word alignment was first proposed as an intermediate result of statistical machine translation (Brown et al., 1993). In recent years, many researchers have employed statistical models (Wu, 1997; Och and Ney, 2003; Cherry and Lin, 2003) or association measures (Smadja et al., 1996; Ahrenberg et al., 1998; Tufis and Barbu, 2002) to build alignment links. In order to achieve satisfactory results, all of these methods require a large-scale bilingual corpus for training. When the large-scale bilingual corpus is not available, some researchers use existing dictionaries to improve word alignment (Ker and Chang, 1997). However, only a few studies (Wu and Wang, 2004) directly address the problem of domain-specific word alignment when neither the large-scale

domain-specific bilingual corpus nor the domain-specific translation dictionary is available.

In this paper, we address the problem of word alignment in a specific domain, in which only a small-scale corpus is available. In the domain-specific (in-domain) corpus, there are two kinds of words: general words, which also frequently occur in the out-of-domain corpus, and domain-specific words, which only occur in the specific domain. Thus, we can use the out-of-domain bilingual corpus to improve the alignment for general words and use the in-domain bilingual corpus for domain-specific words. We implement this by using alignment model adaptation.

Although the adaptation technology is widely used for other tasks such as language modeling (Iyer et al., 1997), only a few studies, to the best of our knowledge, directly address word alignment adaptation. Wu and Wang (2004) adapted the alignment results obtained with the out-of-domain corpus to the results obtained with the in-domain corpus. This method first trained two models and two translation dictionaries with the in-domain corpus and the out-of-domain corpus, respectively. Then these two models were applied to the in-domain corpus to get different results. The trained translation dictionaries were used to select alignment links from these different results. Thus, this method performed adaptation through result combination. The experimental results showed a significant error rate reduction as compared with the method directly combining the two corpora as training data.

In this paper, we improve domain-specific word alignment through statistical alignment model adaptation instead of result adaptation. Our method includes the following steps: (1) two word alignment models are trained using a small-scale in-domain bilingual corpus and a large-scale

out-of-domain bilingual corpus, respectively. (2) A new alignment model is built by interpolating the two trained models. (3) A translation dictionary is also built by interpolating the two dictionaries that are trained from the two training corpora. (4) The new alignment model and the translation dictionary are employed to improve domain-specific word alignment results. Experimental results show that our approach improves domain-specific word alignment in terms of both precision and recall, achieving a relative error rate reduction of 6.56% as compared with the state-of-the-art technologies.

The remainder of the paper is organized as follows. Section 2 introduces the statistical word alignment model. Section 3 describes our alignment model adaptation method. Section 4 describes the method used to build the translation dictionary. Section 5 describes the model adaptation algorithm. Section 6 presents the evaluation results. The last section concludes our approach.

## 2 Statistical Word Alignment

According to the IBM models (Brown et al., 1993), the statistical word alignment model can be generally represented as in Equation (1).

$$p(a | f, e) = \frac{p(a, f | e)}{\sum_{a'} p(a', f | e)} \quad (1)$$

In this paper, we use a simplified IBM model 4 (Al-Onaizan et al., 1999), which is shown in Equation (2). This simplified version does not take word classes into account as described in (Brown et al., 1993).

$$\begin{aligned} p(a, f | e) &= \sum_{(\tau, \pi)} \Pr(\tau, \pi | e) \\ &= \binom{m - \phi_0}{\phi_0} p_0^{m - 2\phi_0} p_1^{\phi_0} \cdot \\ &\quad \prod_{i=1}^l n(\phi_i | e_i) \cdot \prod_{j=1}^m t(f_j | e_{a_j}) \cdot \\ &\quad \left( \prod_{j=1, a_j \neq 0}^m ([j = h(a_j)] \cdot d_1(j - c_{\rho_{a_j}})) + \right. \\ &\quad \left. \prod_{j=1, a_j \neq 0}^m ([j \neq h(a_j)] \cdot d_{>1}(j - p(j))) \right) \end{aligned} \quad (2)$$

$l, m$  are the lengths of the target sentence and the source sentence respectively.

$j$  is the position index of the source word.

$a_j$  is the position of the target word aligned to the  $j^{\text{th}}$  source word.

$\phi_i$  is the fertility of  $e_i$ .

$p_1$  is the fertility probability for  $e_0$ , and  $p_0 + p_1 = 1$ .

$t(f_j | e_{a_j})$  is the word translation probability.

$n(\phi_i | e_i)$  is the fertility probability.

$d_1(j - c_{\rho_{a_j}})$  is the distortion probability for the head of each cept<sup>1</sup>.

$d_{>1}(j - p(j))$  is the distortion probability for the remaining words of the cept.

$h(i) = \min_k \{k : i = a_k\}$  is the head of cept  $i$ .

$p(j) = \max_{k < j} \{k : a_j = a_k\}$

$\rho_i$  is the first word before  $e_i$  with non-zero fertility. If  $|\{i' : \phi_{i'} > 0 \wedge 0 < i' < i\}| > 0$ ,

$\rho_i = \max\{i' : \phi_{i'} > 0 \wedge 0 < i' < i\}$ ; else  $\rho_i = 0$ .

$c_i = \frac{\sum_j [a_j = i] \cdot j}{\phi_i}$  is the center of cept  $i$ .

During the training process, IBM model 3 is first trained, and then the parameters in model 3 are employed to train model 4. During the testing process, the trained model 3 is also used to get an initial alignment result, and then the trained model 4 is employed to improve this alignment result. For convenience, we describe model 3 in Equation (3). The main difference between model 3 and model 4 lies in the calculation of distortion probability.

$$\begin{aligned} p(a, f | e) &= \sum_{(\tau, \pi)} \Pr(\tau, \pi | e) \\ &= \binom{m - \phi_0}{\phi_0} p_0^{m - 2\phi_0} p_1^{\phi_0} \cdot \\ &\quad \prod_{i=1}^l n(\phi_i | e_i) \cdot \prod_{i=1}^l \phi_i! \cdot \\ &\quad \prod_{j=1}^m t(f_j | e_{a_j}) \cdot \prod_{j: a_j \neq 0}^m d(j | a_j, l, m) \end{aligned} \quad (3)$$

<sup>1</sup> A cept is defined as the set of target words connected to a source word (Brown et al., 1993).

However, both model 3 and model 4 do not take the multiword cept into account. Only one-to-one and many-to-one word alignments are considered. Thus, some multi-word units in the domain-specific corpus cannot be correctly aligned. In order to deal with this problem, we perform word alignment in two directions (source to target, and target to source) as described in (Och and Ney, 2000). The GIZA++ toolkit<sup>2</sup> is used to perform statistical word alignment.

We use  $SG_1$  and  $SG_2$  to represent the bi-directional alignment sets, which are shown in Equation (4) and (5). For alignment in both sets, we use  $j$  for source words and  $i$  for target words. If a target word in position  $i$  is connected to source words in positions  $j_1$  and  $j_2$ , then  $A_i = \{j_1, j_2\}$ . We call an element in the alignment set an *alignment link*.

$$SG_1 = \{(A_j, i) \mid A_j = \{j \mid a_j = i, a_j \geq 0\}\} \quad (4)$$

$$SG_2 = \{(j, A_j) \mid A_j = \{i \mid i = a_j, a_j \geq 0\}\} \quad (5)$$

### 3 Word Alignment Model Adaptation

In this paper, we first train two models using the out-of-domain training data and the in-domain training data, and then build a new alignment model through linear interpolation of the two trained models. In other words, we make use of the out-of-domain training data and the in-domain training data by interpolating the trained alignment models. One method to perform model adaptation is to directly interpolate the alignment models as shown in Equation (6).

$$p(a \mid f, e) = \lambda \cdot p_I(a \mid f, e) + (1 - \lambda) \cdot p_O(a \mid f, e) \quad (6)$$

$p_I(a \mid f, e)$  and  $p_O(a \mid f, e)$  are the alignment model trained using the in-domain corpus and the out-of-domain corpus, respectively.  $\lambda$  is an interpolation weight. It can be a constant or a function of  $f$  and  $e$ .

However, in both model 3 and model 4, there are mainly three kinds of parameters: translation probability, fertility probability and distortion probability. These three kinds of parameters have their own interpretation in these two models. In order to obtain fine-grained interpolation models, we separate the alignment model interpolation into

three parts: translation probability interpolation, fertility probability interpolation and distortion probability interpolation. For these probabilities, we use different interpolation methods to calculate the interpolation weights. After interpolation, we replace the corresponding parameters in equation (2) and (3) with the interpolated probabilities to get new alignment models.

In the following subsections, we will perform linear interpolation for word alignment in the source to target direction. For the word alignment in the target to source direction, we use the same interpolation method.

#### 3.1 Translation Probability Interpolation

The word translation probability  $t(f_j \mid e_{a_j})$  is very important in translation models. The same word may have different distributions in the in-domain corpus and the out-of-domain corpus. Thus, the interpolation weight for the translation probability is taken as a variant. The interpolation model for  $t(f_j \mid e_{a_j})$  is described in Equation (7).

$$t(f_j \mid e_{a_j}) = \lambda_t(e_{a_j}) \cdot t_I(f_j \mid e_{a_j}) + (1 - \lambda_t(e_{a_j})) \cdot t_O(f_j \mid e_{a_j}) \quad (7)$$

The interpolation weight  $\lambda_t(e_{a_j})$  in (7) is a function of  $e_{a_j}$ . It is calculated as shown in Equation (8).

$$\lambda_t(e_{a_j}) = \left( \frac{p_I(e_{a_j})}{p_I(e_{a_j}) + p_O(e_{a_j})} \right)^\alpha \quad (8)$$

$p_I(e_{a_j})$  and  $p_O(e_{a_j})$  are the relative frequencies of  $e_{a_j}$  in the in-domain corpus and in the out-of-domain corpus, respectively.  $\alpha$  is an adaptation coefficient, such that  $\alpha \geq 0$ .

Equation (8) indicates that if a word occurs more frequently in a specific domain than in the general domain, it can usually be considered as a domain-specific word (Peñas et al., 2001). For example, if  $p_I(e_{a_j})$  is much larger than  $p_O(e_{a_j})$ , the word  $e_{a_j}$  is a domain-specific word and the interpolation weight approaches to 1. In this case, we trust more on the translation probability obtained from the in-domain corpus than that obtained from the out-of-domain corpus.

<sup>2</sup> It is located at <http://www.fjoch.com/GIZA++.html>.

### 3.2 Fertility Probability Interpolation

The fertility probability  $n(\phi_i | e_i)$  describes the distribution of the number of words that  $e_i$  is aligned to. The interpolation model is shown in (9).

$$n(\phi_i | e_i) = \lambda_n \cdot n_I(\phi_i | e_i) + (1 - \lambda_n) \cdot n_O(\phi_i | e_i) \quad (9)$$

Where,  $\lambda_n$  is a constant. This constant is obtained using a manually annotated held-out data set. In fact, we can also set the interpolation weight to be a function of the word  $e_i$ . From the word alignment results on the held-out set, we conclude that these two weighting schemes do not perform quite differently.

### 3.3 Distortion Probability Interpolation

The distortion probability describes the distribution of alignment positions. We separate it into two parts: one is the distortion probability in model 3, and the other is the distortion probability in model 4. The interpolation model for the distortion probability in model 3 is shown in (10). Since the distortion probability is irrelevant with any specific source or target words, we take  $\lambda_d$  as a constant. This constant is obtained using the held-out set.

$$d(j | a_j, l, m) = \lambda_d \cdot d_I(j | a_j, l, m) + (1 - \lambda_d) \cdot d_O(j | a_j, l, m) \quad (10)$$

For the distortion probability in model 4, we use the same interpolation method and take the interpolation weight as a constant.

## 4 Translation Dictionary Acquisition

We use the translation dictionary trained from the training data to further improve the alignment results. When we train the bi-directional statistical word alignment models with the training data, we get two word alignment results for the training data. By taking the intersection of the two word alignment results, we build a new alignment set. The alignment links in this intersection set are extended by iteratively adding word alignment links into it as described in (Och and Ney, 2000). Based on the extended alignment links, we build a translation dictionary. In order to filter the noise caused by the error alignment links, we only retain those translation pairs whose log-likelihood ratio scores (Dunning, 1993) are above a threshold.

Based on the alignment results on the

out-of-domain corpus, we build a translation dictionary  $D_1$  filtered with a threshold  $\delta_1$ . Based on the alignment results on a small-scale in-domain corpus, we build another translation dictionary  $D_2$  filtered with a threshold  $\delta_2$ .

After obtaining the two dictionaries, we combine two dictionaries through linearly interpolating the translation probabilities in the two dictionaries, which is shown in (11). The symbols  $f$  and  $e$  represent a single word or a phrase in the source and target languages. This differs from the translation probability in Equation (7), where these two symbols only represent single words.

$$p(f | e) = \lambda(e) \cdot p_I(f | e) + (1 - \lambda(e)) \cdot p_O(f | e) \quad (11)$$

The interpolation weight is also a function of  $e$ . It is calculated as shown in (12)<sup>3</sup>.

$$\lambda(e) = \frac{p_I(e)}{p_I(e) + p_O(e)} \quad (12)$$

$p_I(e)$  and  $p_O(e)$  represent the relative frequencies of  $e$  in the in-domain corpus and out-of-domain corpus, respectively.

## 5 Adaptation Algorithm

The adaptation algorithms include two parts: a training algorithm and a testing algorithm. The training algorithm is shown in Figure 1.

After getting the two adaptation models and the translation dictionary, we apply them to the in-domain corpus to perform word alignment. Here we call it *testing algorithm*. The detailed algorithm is shown in Figure 2. For each sentence pair, there are two different word alignment results, from which the final alignment links are selected according to their translation probabilities in the dictionary  $D$ . The selection order is similar to that in the competitive linking algorithm (Melamed, 1997). The difference is that we allow many-to-one and one-to-many alignments.

## 6 Evaluation

We compare our method with four other methods. The first method is described in (Wu and Wang, 2004). We call it ‘‘Result Adaptation (ResAdapt)’’.

<sup>3</sup> We also tried an adaptation coefficient to calculate the interpolation weight as in (8). However, the alignment results are not improved by using this coefficient for the dictionary.

<b>Input:</b> In-domain training data Out-of-domain training data
<ol style="list-style-type: none"> <li>(1) Train two alignment models <math>M_I^{st}</math> (source to target) and <math>M_I^{ts}</math> (target to source) using the in-domain corpus.</li> <li>(2) Train the other two alignment models <math>M_O^{st}</math> and <math>M_O^{ts}</math> using the out-of-domain corpus.</li> <li>(3) Build an adaptation model <math>M^{st}</math> based on <math>M_I^{st}</math> and <math>M_O^{st}</math>, and build the other adaptation model <math>M^{ts}</math> based on <math>M_I^{ts}</math> and <math>M_O^{ts}</math> using the interpolation methods described in section 3.</li> <li>(4) Train a dictionary <math>D_1</math> using the alignment results on the in-domain training data.</li> <li>(5) Train another dictionary <math>D_2</math> using the alignment results on the out-of-domain training data.</li> <li>(6) Build an adaptation dictionary <math>D</math> based on <math>D_1</math> and <math>D_2</math> using the interpolation method described in section 4.</li> </ol>
<b>Output:</b> Alignment models $M^{st}$ and $M^{ts}$ Translation dictionary $D$

Figure 1. Training Algorithm

<b>Input:</b> Alignment models $M^{st}$ and $M^{ts}$ , translation dictionary $D$ , and testing data
<ol style="list-style-type: none"> <li>(1) Apply the adaptation model <math>M^{st}</math> and <math>M^{ts}</math> to the testing data to get two different alignment results.</li> <li>(2) Select the alignment links with higher translation probability in the translation dictionary <math>D</math>.</li> </ol>
<b>Output:</b> Alignment results on the testing data

Figure 2. Testing Algorithm

The second method “Gen+Spec” directly combines the out-of-domain corpus and the in-domain corpus as training data. The third method “Gen” only uses the out-of-domain corpus as training data. The fourth method “Spec” only uses the in-domain corpus as training data. For each of the last three methods, we first train bi-directional alignment

models using the training data. Then we build a translation dictionary based on the alignment results on the training data and filter it using log-likelihood ratio as described in section 4.

## 6.1 Training and Testing Data

In this paper, we take English-Chinese word alignment as a case study. We use a sentence-aligned out-of-domain English-Chinese bilingual corpus, which includes 320,000 bilingual sentence pairs. The average length of the English sentences is 13.6 words while the average length of the Chinese sentences is 14.2 words.

We also use a sentence-aligned in-domain English-Chinese bilingual corpus (operation manuals for diagnostic ultrasound systems), which includes 5,862 bilingual sentence pairs. The average length of the English sentences is 12.8 words while the average length of the Chinese sentences is 11.8 words. From this domain-specific corpus, we randomly select 416 pairs as testing data. We also select 400 pairs to be manually annotated as held-out set (development set) to adjust parameters. The remained 5,046 pairs are used as domain-specific training data.

The Chinese sentences in both the training set and the testing set are automatically segmented into words. In order to exclude the effect of the segmentation errors on our alignment results, the segmentation errors in our testing set are post-corrected. The alignments in the testing set are manually annotated, which includes 3,166 alignment links. Among them, 504 alignment links include multiword units.

## 6.2 Evaluation Metrics

We use the same evaluation metrics as described in (Wu and Wang, 2004). If we use  $S_G$  to represent the set of alignment links identified by the proposed methods and  $S_C$  to denote the reference alignment set, the methods to calculate the precision, recall, f-measure, and alignment error rate (AER) are shown in Equation (13), (14), (15), and (16). It can be seen that the higher the f-measure is, the lower the alignment error rate is. Thus, we will only show precision, recall and AER scores in the evaluation results.

$$precision = \frac{|S_G \cap S_C|}{|S_G|} \quad (13)$$

$$recall = \frac{|S_G \cap S_C|}{|S_C|} \quad (14)$$

$$fmeasure = \frac{2 \times |S_G \cap S_C|}{|S_G| + |S_C|} \quad (15)$$

$$AER = 1 - \frac{2 \times |S_G \cap S_C|}{|S_G| + |S_C|} = 1 - fmeasure \quad (16)$$

### 6.3 Evaluation Results

We use the held-out set described in section 6.1 to set the interpolation weights. The coefficient  $\alpha$  in Equation (8) is set to 0.8, the interpolation weight  $\lambda_n$  in Equation (9) is set to 0.1, the interpolation weight  $\lambda_d$  in model 3 in Equation (10) is set to 0.1, and the interpolation weight  $\lambda_d$  in model 4 is set to 1. In addition, log-likelihood ratio score thresholds are set to  $\delta_1 = 30$  and  $\delta_2 = 25$ . With these parameters, we get the lowest alignment error rate on the held-out set.

Using these parameters, we build two adaptation models and a translation dictionary on the training data, which are applied to the testing set. The evaluation results on our testing set are shown in Table 1. From the results, it can be seen that our approach performs the best among all of the methods, achieving the lowest alignment error rate. Compared with the method “ResAdapt”, our method achieves a higher precision without loss of recall, resulting in an error rate reduction of 6.56%. Compared with the method “Gen+Spec”, our method gets a higher recall, resulting in an error rate reduction of 17.43%. This indicates that our model adaptation method is very effective to alleviate the data-sparseness problem of domain-specific word alignment.

Method	Precision	Recall	AER
Ours	0.8490	0.7599	0.1980
ResAdapt	0.8198	0.7587	0.2119
Gen+Spec	0.8456	0.6905	0.2398
Gen	0.8589	0.6576	0.2551
Spec	0.8386	0.6731	0.2532

Table 1. Word Alignment Adaptation Results

The method that only uses the large-scale out-of-domain corpus as training data does not

produce good result. The alignment error rate is almost the same as that of the method only using the in-domain corpus. In order to further analyze the result, we classify the alignment links into two classes: single word alignment links (SWA) and multiword alignment links (MWA). Single word alignment links only include one-to-one alignments. The multiword alignment links include those links in which there are multiword units in the source language or/and the target language. The results are shown in Table 2. From the results, it can be seen that the method “Spec” produces better results for multiword alignment while the method “Gen” produces better results for single word alignment. This indicates that the multiword alignment links mainly include the domain-specific words. Among the 504 multiword alignment links, about 60% of the links include domain-specific words. In Table 2, we also present the results of our method. Our method achieves the lowest error rate results on both single word alignment and multiword alignment.

Method	Precision	Recall	AER
Ours (SWA)	0.8703	0.8621	0.1338
Ours (MWA)	0.5635	0.2202	0.6833
Gen (SWA)	0.8816	0.7694	0.1783
Gen (MWA)	0.3366	0.0675	0.8876
Spec (SWA)	0.8710	0.7633	0.1864
Spec (MWA)	0.4760	0.1964	0.7219

Table 2. Single Word and Multiword Alignment Results

In order to further compare our method with the method described in (Wu and Wang, 2004). We do another experiment using almost the same-scale in-domain training corpus as described in (Wu and Wang, 2004). From the in-domain training corpus, we randomly select about 500 sentence pairs to build the smaller training set. The testing data is the same as shown in section 6.1. The evaluation results are shown in Table 3.

Method	Precision	Recall	AER
Ours	0.8424	0.7378	0.2134
ResAdapt	0.8027	0.7262	0.2375
Gen+Spec	0.8041	0.6857	0.2598

Table 3. Alignment Adaptation Results Using a Smaller In-Domain Corpus

Compared with the method “Gen+Spec”, our method achieves an error rate reduction of 17.86%

while the method “ResAdapt” described in (Wu and Wang, 2004) only achieves an error rate reduction of 8.59%. Compared with the method “ResAdapt”, our method achieves an error rate reduction of 10.15%.

This result is different from that in (Wu and Wang, 2004), where their method achieved an error rate reduction of 21.96% as compared with the method “Gen+Spec”. The main reason is that the in-domain training corpus and testing corpus in this paper are different from those in (Wu and Wang, 2004). The training data and the testing data described in (Wu and Wang, 2004) are from a single manual. The data in our corpus are from several manuals describing how to use the diagnostic ultrasound systems.

In addition to the above evaluations, we also evaluate our model adaptation method using the "refined" combination in Och and Ney (2000) instead of the translation dictionary. Using the "refined" method to select the alignments produced by our model adaptation method (AER: 0.2371) still yields better result than directly combining out-of-domain and in-domain corpora as training data of the "refined" method (AER: 0.2290).

#### 6.4 The Effect of In-Domain Corpus

In general, it is difficult to obtain large-scale in-domain bilingual corpus. For some domains, only a very small-scale bilingual sentence pairs are available. Thus, in order to analyze the effect of the size of in-domain corpus, we randomly select sentence pairs from the in-domain training corpus to generate five training sets. The numbers of sentence pairs in these five sets are 1,010, 2,020, 3,030, 4,040 and 5,046. For each training set, we use model 4 in section 2 to train an in-domain model. The out-of-domain corpus for the adaptation experiments and the testing set are the same as described in section 6.1.

# Sentence Pairs	Precision	Recall	AER
1010	0.8385	0.7394	0.2142
2020	0.8388	0.7514	0.2073
3030	0.8474	0.7558	0.2010
4040	0.8482	0.7555	0.2008
5046	0.8490	0.7599	0.1980

Table 4. Alignment Adaptation Results Using In-Domain Corpora of Different Sizes

# Sentence Pairs	Precision	Recall	AER
1010	0.8737	0.6642	0.2453
2020	0.8502	0.6804	0.2442
3030	0.8473	0.6874	0.2410
4040	0.8430	0.6917	0.2401
5046	0.8456	0.6905	0.2398

Table 5. Alignment Results Directly Combining Out-of-Domain and In-Domain Corpora

The results are shown in Table 4 and Table 5. Table 4 describes the alignment adaptation results using in-domain corpora of different sizes. Table 5 describes the alignment results by directly combining the out-of-domain corpus and the in-domain corpus of different sizes. From the results, it can be seen that the larger the size of in-domain corpus is, the smaller the alignment error rate is. However, when the number of the sentence pairs increase from 3030 to 5046, the error rate reduction in Table 4 is very small. This is because the contents in the specific domain are highly replicated. This also shows that increasing the domain-specific corpus does not obtain great improvement on the word alignment results. Comparing the results in Table 4 and Table 5, we find out that our adaptation method reduces the alignment error rate on all of the in-domain corpora of different sizes.

#### 6.5 The Effect of Out-of-Domain Corpus

In order to further analyze the effect of the out-of-domain corpus on the adaptation results, we randomly select sentence pairs from the out-of-domain corpus to generate five sets. The numbers of sentence pairs in these five sets are 65,000, 130,000, 195,000, 260,000, and 320,000 (the entire out-of-domain corpus). In the adaptation experiments, we use the entire in-domain corpus (5046 sentence pairs). The adaptation results are shown in Table 6.

From the results in Table 6, it can be seen that the larger the size of out-of-domain corpus is, the smaller the alignment error rate is. However, when the number of the sentence pairs is more than 130,000, the error rate reduction is very small. This indicates that we do not need a very large bilingual out-of-domain corpus to improve domain-specific word alignment results.

# Sentence Pairs (k)	Precision	Recall	AER
65	0.8441	0.7284	0.2180
130	0.8479	0.7413	0.2090
195	0.8454	0.7461	0.2073
260	0.8426	0.7508	0.2059
320	0.8490	0.7599	0.1980

Table 6. Adaptation Alignment Results Using Out-of-Domain Corpora of Different Sizes

## 7 Conclusion

This paper proposes an approach to improve domain-specific word alignment through alignment model adaptation. Our approach first trains two alignment models with a large-scale out-of-domain corpus and a small-scale domain-specific corpus. Second, we build a new adaptation model by linearly interpolating these two models. Third, we apply the new model to the domain-specific corpus and improve the word alignment results. In addition, with the training data, an interpolated translation dictionary is built to select the word alignment links from different alignment results. Experimental results indicate that our approach achieves a precision of 84.90% and a recall of 75.99% for word alignment in a specific domain. Our method achieves a relative error rate reduction of 17.43% as compared with the method directly combining the out-of-domain corpus and the in-domain corpus as training data. It also achieves a relative error rate reduction of 6.56% as compared with the previous work in (Wu and Wang, 2004). In addition, when we train the model with a smaller-scale in-domain corpus as described in (Wu and Wang, 2004), our method achieves an error rate reduction of 10.15% as compared with the method in (Wu and Wang, 2004).

We also use in-domain corpora and out-of-domain corpora of different sizes to perform adaptation experiments. The experimental results show that our model adaptation method improves alignment results on in-domain corpora of different sizes. The experimental results also show that even a not very large out-of-domain corpus can help to improve the domain-specific word alignment through alignment model adaptation.

## References

L. Ahrenberg, M. Merkel, M. Andersson. 1998. *A*

*Simple Hybrid Aligner for Generating Lexical Correspondences in Parallel Tests*. In Proc. of ACL/COLING-1998, pp. 29-35.

- Y. Al-Onaizan, J. Curin, M. Jahr, K. Knight, J. Lafferty, D. Melamed, F. J. Och, D. Purdy, N. A. Smith, D. Yarowsky. 1999. *Statistical Machine Translation Final Report*. Johns Hopkins University Workshop.
- P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. Mercer. 1993. *The Mathematics of Statistical Machine Translation: Parameter Estimation*. Computational Linguistics, 19(2): 263-311.
- C. Cherry and D. Lin. 2003. *A Probability Model to Improve Word Alignment*. In Proc. of ACL-2003, pp. 88-95.
- T. Dunning. 1993. *Accurate Methods for the Statistics of Surprise and Coincidence*. Computational Linguistics, 19(1): 61-74.
- R. Iyer, M. Ostendorf, H. Gish. 1997. *Using Out-of-Domain Data to Improve In-Domain Language Models*. IEEE Signal Processing Letters, 221-223.
- S. J. Ker and J. S. Chang. 1997. *A Class-based Approach to Word Alignment*. Computational Linguistics, 23(2): 313-343.
- I. D. Melamed. 1997. *A Word-to-Word Model of Translational Equivalence*. In Proc. of ACL 1997, pp. 490-497.
- F. J. Och and H. Ney. 2000. *Improved Statistical Alignment Models*. In Proc. of ACL-2000, pp. 440-447.
- A. Peñas, F. Verdejo, J. Gonzalo. 2001. *Corpus-based Terminology Extraction Applied to Information Access*. In Proc. of the Corpus Linguistics 2001, vol. 13.
- F. Smadja, K. R. McKeown, V. Hatzivassiloglou. 1996. *Translating Collocations for Bilingual Lexicons: a Statistical Approach*. Computational Linguistics, 22(1): 1-38.
- D. Tufis and A. M. Barbu. 2002. *Lexical Token Alignment: Experiments, Results and Application*. In Proc. of LREC-2002, pp. 458-465.
- D. Wu. 1997. *Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora*. Computational Linguistics, 23(3): 377-403.
- H. Wu and H. Wang. 2004. *Improving Domain-Specific Word Alignment with a General Bilingual Corpus*. In R. E. Frederking and K. B. Taylor (Eds.), Machine Translation: From Real Users to Research: 6<sup>th</sup> conference of AMTA-2004, pp. 262-271.