# A Semi-supervised Word Alignment Algorithm with Partial Manual Alignments

**Qin Gao, Nguyen Bach** and **Stephan Vogel**
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA, 15213
{qing, nbach, stephan.vogel}@cs.cmu.edu

## Abstract

We present a word alignment framework that can incorporate partial manual alignments. The core of the approach is a novel semi-supervised algorithm extending the widely used IBM Models with a constrained EM algorithm. The partial manual alignments can be obtained by human labelling or automatically by high-precision-low-recall heuristics. We demonstrate the usages of both methods by selecting alignment links from manually aligned corpus and apply links generated from bilingual dictionary on unlabelled data. For the first method, we conduct controlled experiments on Chinese-English and Arabic-English translation tasks to compare the quality of word alignment, and to measure effects of two different methods in selecting alignment links from manually aligned corpus. For the second method, we experimented with moderate-scale Chinese-English translation task. The experiment results show an average improvement of 0.33 BLEU point across 8 test sets.

## 1 Introduction

Word alignment is used in various natural language processing applications, and most statistical machine translation systems rely on word alignment as a preprocessing step. Traditionally the word alignment model is trained in an unsupervised manner, e.g. the most widely used tool GIZA++ (Och and Ney, 2003), which implements the IBM Models (Brown et. al., 1993) and the HMM model (Vogel et al., 1996). However, for language pairs such as Chinese-English, the word alignment quality is often unsatisfactory (Guzman et al., 2009). There has been increasing interest on using manual alignments in word alignment tasks. Ittycheriah and Roukos (2005) proposed to use only manual alignment links in a maximum entropy model. A number of semi-supervised word aligners are proposed (Blunsom and Cohn, 2006; Niehues and Vogel, 2008; Taskar et al., 2005; Liu et al., 2005; Moore, 2005). These approaches use held-out manual alignments to tune the weights for discriminative models, with the model parameters, model scores or alignment links from unsupervised word aligners as features. Also, several models are proposed to address the problem of improving generative models with small amount of manual data, including Model 6 (Och and Ney, 2003) and the model proposed by Fraser and Marcu (2006) and its extension called LEAF aligner (Fraser and Marcu, 2007). The approaches use labelled data to tune parameters to combine different components of the IBM Models.

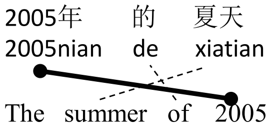

Figure 1: Partial and full alignments

An interesting question is, if we only have partial alignments of sentences, can we make use of them? Figure 1 shows the comparison of partial alignments (the bold link) and full alignments (both of the dashed and the bold links). A partial alignment of a sentence only provides a portion of links of the full alignment. Although it seems to be trivial, they actually convey different information. In the example, if the full alignment is given, we can assert *2005* is only aligned to *2005nian*, not to *de* or *xiatian*, but if only the partial alignment is given we cannot make such assertion.

Partial alignments can be obtained from various sources, for example, we can fetch them by manually correcting unsupervised alignments, by simple heuristics such as dictionaries of technical

*Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pages 1–10,
Uppsala, Sweden, 11-16 July 2010. ©2010 Association for Computational Linguistics

terms, by rule-based alignment systems that have high accuracy but low recall rate. The functionality is considered useful in many scenarios. For example, the researchers can analyse the alignments generated by GIZA++ and fix common error patterns, and perform training again. On another way, an application can combine active learning (Arora et al., 2009) and crowdsourcing, asking non-expertise such as workers of Amazon Mechanical Turk to label crucial alignment links that can improve the system with low cost, which is now a promising methodology in NLP areas (Callison-Burch, 2009).

In this paper, we propose a semi-supervised extension of the IBM Models that can utilize partial alignment links. More specifically, we are seeking answers for the following questions:

- Given the partial alignment of a sentence, how to find the most probable alignment that is consistent with the partial alignment.
- Given a set of partially aligned sentences, how to get the parameters that maximize the likelihood of the sentence pairs with alignments consistent with the partial alignments
- Given a set of partially aligned sentences, with conflicting partial alignments, how to answer the two questions above.

In the proposed approach, the manual partial alignment links are treated as ground truth, therefore, they will be fixed. However, for all other links we make no additional assumption. When using manual alignments, there can be links conflicting with each other. These conflicting evidences are treated as options and the generative model will choose the most probable alignment from them. An efficient training algorithm for fertility-based models is proposed. The algorithm manipulates the Moving and Swapping matrices used in the hill-climbing algorithm (Och and Ney, 2003) to rule out inconsistent alignments in both E-step and M-step of the training.

A similar attempt has been made by Callison-Burch et al. (2004), where the authors interpolate the parameters estimated by sentence-aligned and word-aligned corpus. Our approach is different from their method that we do not require fully aligned data and we do not need to interpolate two parameter sets. All the training is done within a unified framework. Our approach is also different from LEAF (Fraser and Marcu, 2007) and Model 6 (Och and Ney, 2003) that we do not use these additional links to tune additional parameters to combine model components, as a result, it is not limited to fully aligned corpus.

A question may raise why the proposed method is superior over using the partial alignment links as features in discriminative aligners? There are three possible explanations. First, the method preserves the power of the generative model in which the algorithm utilizes large amount of unlabeled data. More importantly, the additional information can propagate over the whole corpus through better estimation of model parameters. In contrast, if we use the alignment links in discriminative aligners as a feature, one link can only affect the particular word, or at most the sentence. Second, although the discriminative word alignment methods provide flexibility to utilize labeled data, most of them still rely on generative aligners. Some rely on the model parameters of the IBM Models (Liu et al., 2005; Blunsom and Cohn, 2006), others rely on the alignment links from GIZA++ as features or as training data (Taskar et al., 2005), or use both the model parameters and the alignment links (Niehues and Vogel, 2008). Therefore, improving the generative aligner is still important even when using discriminative aligners. Third, these methods require full alignment of sentences to provide positive (aligned) and negative (non-aligned) information, which limits the availability of data (Niehues and Vogel, 2008).

The proposed method has been successfully applied on various tasks, such as utilizing manual alignments harvested from Amazon Mechanical Turk (Gao and Vogel, 2010), and active learning methods for improving word alignment (Ambati et al., 2010). This paper provides the detailed algorithm of the method and controlled experiments to demonstrate its behavior.

The paper is organized as follows, in section 2 we describe the proposed model as well as the modified training algorithm. Section 3 presents two approaches of obtaining manual alignment links, The experimental results will be shown in section 4. We conclude the paper in section 5.

## 2 Semi-supervised word alignment

### 2.1 Problem Setup

The IBM Models (Brown et. al., 1993) are a series of generative models for word alignment. GIZA++ (Och and Ney, 2003) is the most widely used implementation of the IBM Models and the

HMM model (Vogel et al., 1996). Given two strings from target and source languages $f_1^J = f_1, \cdots, f_j, \cdots f_J$ and $e_1^I = e_1, \cdots, e_i, \cdots e_I$, an alignment of the sentence pair is defined as $a_1^J = [a_1, a_2, \cdots, a_J], a_j \in [0, I]$. The IBM Models assume all the target words must be covered exactly once (Brown et. al., 1993). We try to model $P(f_1^J|e_1^I)$, which is the probability of observing source sentence given target sentence $e_1^I$. In statistical models a hidden alignment variable is introduced, so that we can write the probability as $P(f_1^J|e_1^I) = \sum_{a_1^J} Pr(f_1^J, a_1^J|e_1^J, \theta)$, where $Pr(\cdot)$ is the estimated probability given the parameter set $\theta$. The IBM Models define several different set of parameters, from Model 1 to Model 5. Starting from Model 3, the fertility model is introduced.

EM algorithm is employed to estimate the model parameters of the IBM Models. In E-step, it is possible to obtain sufficient statistics from all possible alignments with simplified formulas for simple models such as Model 1 and Model 2. Meanwhile for fertility-based models, enumerating all possibilities is NP-complete and hence it cannot be carried out for long sentences. A solution is to explore only the "neighbors" of Viterbi alignments. However, obtaining Viterbi alignments itself is NP-complete for these models. In practice, a greedy algorithm is employed to find a local optimal alignments based on Viterbi alignments generated by simpler models.

First, we define the neighbor alignments of $a$ as the set of alignments that differ by one of the two operators from the original "**center alignment**".

- Move operator $m_{[i,j]}$, that changes $a_j := i$, i.e. arbitrarily set word $f_j$ in source sentence to align to word $f_i$ in target sentence.
- Swap operator $s_{[j_1,j_2]}$ that exchanges $a_{j_1}$ and $a_{j_2}$.

We denote the **neighbor alignments** set of current center alignment $a$ as $nb(a)$. In each step of hill-climbing algorithm, we find the alignment $b(a)$ in $nb(a)$, s.t. $b(a) = \arg\max_{a' \in nb(a)} p(a'|e, f)$, and update the current center alignment. The algorithm iterates until there is no update could be made. The statistics of the neighbor alignments of the final center alignment will be collected for normalization step (M-step). The algorithm is greedy, so a reasonable start point is important. In practice GIZA++ uses Model 2 or HMM to generate the **seed alignment**.

To improve the speed of hill climbing, GIZA++ caches the cost of all possible move and swap operations in two matrices. In the so called Moving Matrix $M$, the element $M_{ij}$ stores the likelihood difference of a move operator $a_j = i$:

$$M_{ij} = \frac{Pr(m_{[i,j]}(a)|e, f)}{Pr(a|e, f)} \cdot (1 - \delta(a_j, i)) \quad (1)$$

and in the Swapping Matrix $S$, the element $S_{jj'}$ stores the likelihood difference of a swap operator between $a_j$ and $a_{j'}$:

$$S_{jj'} = \begin{cases} \frac{Pr(S_{[j,j']}(a)|e,f)}{Pr(a|e,f)} \cdot (1 - \delta(a_j, a_{j'})) & \text{if } j < j' \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The matrices will be updated whenever an operator is made, but the update is limited to the rows and columns involved in the operator.

We define a **partial alignment** of a sentence pair $(f_1^J, e_1^I)$ as $\alpha_I^J = \{(i,j), 0 \leq i < I, 0 \leq j < J\}$, note that the partial alignment does not assume 1-to-N restriction on either side, and the word from neither source nor target side need to be covered with links. If an index is missing, it does not mean the word is aligned to the empty word. Instead it just means no information is provided. We use a link $(0, j)$ or $(i, 0)$ to explicitly represent the information that word $f_j$ or $e_i$ is aligned to the empty word.

In order to find *the most probable alignment that is consistent the partial alignments*, we treat the partial alignment as constraints, i.e. for an alignment $a_1^J = [a_1, a_2, \cdots, a_j]$ on the sentence pair $f_1^J, e_1^I$, the translation probability $Pr(f_1^J, a_1^J|e_1^I, \alpha_I^J)$ will be zero if the alignment is inconsistent with the partial alignments.

$$Pr(f_1^J|e_1^I, a_1^J, \alpha_I^J) = \begin{cases} 0, a_1^J \text{is inconsistent with} \alpha_I^J \\ Pr(f_1^J|e_1^I, a_1^J, \theta), \text{otherwise} \end{cases} \quad (3)$$

Under the constraints of the IBM Models, there are two situations that $a_1^J$ is inconsistent with $\alpha_I^J$:

1. Target word misalignment: The IBM Models assume one target word can only be aligned to one source word. Therefore, if the target word $f_j$ aligns to a source word $e_i$, while the constraint $\alpha_I^J$ suggests $f_j$ should be aligned to $e_{i'}$, the alignment violates the constraint and thus is considered inconsistent.

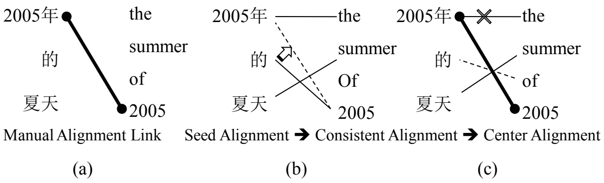

Figure 2: Illustration of Algorithm 1

2. Source word to empty word misalignment: Since one source word can be aligned to multiple target words, it is hard to constrain the alignments of source words. However, if a source word is aligned to the empty word, it cannot be aligned to any concrete target word.

However, we are facing the problem of conflicting evidences. The problem is not necessarily caused by errors in manual alignments, but the assumption of the IBM Models that one target word can only be aligned to one source word. This assumption causes multiple alignment links from one target word conflict with each other. In this case, we relax the constraints of situation 1 that if the alignment link $a_{j^*}$ is consistent with any target-to-source links $(i, j)$ that $j = j^*$, it will be considered consistent. Also, we arbitrarily assign the source word to empty word constraints higher priorities than other constraints.

In EM algorithm, to ensure the final model be *marginalized* on the fixed alignment links, and the final Viterbi alignment is *consistent* with the fixed alignment links, we need to guarantee that no statistics from inconsistent alignments be collected into the sufficient statistics. On fertility-based models, we have to make sure:

1. The hill-climbing algorithm outputs alignment links consistent with the fixed alignment links.
2. The count collection algorithm rules out all the inconsistent statistics.

With the constrained hill-climbing algorithm and count collection algorithm which will be described below, the above two criteria are satisfied.

### 2.2 Constrained hill-climbing algorithm

Algorithm 1 shows the algorithm outline of constrained hill-climbing. First, similar to the original hill-climbing algorithm described above, HMM (or Model 2) is used to obtain a seed alignment. To ensure the resulting center alignment be consistent with manual alignment, we need to split the hill-climbing algorithm into two stages, i.e. optimize towards the constraints and towards the optimal alignment under the constraints.

**Algorithm 1** Constrained Hill-Climbing

```
 1: Calculate the seed alignment $a_0$ using HMM model
 2: while $ic(a_0) > 0$ do
 3:    if $\{a : ic(a) < ic(a_0)\} = \emptyset$ then
 4:       break
 5:    end if
 6:    $a_0 := \arg\max_{a \in nb(a_0), ic(a) < ic(a_0)} Pr(f|e, a)$
 7: end while
 8: $M_{ij} := -1$ if $(i, j) \notin \alpha_I^J$ or $(i, 0) \in \alpha_I^J$
 9: loop
10:    $S_{jj'} := -1$ if $(j, a_{j'}) \notin \alpha_I^J$ or $(j', a_j) \notin \alpha_I^J$
11:    $M_{i_1 j_1} = \arg\max M_{ij}$ ; $S_{j_1 j'_1} = \arg\max S_{ij}$
12:    if $M_{i_1 j_1} \leq 1$ and $S_{j_1 j'_1} \leq 1$ then
13:       Break
14:    end if
15:    if $M_{i_1 j_1} > S_{j_1 j'_1}$ then
16:       Update $M_{i_1 *}, M_{j_1 *}, M_{* i_1}, M_{* j_1}$
          and $S_{i_1 *}, S_{j_1 *}, S_{* i_1}, S_{* j_1}$, set $a_0 := M_{i_1 j_1}(a_0)$
17:    else
18:       Update $M_{j_1 *}, M_{j'_1 *}, M_{* j_1}, M_{* j'_1}$
          and $S_{j'_1 *}, S_{j_1 *}, S_{* j'_1}, S_{* j_1}$, set $a_0 := S_{j_1 j'_1}(a_0)$
19:    end if
20: end loop
21: Return $a_0$
```

From a seed alignment, we first try to move the alignment towards the constraints by choosing a move or swap operator that:

1. has highest likelihood among alignments generated by other operators, excluding the original alignment,
2. eliminates at least one inconsistent link.

The first step reflects in line 2 through 7 in the algorithm, where we use $ic(\cdot)$ to denote the total number of inconsistent links in the alignment, and $nb(\cdot)$ to denote the neighbor alignments.

We iteratively update the alignment until no additional inconsistent link can be removed. The algorithm implies that we force the seed alignment to become closer to the constraints while trying to find the best consistent alignment. Figure 2 demonstrates the idea, given the manual alignment link shown in (a), and the seed alignment shown as solid links in (b), we move the inconsistent link to the dashed link by a move operation.

After we find the consistent alignment, we proceed to optimize towards the optimal alignment within the constraints. The algorithm sets the cells to negative if the corresponding operations are not allowed. The Moving matrix only need to be updated once, as in line 8 of the algorithm. Whereas the swapping matrix need to be updated every iteration, Since once the alignment is updated, the possible violations will also change. This is done in line 10.

If source words $i_k$ are aligned to the empty word, we set $M_{i_k,j} = -1, \forall j$, as shown in line 8. The swapping matrix does not need to be modified in this case because the swapping operator will not introduce new links. Again, Figure 2 demonstrates the optimization step in (c), two move operators or one swap operator can move the link marked with cross to the dashed line, which can be a better alignment.

Because the cells that can lead to violations are set to negative, the operators will never be picked in line 11, therefore we effectively ensure the consistency of the final center alignment.

The algorithm will end when no better update can be made (line 12 through 14), otherwise, we pick the new update with highest likelihood as new center alignment and update the cells in the Moving and Swapping matrices that will be affected by the update. Line 15 through line 19 perform the operation.

### 2.3 Count Collection

After finding the center alignment, we collect counts from the neighbor alignments so that the M-step can normalize the counts to produce the model parameters for the next step. All statistics from inconsistent alignments are ruled out to ensure the final sufficient statistics marginalized on the fixed alignment links. Similar to the constrained hill climbing algorithm, we can manipulate the Moving/Swapping matrices to effectively exclude inconsistent alignments. We just need to bypass all the cells whose values are negative, i.e. represent inconsistent alignments.

By combining the constrained EM algorithm and the count collection, the Viterbi alignment is *guaranteed to be consistent* with the fixed alignment links, and the sufficient statistics is *guaranteed to contain no statistics from inconsistent alignments*.

### 2.4 Training scheme

We extend the multi-thread GIZA++ (Gao and Vogel, 2008) to load the alignments from a modified corpus file. The links are appended to the end of each sentence in the corpus file in the form of indices pairs, which will be read by the aligner during training. In practice, we first training unconstrained models up to Model 4, and then switch to constrained Model 4 and continue training for several iterations, the actual number of training order is: 5 iterations of Model 1, 5 iterations of HMM, 3 iterations of Model 3, 3 iterations of unconstrained Model 4 and 3 iterations of constrained Model 4. Because here we actually have more Model 4 iterations, to make the comparison fair, in all the experiments below we perform 6 iterations of Model 4 in the baseline systems.

## 3 Obtaining alignment links

Given the algorithm described in the Section 2, we still face the problem of obtaining alignment links to constrain the system. In this section, we describe two approaches to obtain the links, the first is to resort to human labels, while the second applies high-precision-low-recall heuristic-based aligner on large unsupervised corpus.

### 3.1 Using manual alignment links

Using manual alignment links is simple and straight-forward, however the problem is how to select links for human to label given that labelling the whole corpus is impossible. We propose two link selectors, the first is the random selector in which every links in the manual alignment has equal probability of being selected. Obviously, the random selecting method is far from optimal because it pays no attention on the quality of existing links. In order to demonstrate that by selecting links carefully we can achieve better alignment quality with less manual alignment links, we propose the second selector based on disagreements of alignments from two directions. We first classify the source and target words $f_j$ and $e_i$ into three categories. Use $f_j$ as an example, the categories are:

- $C1$: $f_j$ aligns to $e_i, i > 0$ in $e \rightarrow f$,[1] but in reversed direction $e_i$ does not align to $f_j$ but to another word.
- $C2$: $f_j$ aligns to $e_i, i > 0$, in $f \rightarrow e$, but in reversed direction ($e \rightarrow f$), $f_j$ aligns to the empty word.
- $C3$: no word aligns to $f_j$, in $f \rightarrow e$, but in reversed direction $f_j$ aligns to $e_i, i > 0$.[2]

The criteria of $e_i$ are the same as $f_j$ after swapping the definitions of "source" and "target".

We prioritize the links $\alpha_I^J = (i, j)$ by looking at the classes of the source/target words. The order of

[1] Recall that $f_j$ can align to only one word.

[2] This class is different from $C1$ that whether $e_i$ aligns to concrete words or the empty word.

| Order | Criterion | Order | Criterion |
|---|---|---|---|
| 1 | $f_j \in C1$ | 5 | $e_i \in C2$ |
| 2 | $f_j \in C2$ | 4 | $e_i \in C1$ |
| 3 | $f_j \in C3$ | 6 | $e_i \in C3$ |

Table 1: The priorities of alignment links

priorities is shown in Table 1. All the links not in the six classes will have the lowest priorities. The links with higher priorities will be selected first, but the order of two links in a same priority class is not defined and they will be selected randomly.

### 3.2 Using heuristics on unlabelled data

Another possible way of getting alignment links is to make use of heuristics to generate high-precision-low-recall links and feed them into the aligner. The heuristics can be number mapping, person name translator or more sophisticated methods such as alignment confidence measure (Huang, 2009). In this paper we propose to use manual dictionaries to generate alignment links.

First we filter out from the dictionary the entries with high frequency in the source side, and then build an aligner based on it. The aligner output links between words if them match an entry in the dictionary. The method can be applied on large unlabelled corpus and generate large number of links, after that we use the links as manual alignment links in proposed method.

The readers may notice that GIZA++ supports utilizing manual dictionary as well, however it is different from our method. The dictionary is used in GIZA++ only in the initialization step of Model 1, where only the statistics of the word pairs appeared in the dictionary will be collected and normalized. Given the fact that Model 1 converges to global optimal, the effect will fade out after several iterations. In contrast, our method impose a hard constraint on the alignments. Also, our method can be used side-by-side with the method in GIZA++.

## 4 Experiments

### 4.1 Experiments on manual link selectors

We designed a set of controlled experiments to show that the algorithm acts as desired. Particularly, with a number of manual alignment links fed into the aligner, we should be able to correct more misaligned alignment links than the manual alignment links through better alignment models. Also, carefully selected alignment links should outperform randomly selected alignment links.

We used Chinese-English and Arabic-English manually aligned corpus in the experiments. Table 2 shows the statistics of the corpora:

| | Number of Sentences | Num. of Words | | Alignment Links |
|---|---|---|---|---|
| | | Source | Target | |
| Ch-En | 21,863 | 424,683 | 524,882 | 687,247 |
| Ar-En | 29,876 | 630,101 | 821,938 | 830,349 |

Table 2: Corpus statistics of the corpora

First the corpora is trained as unlabelled data to serve as baselines, and then we feed a portion of alignment links into the proposed aligner. We experimented with different methods of choosing alignment links and adjust the number of links visible to the aligner. Because of the limitations of the IBM Models, such as no N-to-1 alignments, the manual alignment is not reachable from either direction. We then define the best alignment that the IBM Models can express "*oracle alignment*", which can be obtained by dropping all N-to-1 links from manual alignment. Also, to show the upper-bound performance, we feed all the manual alignment links to our aligner, and call the alignment "*force alignment*". Table 3 shows the alignment qualities of oracle alignments and force alignments of both systems. For force alignments, we show the scores with and without implicit empty links derived from the manual alignment.[3] The oracle alignments are the performance upper-bounds of all aligners under IBM Model's 1-to-N assumption. The result from Table 3 shows that, if we include the derived empty links, the force alignments are close to the oracle results. Then the question is how fast we can approach the upper-bound.

To answer the question, we gradually increase the number of links being fed into the aligner. In these experiments the seeds for random number generator are fixed so that the links selected in later experiments are always superset of that of earlier experiments. The comparison of the alignment quality is shown in Figure 3 and 4. To show the actual improvement brought in by the algorithm instead of the manual alignment links themselves, we compare the alignment results of the proposed method with directly fixing the alignments from original GIZA++ training. By fixing alignments we mean that first the conventional

[3] We can derive empty links if one word has no alignment link from the full alignment we have access to.

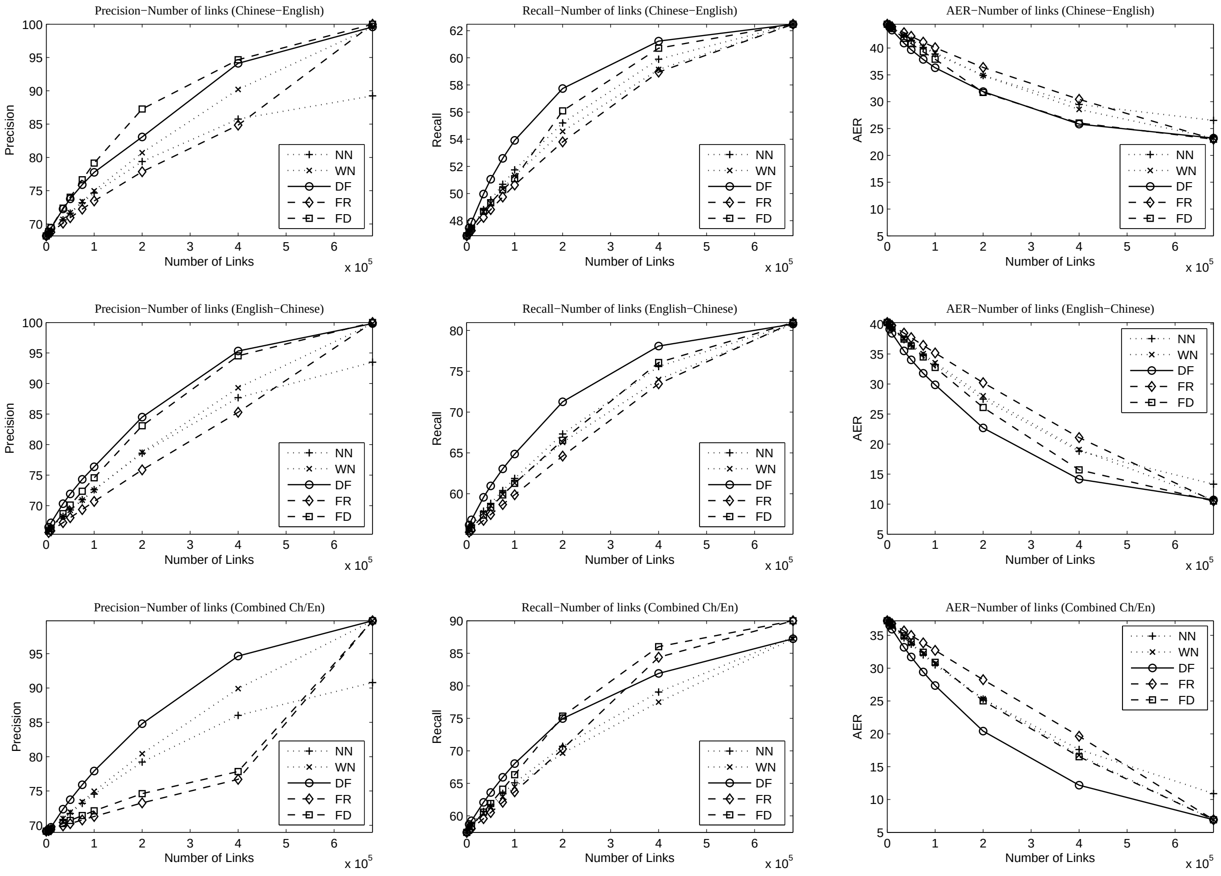

Figure 3: Alignment qualities of Chinese-English word alignment, NN: Random selector without empty links, WN: Random seletor with empty links, DF: Disagreement selector, FR: Directly fixing the alignments with random selector, FD: Directly fixing the alignments with disagreement selector. Each row shows the precision, recall and AER when applying different number of manual alignment links. The three rows are for Chinese-English, English-Chinese and heuristically symmetrized alignments (grow-diag-final-and) accordingly.

GIZA++ training is performed and then we add the manual alignment links to the resulting alignment. In case that the 1-to-N restriction of the IBM Models is violated, we keep the manual alignment links and remove the links from GIZA++.

We show the results as FR (dashed curves with diamond markers) and FD (dashed curves with square markers) in the plots, corresponding to alignments selected from the random link selector and the disagreement-based link selector. These two curves serve as baseline, and the gaps between the FR curves and the WN curves (dotted curves with cross markers) and the gaps between the FD curves and the DF curves (solid curves) show the amount of improvement we achieved using the method in addition to the manual alignment links. Therefore, they represent the effectiveness of the proposed alignment approach. Also the gaps between DF and WN curves indicate the differences in the performance of two link selectors.

The plots illustrate that when the number of links is small, the WN and DF curves are always higher than the FR/FD curves. It proves that our system does not just fix the links provided by manual alignments, instead the information propagates to other links. The largest gap between FD and DF is **8% absolute** in combined alignment of Chinese-English system with 200,000 manual alignment links. Also, we can see that the disagreement-based link selector (DF) always outperform the random selector (WN). It suggest that, if we want to harvest manual alignment links, it is possible to apply active learning method to minimize the user labelling effort while maximizing the improvement on word alignment qualities. Especially, notice that in the lower parts

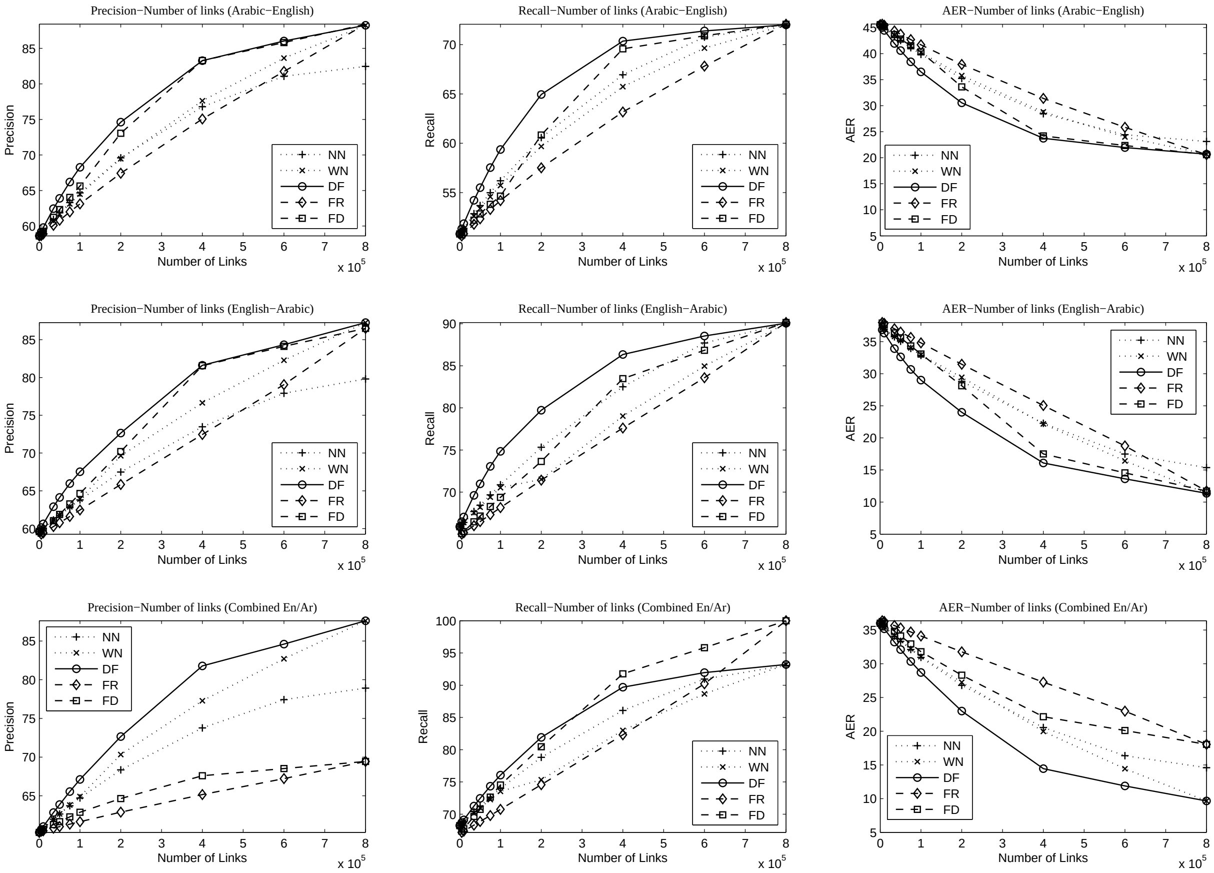

Figure 4: Alignment qualities of Arabic-English word alignment, NN: Random selector without empty links, WN: Random selector with empty links, DF: Disagreement selector, FR: Directly fixing the alignments with random selector, FD: Directly fixing the alignments with disagreement selector. Each row shows the precision, recall and AER when applying different number of manual alignment links. The three rows are for Arabic-English, English-Arabic and heuristically symmetrized alignments (grow-diag-final-and) accordingly.

of the curves, with a small number of manual alignment links, we can already improve the alignment quality by a large gap. This observation can benefit low-resource word alignment tasks.

## 4.2 Experiment on using heuristics

The previous experiment shows the potential of using the method on manual aligned corpus, here we demonstrate another possible usage of the proposed method that uses heuristics to generate high-precision-low-recall links. We use LDC Chinese-English dictionary as an example. The entries with single Chinese character and more than six English words are filtered out. The heuristic-based aligner yields alignment that has 79.48% precision and 17.36% recall rate on the test set we used in 4.1. By applying the links as manual links, we run proposed method on the same Chinese-English test data presented in 4.1, and the results of alignment qualities are shown in 5. As we can see, the AER reduced by 1.64 from 37.23 to 35.61 on symmetrized alignment.

We also experimented with translation tasks with moderate-size corpus. We used the corpus LDC2006G05 with 25 million words. The training scheme is the same as previous experiments, where the filtered LDC dictionary is used. After word alignment, standard Moses phrase extraction tool (Och and Ney, 2004) is used to build the translation models and finally Moses (Koehn et. al., 2007) is used to tune and decode.

We tune the system on the NIST MT06 test set (1664 sentences), and test on the MT08 (1357 sentences) and the DEV07[5] (1211 sentences) test sets, which are further divided into two sources (newswire and web data). A trigram language

[5] It is a test set used by GALE Rosseta Team

|  | MT02 | MT03 | MT04 | MT05 | MT08-NW | MT08-WB | Dev07NW | Dev07WB |
|---|---|---|---|---|---|---|---|---|
| Baseline | 28.87 | 27.82 | 30.08 | 26.77 | 25.09 | 17.72 | 24.88 | 21.76 |
| Dict-Link | **29.59** | *27.67* | **31.01** | 27.13 | 25.14 | 17.96 | **25.51** | 21.88 |

Table 4: Comparison of the performance of baseline and the alignment generated by new aligner with dictionary links in BLEU scores

|  |  | Precision | Recall | AER |
|---|---|---|---|---|
| Ch-En | ORL | 100.00 | 62.61 | 23.00 |
|  | F/NE | 89.25 | 62.47 | 26.50 |
|  | F/WE | 99.59 | 62.47 | 23.22 |
| En-Ch | ORL | 100.00 | 80.98 | 10.51 |
|  | F/NE | 93.49 | 80.79 | 13.32 |
|  | F/WE | 99.82 | 80.79 | 10.70 |
| Comb | F/NE | 90.79 | 87.49 | 10.89 |
|  | F/WE | 99.78 | 87.23 | 6.92 |
| Ar-En | ORL | 100.00 | 72.07 | 16.23 |
|  | F/NE | 82.46 | 72.00 | 23.13 |
|  | F/WE | 94.25 | 72.00 | 18.36 |
| En-Ar | ORL | 100.00 | 90.14 | 5.18 |
|  | F/NE | 79.81 | 90.06 | 15.37 |
|  | F/WE | 93.27 | 90.10 | 8.34 |
| Comb | F/NE | 78.91 | 93.07 | 14.59 |
|  | F/WE | 94.64 | 93.21 | 6.08 |

Table 3: Alignment quality of oracle alignment and force alignment, the rows with "ORL" in the second column are oracle alignments, "F/NE" and "F/WE" represent force alignments with empty links and without empty links correspondingly. For "F/NE" and "F/WE" we also listed the scores of heuristically symmetrized alignment[4]. ("Comb")

| Chinese-English |  |  |  |
|---|---|---|---|
|  | Precision | Recall | AER |
| Baseline | 68.22 | 46.88 | 44.43 |
| Dict-Link | 69.93 | 48.28 | 42.88 |
| English-Chinese |  |  |  |
|  | Precision | Recall | AER |
| Baseline | 65.35 | 55.05 | 40.24 |
| Dict-Link | 66.70 | 56.45 | 38.85 |
| grow-diag-final-and |  |  |  |
|  | Precision | Recall | AER |
| Baseline | 69.15 | 57.47 | 37.23 |
| Dict-Link | 70.11 | 59.54 | 35.61 |

Table 5: Comparison on alignment error rate by using alignment links generated by dictionaries

model trained from GigaWord V1 and V2 corpora is used. Table 4 shows the comparison of the performances on BLEU metric (Papineni et al., 2002). As we can observe from the results, the proposed method outperforms the baseline on all test sets except MT03, and has significant[6] improvement on MT02 (+0.72), MT04 (+0.93), and Dev07NW(+0.63). The average improvement across all test sets is 0.35 BLEU points.

As a summary, the purpose of the this experiment is to demonstrate an important characteristic of the proposed method. Even with imperfect manual alignment links, we can get better alignment by applying our method. This characteristic opens a possibility to integrate other more sophisticated aligners.

## 5 Conclusion

In this study, our major contribution is a novel generative model extended from IBM Model 4 to utilize partial manual alignments. The proposed method enables us to efficiently enforce subtle alignment constraints into the EM training. We performed experiments on manually aligned corpora to prove the validity. We also demonstrated using the method with simple heuristics to boost the translation quality on moderate size unlabelled corpus. The results show that our method is effective in promoting the word alignment qualities with small amounts of partial alignments and with high-precision-low-recall heuristics. Also the method of using dictionary to generate manual alignment links showed an average improvement of 0.35 BLEU points across 8 test sets.

The algorithm has small impact on the speed of GIZA++, and can easily be added to current multi-thread implementation of GIZA++. Therefore it is suitable for large scale training.

Future work includes applying the proposed approach on low resource language pairs and integrating the algorithm with other rule-based or discriminative aligners that can generate high-precision-low-recall partial alignments.

## Acknowledgement

This work is supported by DARPA GALE project and NSF CluE project.

[6] We used the confidence measurement described in (Zhang and Vogel, 2004)

## References

V. Ambati, S. Vogel, and J. Carbonell. 2010. Active semi-supervised learning for improving word alignment. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*.

S. Arora, E. Nyberg, and C. P. Rosé. 2009. Estimating annotation cost for active learning in a multi-annotator environment. In *HLT '09: Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 18–26.

P. Blunsom and T. Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 65–72.

P. F. Brown et. al. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics*, volume 19(2), pages 263–331.

C. Callison-Burch, D. Talbot, and M. Osborne. 2004. Statistical machine translation with word-and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 175–183.

C. Callison-Burch. 2009. Fast, cheap, and creative: Evaluating translation quality using amazons mechanical turk. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 286–295.

A. Fraser and D. Marcu. 2006. Semi-supervised training for statistical word alignment. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 769–776.

A. Fraser and D. Marcu. 2007. Getting the structure right for word alignment: LEAF. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 51–60.

Q. Gao and S. Vogel. 2008. Parallel implementations of word alignment tool. In *Proceedings of the ACL 2008 Software Engineering, Testing, and Quality Assurance Workshop*, pages 49–57.

Q. Gao and S. Vogel. 2010. Consensus versus expertise : A case study of word alignment with mechanical turk. In *NAACL 2010 Workshop on Creating Speech and Language Data With Mechanical Turk*, pages 30–34.

F. Guzman, Q. Gao, and S. Vogel. 2009. Reassessment of the role of phrase extraction in pbsmt. In *The twelfth Machine Translation Summit*.

F. Huang. 2009. Confidence measure for word alignment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 932–940. Association for Computational Linguistics.

A. Ittycheriah and S. Roukos. 2005. A maximum entropy word aligner for arabic-english machine translation. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 89–96.

P. Koehn et. al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Y. Liu, Q. Liu, and S. Lin. 2005. Log-linear models for word alignment. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 459–466.

R. C Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 81–88.

J. Niehues and S. Vogel. 2008. Discriminative word alignment via alignment matrix modeling. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25.

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

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. In *Computational Linguistics*, volume 30, pages 417–449.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of ACL'02*, pages 311–318, Philadelphia, PA, July.

B. Taskar, S. Lacoste-Julien, and Klein D. 2005. A discriminative matching approach to word alignment. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 73–80.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM based word alignment in statistical machine translation. In *Proceedings of 16th International Conference on Computational Linguistics)*, pages 836–841.

Y. Zhang and S. Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of The 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, October.