RC23491 (W0501-050) January 14, 2005
Computer Science

IBM Research Report

Inner-Outer Bracket Models for Word Alignment Using Hidden Blocks

Bing Zhao
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Niyu Ge, Kishore Papineni
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Inner-Outer Bracket Models for Word Alignment using Hidden Blocks

Bing Zhao
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, 15213
bzhao@cs.cmu.edu

Niyu Ge, Kishore Papineni
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
{niyuge,papineni}@us.ibm.com

Abstract

Most of the best translation systems today are based on phrase translation pairs (i.e. "blocks"). The blocks are obtained from word alignment. In this paper, we use blocks to improve word alignment. Improved word alignment in turn leads to better inference of blocks. We propose two new probabilistic models and EM-based algorithms to estimate their parameters. The first model recovers IBM Model-1 as a special case. Both models outperform bidirectional alignments based on HMM or IBM Model-4: up to 10% absolute improvement in Chinese-English word alignment over GIZA Model-4 bidirectional alignment. Using blocks obtained from the proposed models, we also get statistically significant improvement in BLEU on NIST MT-03 test set.

1 Introduction

Today's state-of-the-art statistical machine translation systems use phrase translation pairs ("blocks"). Here, phrase is simply a contiguous sequence of words. Blocks are obtained automatically from pairs of sentences that are translations of each other. This block extraction is based on underlying alignment of words between the parallel sentences. Therefore, word alignment is fundamental to statistical machine translation. It is also challenging in that automatic word alignment accuracy is not yet close to inter-annotator agreement – at least in some language pairs: for Chinese-English, inter-annotator agreement is in the 90's (F-measure) whereas Model-4 or HMM accuracy is in the 70's.

Word alignment traditionally is based on IBM Model-1 to Model-5 (Brown, et.al. 1993) or Hidden Markov Models (Vogel, et.al. 1996). HMM-based alignment assumes that words "close-in-source" are aligned to words "close-in-target". While this locality assumption is generally sound, HMMs do have limitations: the self-transition probability of a state (word) controls only the duration in the state: the length of the phrase aligned to the word. But there is no natural way to control repeated non-contiguous visits to a state. For instance, when there is a single comma in the English sentence and there are three commas scattered in the Chinese sentence ("observation"), HMM models incorrectly align all the Chinese commas to the English comma. However, HMMs are attractive for their speed and reasonable accuracy.

In this paper, we study another way of localizing alignments. We use blocks to achieve locality in the following manner: a block in a sentence pair is a source phrase aligned to a target phrase. But we assume that words in the source phrase cannot align to words outside the target phrase and that words outside the source phrase cannot align to words in the target phrase.

Furthermore, a block divides the sentence pair into two smaller regions: the **inner** part of the block, which corresponds to the source and target phrase in the block, and the **outer** part of the block, which corresponds to the remaining source and target words in the parallel sentence excluding the block. The two regions are non-overlapping, and each of them is shorter than the original parallel sentence pair. It is easier to align shorter sentence pairs. We can carry out the alignment for each of these smaller sentence pairs (e.g. using IBM Model-1), collect the fractional counts from them, and weigh the counts by the posterior probability of the block.

In the above, we used a single block to split the sentence pair into two regions. But it is not clear which block we should pick for this purpose. We treat the splitting block as a hidden variable. This approach is far simpler than treating the entire sentence as a sequence of phrases and considering such segmentation as a hidden variable. It is also simpler than segmenting the sentence pair into non-overlapping block sequence as in (Marcu and Wong, 2002).

The paper is organized as follows: Section 2 formally introduces the segmentation induced by a block. Section 3 describes our new models, Inner-Outer Bracket Model-A and the Inner-Outer Bracket Model-B, together with a new Null-word model. Section 4 describes a maximum posterior search for word alignment. Section 5 presents our experimental results and Section 6 the conclusions.

2 Segmentation by a Block

We use the following notation throughout the paper: E denotes the English sentence and F denotes the foreign-language sentence. The English sentence is indexed with i, with sentence length of I, and the foreign sentence is indexed with j, with sentence length J; e is a word in E and f a word in F; A is the alignment vector with a_j indicating the position of the English word to which f_j connects. We therefore have the standard limitation that one foreign word cannot be connected to more than one English word.

A bracket is a pair of indices specifying a span of contiguous positions in the sentence.

$$\begin{aligned} B_E &= Bracket(E) = [i_{left}, i_{right}] \\ B_F &= Bracket(F) = [j_{left}, j_{right}] \end{aligned} \tag{1}$$

We define a block as a pair of brackets: one bracket in English sentence together with its projection in the foreign sentence, which is also a bracket.

$$Block = \{Bracket(E),\ \ Bracket(F)\} \tag{2}$$

B_E segments E into two parts: the inner part $[i_{left}, i_{right}]$, and the outer part $[0, i_{left}) \cup (i_{right}, I]$. B_F segments F similarly. The block splits the parallel sentence pair into two non-overlapping regions: Inner and Outer parts. Figure 1 shows such a segmentation of the parallel sentence by one block, and the resulting inner and outer parts.

We do not allow alignment to cross the region boundaries. This is our proposed new localization method.

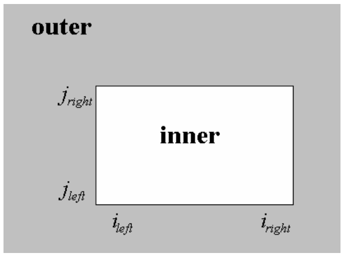

Figure 1. A block induces segmentation.

3 Inner-Outer Bracket Models

As introduced in the previous section, each block is a local constraint for the alignment decisions. If the block is reasonably good, we can expect better word alignment. We treat the constraining block as a hidden variable in a generative model as shown in Equation 3.

$$\begin{aligned} P(F \mid E) &= \sum_{B_E} P(F \mid B_E, E) P(B_E \mid E) \\ &= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F, B_F \mid B_E, E) \cdot P(B_E \mid E) \end{aligned} \tag{3}$$

where B_E is one bracket in English sentence; $proj(B_E)$ is a set of projections of B_E in the foreign sentence, and B_F is one such projection. $P(B_E \mid E)$ is a monolingual bracketing model, and $P(F, B_F \mid B_E, E)$ is a generative model which implements the inner-outer constraint in the generation process. Two different interpretations of $P(F, B_F \mid B_E, E)$ result in two sub models described below.

3.1 Inner-Outer Bracket Model-A

The first simplified model $P(F, B_F \mid B_E, E)$ assumes that the inner part and the outer part are generated independently:

$$\begin{aligned} P(F \mid E) &= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F, B_F \mid B_E, E) \cdot P(B_E \mid E) \\ &= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F_{i(B_F)}, F_{o(B_F)} \mid B_E, E) \cdot P(B_E \mid E) \\ &= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F_{in(B_F)} \mid E_{in(B_E)}) P(F_{out(B_F)} \mid E_{out(B_E)}) \\ &\quad \cdot P(B_E \mid E) \end{aligned} \tag{4}$$

where $P(F_{in(B_F)} \mid E_{in(B_E)}) = \sum_{\overset{\varpi}{a}, a_j \in in(B_E)} \prod_{j \in in(B_F)} P(f_j \mid e_{a_j}) P(e_{a_j} \mid E_{in(B_E)})$ and $\overset{\varpi}{a} = \{a_1, a_2, \Lambda\ , a_J\}$ is the word alignment vector, which is the other hidden variable in our model.

$P(e_{a_j} \mid E_{in(B_E)})$ is a bracket level English ngram language model.

For simplicity, we can assume IBM-Model-1 alignments for both inner and outer parts as in Equation 5:

$$P(F_{in(B_F)} \mid E_{in(B_E)}) = \prod_{j \in in(B_F)} \sum_{i \in in(B_E)} P(f_j \mid e_i) P(e_i \mid E_{in(B_E)})$$
$$P(F_{out(B_F)} \mid E_{out(B_E)}) = \prod_{j \in out(B_F)} \sum_{i \in out(B_E)} P(f_j \mid e_i) P(e_i \mid E_{out(B_E)}) \quad (5)$$

Equation 4 shows a bracket B_F segments F into two non-overlapping regions: *inside(B_F)* and *outside(B_F)*, and they are generated independently.

The next independence assumption comes from the English side, where we assume the English words inside the block $E_{inside(B_E)}$ can only generate the words in $F_{inside(B_F)}$, and nothing else; likewise $E_{outside(B_E)}$ only generates $F_{outside(B_F)}$. In Equation 5, for a particular B_F, if B_E is too small, $P(F_{inside(B_F)} \mid E_{inside(B_E)})$ will suffer, and if B_E is too big $P(F_{outside(B_F)} \mid E_{outside(B_E)})$ will suffer. Overall, our proposed model in Equation 4 and 5 combines both costs, and requires both inner and outer parts to be explained well at the same time.

We can simply apply IBM Model-1 shown as in Equation 5 to model both $P(F_{inside(B_F)} \mid E_{inside(B_E)})$ and $P(F_{outside(B_F)} \mid E_{outside(B_E)})$, and the key E-step computations are shown as follows:

$$P(B_E \mid E, F) \approx P(B_E \mid |E|) = \frac{1}{num\_brackets}$$

$$P(\overset{\varpi}{a} \mid B_F, F, B_E, E) = \prod_{j=1}^{J} P(a_j \mid B_F, F, B_E, E)$$
$$= \prod_{j=1}^{J} \{P(a_j \mid F_{in(B_F)}, E_{in(B_E)}) + P(a_j \mid F_{out(B_F)}, E_{out(B_E)})\}$$

$$P(a_j = i \mid F_{out(B_F)}, E_{out(B_E)}) = \frac{P(f_j \mid e_i)}{\sum_{k \in out(B_E)} p(f_j \mid e_k)}$$

$$P(B_F \mid E, B_E) = \frac{\prod_{j \in B_F} P(f_j \mid e_{a_j}) P(e_{a_j} \mid E, B_E)}{\sum_{all\ B'_F} \prod_{j \in B'_F} P(f_j \mid e_{a_j}) P(e_{a_j} \mid E, B_E)}$$

In M-step, we collect all the fractional counts and normalize them to update the parameters as shown in the following equation:

$$\hat{P}(f \mid e) = {}^{1}\!/_{\lambda} \cdot \sum_{all(E,F)} \sum_{\overset{\varpi}{a}} P(\overset{\varpi}{a} \mid B_F, B_E, E, F) \sum_{i,j}^{I,J} \delta(f_j, f) \delta(e_i, e)$$
$$\cdot P(B_F \mid B_E, E) \cdot P(B_E \mid E, F)$$

where λ is the normalization factor.

In principle, B_E can be a bracket of any length not exceeding the sentence length. If we restrict the bracket length to that of the sentence length, we recover IBM Model-1. In practice, the length of B_E is limited to 4 or 5 words.

Figure-2 illustrates the generation process for Inner-Outer Bracket Model-A.

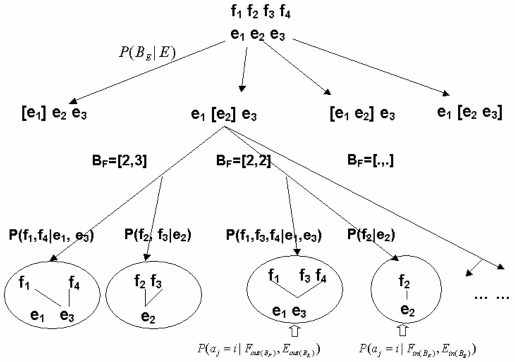

Figure 2. Illustration of generative Model-A

3.2 Inner-Outer Bracket Model-B

Starting from Equation 2, we can go one step further to re-write $P(F, B_F \mid B_E, E)$ by first explicitly predicting the projections for the given English bracket as shown in Equation 6.

$$P(F \mid E) = \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F, B_F \mid B_E, E) \cdot P(B_E \mid E)$$
$$= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F \mid B_F, B_E, E) P(B_F \mid B_E, E) \cdot P(B_E \mid E)$$
$$= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F \mid B_F, B_E, E) P((j_{left}, j_{right})_{B_F} \mid B_E, E) \cdot P(B_E \mid E)$$
$$= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F \mid B_F, B_E, E) P((center, width)_{B_F} \mid B_E, E) \cdot P(B_E \mid E)$$
$$= \sum_{B_E} \sum_{B_F \in proj(B_E)} P(F \mid B_F, B_E, E) P(center_{B_F} \mid B_E, E) P(width_{B_F} \mid B_E, E)$$
$$\cdot P(B_E \mid E) \quad (6)$$

Here, the model first generates the brackets for English sentence, and then generates the bracket's projections. The projection is a bracket in the foreign sentence expressed by the left and right boundaries (j_{left}, j_{right}) in the foreign sentence. This can be equivalently defined as the *center* and *width* of the bracket: $(center, width)_{B_F}$. We assume that center and width can be predicted independently. The width usually depends on the length of the English bracket, while the center is usually dependent on the translational equivalence of the English bracket and its projection .

$P(F \mid B_F, B_E, E)$ is the alignment model, and can be approximated as in Equation 7:

$$P(F \mid B_F, B_E, E) = P(F_{inside(B_F)} \mid E_{inside(B_E)})$$
$$= \sum_{\overset{\varpi}{a}} \prod_{j \in B_F} P_t(f_j \mid e_{a_j}) P(a_j \mid E_{inside(B_E)}) \quad (7)$$

Inner-Outer Bracket Model-B avoids the burden of predicting links in the outside part of the block. In this way, it saves some computation, and practically the model is more focused on the predictions of the inner part of the block.

The EM training is straightforward. The E-step is very similar to Bracket Model-A, as shown in the following equations:

$$P(\overset{\varpi}{a} \mid B_F, F, B_E, E) = \prod_{j=1}^{J} P(a_j \mid B_F, F, B_E, E)$$

$$P(a_j = i \mid F, B_F, E, B_E) = \frac{P(f_j \mid e_i)}{\sum_{k \in B_E} p(f_j \mid e_k)} \cdot 1_{\{j \in B_F,\ k, i \in B_E\}}$$

Modeling $P(center_{B_F} \mid F, E, B_E)$ can be very difficult and complicated. Theoretically, any reasonable score function can be used. To simplify the computation, we first compute the expectations of the center and width, and then apply a local greedy search over all the neighboring candidates close to the computed expectations. If only the top choice from the greedy search for the center and width is chosen, then the posterior is also simplified as $P(B_F \mid F, E, B_E) = 1.0$.

The expectation of *width*$_\text{F}$ depends on B_E's width and the fertilities of English words in B_E. In our case, the expected width is computed as in Equation 8.

$$E(width \mid F, E, B_E) = E(width \mid |B_E|) = \gamma \cdot |B_E| \quad (8)$$

where γ is the phrase length ratio, which is approximated as the sentence length ratio computed from the whole parallel corpus. For Chinese-English in our case, γ is set as 1/1.3 (0.77).

The expectation of center$_\text{F}$ is computed as in Equation 9:

$$E(center \mid F, E, B_E) = \sum_{j=0}^{J} j \cdot P(j \mid F, E, B_E)$$
$$\approx \sum_{j=0}^{J} \sum_{i \in B_E} j \cdot P(f_j \mid e_i) P(e_i \mid B_E, E) \quad (9)$$

where *j* is the position in the foreign sentence, and the expectation is a weighted average of the expected centers from all the individual English words in the bracket of B_E.

Given the expected center and width, which is a good starting point to compute the left and right boundaries, we do a local greedy search for the final best projection for the given English bracket B_E as shown in the following table.

```
For left=center-width; left <=center+1; left++,
For right=center-1;right<=center+width;right++
    Score=
        $P(F_{inside(B_F)} \mid E_{inside(B_E)}) P(F_{outside(B_F)} \mid E_{outside(B_E)})$;
    Record the best score and its boundaries;
  end
end
Retrieve the best score and the
  corresponding left and right boundaries.
```

Table 1. Local Greedy Search for Bracket Boundaries

In this strategy, one can choose Top-1, which corresponding to the Viterbi search, and the posterior is $P(B_F \mid F, E, B_E) = 1.0$; One can also choose Top-N candidates of the projections from this local greedy search, and normalize over these choices, one can also get estimations of the posterior of $P(B_F \mid F, E, B_E)$.

The M-step is similar to Bracket Model-A, but with different interpretations for sub-models. It is essentially a normalization of the fractional counts collected from the E-step.

Figure-3 illustrates the generation process for Inner-Outer Bracket Model-B.

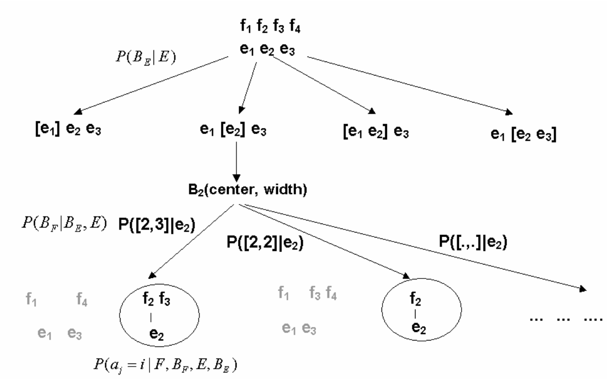

Figure 3. Illustration of generative Model-B

3.3 A NULL Word Model

The null word model enables words to be aligned to nothing. In the traditional IBM models, there is a universal null word, which is attached to every sentence pair to compete with the word generators in the training. This universal null word is too general to be effective for null links.

In our inner-outer bracket models, we propose a context specific null word model, which uses the left and right context as competitors in the generative process for the current word. We show the equations for P(E|F) with the left context for simplicity, and P(F|E) is similar.

$$P(E \mid F) = \prod_{i=1}^{I} P(e_i \mid e_0^{i-1}, F) = \prod_{i=1}^{I} P(e_i \mid e_{i-1}, F)$$
$$= \prod_{i=1}^{I} \{P(e_i \mid e_{i-1}, F) P(e_{i-1} \mid F) + \sum_{j=1}^{J} P(e_i \mid f_j) P(f_j \mid F)\} \qquad (10)$$

The above equation is expanded by the left context e_{i-1}, which competes with foreign words to generate the current word e_i. The null word model is now $P(e_i \mid e_{i-1}, F)$. With it, we can use heuristics to infer which bracket the current word e_i is in, and if its alignment is too far away from the center of that bracket, then the alignment link for e_i will be dropped (i.e. a null link).

4 A Max-Posterior for Word Alignment

A maximum-posterior method is shown to be effective in (Ge, 2004). We also applied it in both our baselines and proposed models to infer word alignment. This method can be applied to any matrix besides posteriors such as HMM or our proposed Inner-Outer Bracket Models.

Let the words in the foreign sentence be the set of states S and the words in the English sentence be the set of observations O. The posterior probabilities P(s|o) (i.e. state given observation) are obtained after the forward-backward training. The maximum-posterior word alignments are obtained by computing

$$S_t^* = \arg\max_s P(S_t = s \mid \vec{O})$$

where *t* is chosen globally over the entire posterior matrix. In Viterbi alignments one computes

$$\vec{S}^* = \arg\max P(S_1, ... S_T \mid \vec{O})$$

that is, the best state *sequence* given the observation. In contrast, the maximum posterior computes the best state one at a time.

Once we find the maximum in the matrix, we also know the corresponding state and observation (S_t, O_t), which is nothing but the word pair (e_i, f_j). We'll then align the pair and continue to find the next posterior maximum. The process is repeated until either every word in one (or both) language is aligned or no more maximum can be found, whichever happens first.

We also observe in parallel corpora that when one word translates into multiple words in another language, it usually translates into a phrase, i.e. a contiguous sequence of words. We therefore impose this contiguity constraint on word alignments. When one foreign word aligns to multiple English words, the English words must be contiguous in the sentence. The procedure to find word alignments is as follows. Given a parallel sentence pair [E, F] with lengths *I* and *J*, let *A* be an alignment matrix with *J* rows and *I* columns.

```
clear $A$
while  (f = $S_t^*$) {
    if  (f is not aligned) align(f, $e_t$)
    else if  ($e_t$ is contiguous to what f is aligned to)
        align(f, $e_t$)
    }
}
return $A$
```

Table 2. A Max-Posterior for word alignment

5 Experiments

To evaluate our proposed models' performance, we run both word alignment experiments and machine translation experiments using these alignments. Our experiments are carried out on Chinese-English language pair.

For word alignment, there are 260 hand-aligned sentence pairs labeled by eight bilingual speakers. There are total 4676 word pair links in this gold standard set. We have one-to-one, one-to-many, and many-to-many alignment links. If the link has one target spontaneous word (in our case, the target is English), it is considered as a spontaneous link. We report the overall F-measures as well as F-measures for both content and spontaneous word links, as also suggested in (Ahrenberg, et.al. 2000). Our significant test shows a confidence interval of +/-1.56% F-measure at the 95% level, and our bootstrap significant test using 200 batches, 13 sentences per batch (sampled with replacement from the data) gives +/- 0.61% as the interval.

We prepared two sets of data. The small training set has 5K sentence pairs, which is a selection of XinHua news stories sentence-aligned by a human annotator. It has 131K English words and 125K

Chinese words[1]. The large training set has 181K sentence pairs (5K + 176K), and the additional 176K sentence pairs are from FBIS and Sinorama provided by LDC, which has 6.7 million English words and 5.8 million Chinese words. It turns out that the 5K subset does not matter in the large setting.

5.1 Baseline Systems

Our baseline is our implementation of HMM with the maximum-posterior method for inferring word alignment. IBM Model-4 is trained using GIZA++, of which we follow the best reported settings in (Och , Ney 2003), and tuned a few parameters including maximum fertility, and smoothing factors for up to IBM Model-4. We collect two directions of the alignments, get the intersections and fill in the gaps with heuristics of looking at neighbors like the algorithm in (Kohen 2004). Table 3 summarizes our baselines.

Data	Settings	Spont	Content	Both
Small (5K)	HMM EC-P	54.69	69.99	**64.78**
	HMM EC-V	31.38	5356	55.59
	HMM CE-P	51.44	69.35	62.69
	HMM CE-V	31.43	63.84	55.45
Large (181K)	HMM EC-P	60.08	78.01	**71.92**
	HMM EC-V	32.80	74.10	64.26
	HMM CE-P	58.45	79.44	71.84
	HMM CE-V	35.41	79.12	68.33
Small (5K)	GIZA MH-bi	45.63	69.48	60.08
	GIZA M4-bi	48.80	73.68	**63.75**
Large (181K)	GIZA MH-bi	49.13	76.51	65.67
	GIZA M4-bi	52.88	81.76	**70.24**
	Fully-aligned[2]	5.10	15.84	9.28

Table 3. Baseline Systems (V: Viterbi, P: Max-Posterior)

The best baseline (HMM using maximum-posterior: HMM EC-P) we have for small training data is F-measure 64.78%, and for large training data is 71.92%. The best result from using GIZA++ is using Model-4 bi-direction alignments with F-measure of 63.75% for small data, and F-measure 70.24% for large training data. Our HMM with max-posterior gives the strongest baseline, and is statistically significant better than HMM Viterbi alignment.

5.2 Inner-Outer Bracket Models

To train our models, we run 5 iterations of HMM, and then load the trained *P(f|e)* to initialize our proposed inner-outer Bracket models. 15~20 EM iterations are carried out to estimate our model parameters. The very initial iteration starts from the fully aligned[2] sentence pairs, which has a F-measure of 9.28%.

Figure 4 shows the performance of Inner-Outer Bracket Model-A (BM-A) over EM iterations. *Top-1* means we only collect fractional counts from Top-1 projection for each given English bracket. *Top-all* means we collect fractional counts from all the possible projections for the given bracket. *Inside* means the fractional counts are collected from the inner part of the block only, and *outside* means collecting the counts from outer parts only. From Figure 4, using Top-1 projection from the inner parts of the block gives the best performance. The peak performance for BM-A is achieved at iteration 5 at an F-measure of 72.29%, about 7.5% improvement over the strongest baseline, and is statistically significant.

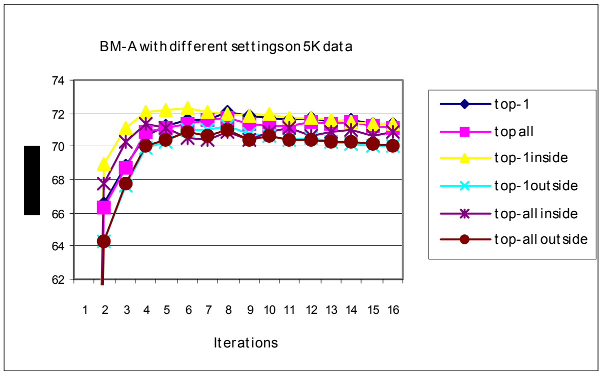

Figure 4. BM-A with different settings on small data

Figure 5 shows the performance of Inner-Outer Bracket Model-B (BM-B) over EM iterations. Similar to BM-A, *Top-1* means we only use top-1 projection for each given English bracket, and *Top-all* means we use all the projections for the given English bracket. Smoothing means in collecting the fractional counts, we weigh the original fractional count by 0.95, and give the remaining 0.05 weight to each confident link, which was also aligned in the previous iteration. "w/null" means we applied the proposed NULL word model to infer null links. We also predefined a list of 15 English function words, for which there might be no corresponding word in Chinese. These 15 English words are "a, an, the, of, to, for, by, up, be, been, being, does, do, did, -". In the "drop-null"

[1] Chinese is word segmented using a segmenter built internally.

[2] Every word pair is aligned.

experiments, the links containing these predefined function words are dropped in the final word alignment (i.e. they are left unaligned)

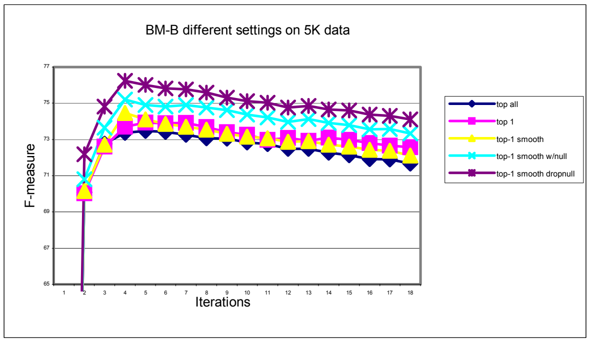

Figure 5. BM-B with different settings on small data

The peak performance is achieved around iteration 4~5. At iteration 5, setting "top-1" gives F-measure of 73.93%, which is significantly better than BM-A's best setting in Figure 4. With smoothing, it reaches to 74.46%. After applying null word model, we achieved an F-measure of 75.20% at iteration 4. If we simply drop links containing the 15 English words, we can achieve F-measure of 76.24%. Figure 6 shows the performance when using the large training data.

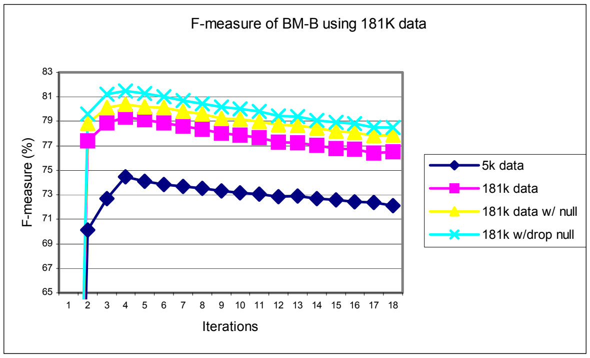

Figure 6. BM-B with different settings on large data

In Figure 6, without dropping the 15 English words, the best performance we achieved is F-measure of 80.38% at iteration 4 using the Top-1 projection per bracket together with the null word model. With dropping 15 English words, we have F-measure of 81.47% using the Top-1 projection per bracket. In this large data setting, the 5K (human sentence-aligned) data gives only 0.5% F-measure difference, which is not statistically significant.

In Figure 7, we show the F-measure performances at different maximum bracket lengths using BM-B with the top-1 projection only on the large training data. When the maximum bracket length equals to 1, the model tries to map unigram to brackets generally longer than unigram. This projection is often incorrect and the performance is close to that of IBM Model-1. When using longer bracket length such as 9-gram, the computation is more expensive, but the performance stays almost the same. So practically, we choose maximum bracket length of 4 or 5.

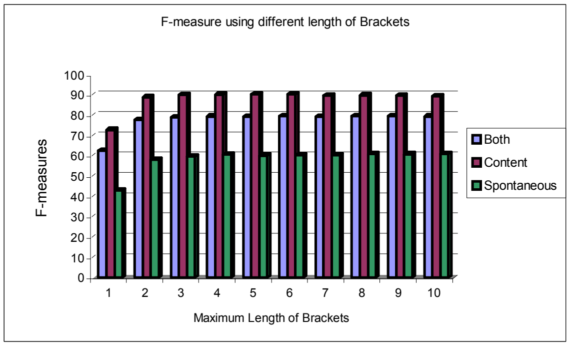

Figure 7. BM-B using different max ngram length

5.3 Evaluate Blocks in the EM Iterations

Out intuition was that good blocks can improve word alignment, and in turn, good word alignment can lead to better blocks. Our experimental results support the first claim. Now we consider the second claim of whether good word alignment leads to better blocks.

Given reference human word alignment, we extract reference blocks up to 5-gram phrases on the Chinese side. This block extraction procedure is the same as the one used to extract blocks for the translation experiments in section 5.4

During the EM iterations, we output all the blocks actually used in the iteration, then evaluate the precision and recall according to the extracted reference blocks. The results are in Figure 8.

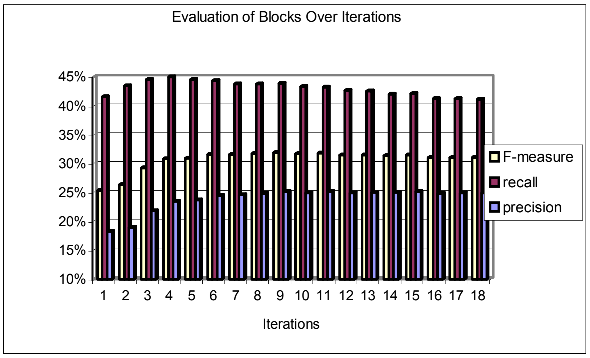

Figure 8. A direct evaluation of Blocks in BM-B

Because we extract all possible N-grams at each position in the English sentence, the precision is low, and the recall is relatively high. The figure shows that blocks do improve, presumably benefiting from better word alignments.

In Table 4, we summarize our Inner-Outer Bracket Model-B at different settings for detailed comparisons.

Data	Settings	Spontaneous	Content	Both
Small (5K)	Baseline	54.69%	69.99%	64.78%
	BM-B-drop	62.76%	82.99%	76.24%
	BM-B w/null	61.24%	82.54%	75.19%
	BM-B smooth	59.61%	82.99%	74.46%
Large (181K)	Baseline	60.08%	78.01%	71.92%
	BM-B-drop	63.95%	90.09%	81.47%
	BM-B w/null	62.24%	89.99%	80.38%
	BM-B smooth	60.49%	90.09%	79.31%

Table 4. Performances of BM-B with different settings

Overall, without dropping the 15 English words, BM-B gives about 8% F-measure improvement in large training data settings and 9% for small training data settings whereas the confidence interval is only +/- 1.5%.

5.4 Evaluation of Translations

We also carried out the translation experiments using our Inner-Outer Bracket Model-B best settings on the TIDES Chinese-English 2003 test set.

We trained our models on 354,252 sentence pairs, drawn from LDC-supplied parallel corpora. With this test-specific training data, we run 5 iterations of EM training of BM-B to infer word alignments. We use a monotone decoder for final translations. Our baseline is using phrase pairs built from the HMM maximum posterior word alignment and using the HMM trained P(f|e). This baseline blue score is 0.2237 +/- 0.0113 (cased) and 0.2453 +/- 0.0117 (uncased). Table-5 summarizes the bleu scores using blocks inferred from improved word alignments over each of the EM iteration.

	EM-1	EM-2	EM-3	EM-4	EM-5
Bleur4n4	0.2515	0.2549	0.2521	0.2530	0.2501
Bleur4n4C	0.2276	0.2303	0.2280	0.2287	0.2257

Table 5. Bleu scores from the word alignment of each EM iterations in Bracket Model-B

If we use the same blocks as used in the baseline, but use BM-B trained lexicon P(f|e) instead, we have bleu score 0.2308 (cased) and 0.2521 (uncased) which is comparable to the numbers in Table 5. If we use both word alignment and the model parameters of P(f|e) trained from BM-B, we improve the translation quality further as shown in Table 6:

	EM-1	EM-2	EM-3	EM-4	EM-5
Bleur4n4	0.2641	0.2703	0.2698	0.2725	0.2696
Bleur4n4C	0.2413	0.2453	0.2450	0.2480	0.2451

Table 6. Bleu scores from the word alignment and p(f|e) trained in Bracket Model-B

Using our proposed model, we got overall improvement in translation from 0.2237 (uncased: 0.2453) to 0.2480 (uncased: 0.2725).

6 Conclusion

Our main contributions are two new Inner-Outer Bracket models, which utilize blocks as features to infer better word alignments. We also proposed one context dependent NULL word model to infer the null links. We show improved word alignments using both the small training data and the large training data. We also show significant improvements in translation quality using the proposed models.

References

Lars Ahrenberg, Magnus Merkel, Anna Sagball Hein, and Jorg Tiedemann. "Evaluation of word alignment system". LREC 2000.

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

Niyu Ge, "A maximum posterior method for word alignment". Presentation given at DARPA/TIDES MT workshop, 2004.

Philipp Koehn. "The Foundation for Statistical Machine Translation at MIT". Talk given at DARPA/TIDES MT workshop, 2004.

Daniel Marcu and William Wong. A phrase-based, joint probability model for statistical machine translation. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA, July 6-7, 2002.

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

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *COLING'96: The 16thInt. Conf. on Computational Linguistics*, pages 836-841, Copenhagen, Denmark, August, 1996.