Discriminative word alignment by learning the alignment structure and syntactic divergence between a language pair

Sriram Venkatapathy[1]
Language Technologies Research Centre, IIIT -Hyderabad
Hyderabad - 500019, India.
sriram@research.iiit.ac.in

Aravind K. Joshi
Department of Computer and Information Science and Institute for Research in Cognitive Science,
University of Pennsylvania, PA, USA.
joshi@linc.cis.upenn.edu

Abstract

Discriminative approaches for word alignment have gained popularity in recent years because of the flexibility that they offer for using a large variety of features and combining information from various sources. But, the models proposed in the past have not been able to make much use of features that capture the likelihood of an alignment structure (the set of alignment links) and the syntactic divergence between sentences in the parallel text. This is primarily because of the limitation of their search techniques. In this paper, we propose a generic discriminative re-ranking approach for word alignment which allows us to make use of structural features effectively. These features are particularly useful for language pairs with high structural divergence (like English-Hindi, English-Japanese). We have shown that by using the structural features, we have obtained a decrease of 2.3% in the absolute value of alignment error rate (AER). When we add the cooccurence probabilities obtained from IBM model-4 to our features, we achieved the best AER (50.50) for the English-Hindi parallel corpus.

1 Introduction

In this paper, we propose a discriminative re-ranking approach for word alignment which allows us to make use of structural features effectively. The alignment algorithm first generates a list of k-best alignments using local features. Then it re-ranks this list of k-best alignments using global features which consider the entire alignment structure (set of alignment links) and the syntactic divergence that exists between the sentence pair. Use of structural information associated with the alignment can be particularly helpful for language pairs for which a large amount of unsupervised data is not available to measure accurately the word cooccurence values but which do have a small set of supervised data to learn the structure and divergence across the language pair. We have tested our model on the English-Hindi language pair. Here is an example of an alignment between English-Hindi which shows the complexity of the alignment task for this language pair.

Figure 1: An example of an alignment between an English and a Hindi sentence

To learn the weights associated with the parameters used in our model, we have used a learning framework called MIRA (The Margin Infused Relaxed Algorithm) (McDonald et al., 2005; Crammer and Singer, 2003). This is an online learning algorithm which looks at one sentence pair at a time and compares the k-best predictions of the alignment algorithm with the gold alignment to update the parameter weights appropriately.

In the past, popular approaches for doing word alignment have largely been generative (Och and Ney, 2003; Vogel et al., 1996). In the past couple of years, the discriminative models for doing word alignment have gained popularity because of

[1]1Part of the work was done at Institute for Research in Cognitive Science (IRCS), University of Pennsylvania, Philadelphia, PA 19104, USA, when he was visiting IRCS as a Visiting Scholar, February to December, 2006.

Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation, pages 49–56,
Rochester, New York, April 2007.

the flexibility they offer in using a large variety of features and in combining information from various sources.

(Taskar et al., 2005) cast the problem of alignment as a maximum weight bipartite matching problem, where nodes correspond to the words in the two sentences. The link between a pair of words, (e_p,h_q) is associated with a score (score(e_p,h_q)) reflecting the desirability of the existence of the link. The matching problem is solved by formulating it as a linear programming problem. The parameter estimation is done within the framework of large margin estimation by reducing the problem to a quadratic program (QP). The main limitation of this work is that the features considered are local to the alignment links joining pairs of words. The score of an alignment is the sum of scores of individual alignment links measured independently i.e., it is assumed that there is no dependence between the alignment links. (Lacoste-Julien et al., 2006) extend the above approach to include features for fertility and first-order correlation between alignment links of consecutive words in the source sentence. They solve this by formulating the problem as a quadratic assignment problem (QAP). But, even this algorithm cannot include more general features over the entire alignment. In contrast to the above two approaches, our approach does not impose any constraints on the feature space except for fertility ($\leq$1) of words in the source language. In our approach, we model the one-to-one and many-to-one links between the source sentence and target sentence. The many-to-many alignment links are inferred in the post-processing stage using simple generic rules. Another positive aspect of our approach is the application of MIRA. It, being an online approach, converges fast and still retains the generalizing capability of the large margin approach.

(Moore, 2005) has proposed an approach which does not impose any restrictions on the form of model features. But, the search technique has certain heuristic procedures dependent on the types of features used. For example, there is little variation in the alignment search between the LLR (Log-likelihood ratio) based model and the CLP (Conditional-Link Probability) based model. LLR and CLP are the word association statistics used in Moore's work (Moore, 2005). In contrast to the above approach, our search technique is more general. It achieves this by breaking the search into two steps, first by using local features to get the k-best alignments and then by using structural features to re-rank the list. Also, by using all the k-best alignments for updating the parameters through MIRA, it is possible to model the entire inference algorithm but in Moore's work, only the best alignment is used to update the weights of parameters. (Fraser and Marcu, 2006) have proposed an algorithm for doing word alignment which applies a discriminative step at every iteration of the traditional Expectation-Maximization algorithm used in IBM models. This model still relies on the generative story and achieves only a limited freedom in choosing the features. (Blunsom and Cohn, 2006) do word alignment by combining features using conditional random fields. Even though their approach allows one to include overlapping features while training a discriminative model, it still does not allow us to use features that capture information of the entire alignment structure.

In Section 2, we describe the alignment search in detail. Section 3 describes the features that we have considered in our paper. Section 4 talks about the Parameter optimization. In Section 5, we present the results of our experiments. Section 6 contains the conclusion and our proposed future work.

2 Alignment Search

The goal of the word alignment algorithm is to link words in the source language with words in the target language to get the alignments structure. The best alignment structure between a source sentence and a target sentence can be predicted by considering three kinds of information, (1) Properties of alignment links taken independently, (2) Properties of the entire alignment structure taken as a unit, and (3) The syntactic divergence between the source sentence and the target sentence, given the alignment structure. Using the set of alignment links, the syntactic structure of the source sentence is first projected onto the target language to observe the divergence.

Let e_p and h_q denote the source and target words respectively. Let n be the number of words in source sentence and m be the number of words in target sentence. Let S be the source sentence and T be the target sentence.

2.1 Populate the Beam

The task in this step is to obtain the k-best candidate alignment structures using the local features. The local features mainly contain the cooccurence information between a source and a target word and are independent of other alignment links in the sentence pair. Let the local feature vector be denoted as $f_L(e_p, h_q)$. The score of a particular alignment link is computed by taking a dot product of the weight vector W with the local feature vector of the alignment link. More formally, the local score of an alignment link is

$$score_L(e_p, h_q) = W.f_L(e_p, h_q)$$

The total score of an alignment structure is computed by adding the scores of individual alignment links present in the alignment. Hence, the score of an alignment structure $\bar{a}$ is,

$$score_{La}(\bar{a}, S, T) = \sum_{(e_p,h_q)\in\bar{a}} score_L(e_p, h_q)$$

We have proposed a dynamic programming algorithm of worst case complexity $O(nm^2 + nk^2)$ to compute the k-best alignments. First, the local score of each source word with every target word is computed and stored in local beams associated with the source words. The local beams corresponding to all the source words are sorted and the top-k alignment links in each beam are retained. This operation has the worst-case complexity of $O(nm^2)$.

Now, the goal is to get the k-best alignments in the global beam. The global beam initially contains no alignments. The k best alignment links of the first source word e_0 are added to the global beam. To add the alignment links of the next source word to the global beam, the k^2 (if $k < m$) combinations of the alignments in the global beam and alignments links in the local beam are taken and the best k are retained in the global beam. If $k > m$, then the total combinations taken are mk. This is repeated till the entries in all the local beams are considered, the overall worst case complexity being $O(nk^2)$ (or $O(nmk)$ if $k > m$).

2.2 Reorder the beam

We now have the k-best alignments using the local features from the last step. We then use global features to reorder the beam. The global features look at the properties of the entire alignment structure instead of the alignment links locally.

Let the global feature vector be represented as $f_G(\bar{a})$. The global score is defined as the dot product of the weight vector and the global feature vector.

$$score_G(\bar{a}) = W.f_G(\bar{a})$$

The overall score is calculated by adding the local score and the global score.

$$score(\bar{a}) = score_{La}(\bar{a}) + score_G(\bar{a})$$

The beam is now sorted based on the overall scores of each alignment. The alignment at the top of the beam is the best possible alignment between source sentence and the target sentence.

2.3 Post-processing

The previous two steps produce alignment structures which contain one-to-one and many-to-one links. In this step, the goal is to extend the best alignment structure obtained in the previous step to include the other alignments links of one-to-many and many-to-many types.

The majority of the links between the source sentence and the target sentence are one-to-one. Some of the cases where this is not true are the instances of idioms, alignment of verb groups where auxiliaries do not correspond to each other, the alignment of case-markers etc. Except for the cases of idioms in target language, most of the many-to-many links between a source and target sentences can be inferred from the instances of one-to-one and many-to-one links using three language language specific rules (Hindi in our case) to handle the above cases. Figure 1, Figure 2 and Figure 3 depict the three such cases where many-to-many alignments can be inferred. The alignments present at the left are those which can be predicted by our alignment model. The alignments on the right side are those which can be inferred in the post-processing stage.

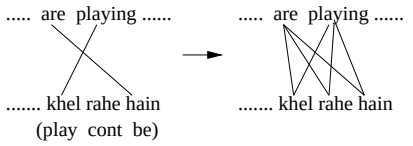

Figure 2: Inferring the many-to-many alignments of verb and auxiliaries

After applying the language specific rules, the dependency structure of the source sentence is traversed to ensure the consistency of the alignment

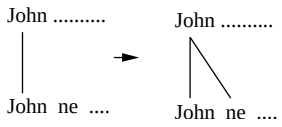

Figure 3: Inferring the one-to-many alignment to case-markers in Hindi

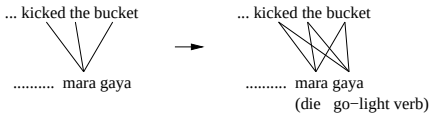

Figure 4: Inferring many-to-many alignment for source idioms

structure. If there is a dependency link between two source words e_o and e_p, where e_o is the head and e_p is the modifier and if e_o and e_p are linked to one or more common target word(s), it is logical to imagine that the alignment should be extended such that both e_o and e_p are linked to the same set of target words. For example, in Figure 4, new alignment link is first formed between 'kick' and 'gayA' using the language specific rule, and as 'kick' and 'bucket' are both linked to 'mara', 'bucket' is also now linked to 'gayA'. Similarity, 'the' is linked to both 'mara' and 'gayA'. Hence, the rules are applied by traversing through the dependency tree associated with the source sentence words in depth-first order. The dependency parser used by us was developed by (Shen, 2006). The following summarizes this step,

- Let w be the next word considered in the dependency tree, let pw be the parent of w.
  - If w and pw are linked to one or more common word(s) in target language, align w to all target words which are aligned to pw.
  - Else, Use the target-specific rules (if they match) to extend the alignments of w.
- Recursively consider all the children of w

3 Parameters

As the number of training examples is small, we chose to use features (both local and structural) which are generic. Some of the features which we used in this experiment are as follows:

3.1 Local features (F_L)

The local features which we consider are mainly co-occurrence features. These features estimate the likelihood of a source word aligning to a target word based on the co-occurrence information obtained from a large sentence aligned corpora[1].

3.1.1 DiceWords

Dice Coefficient of the source word and the target word (Taskar et al., 2005).

$$\mathit{DCoeff}(e_p, h_q) = \frac{2 * \mathit{Count}(e_p, h_q)}{\mathit{Count}(e_p) + \mathit{Count}(h_q)}$$

where $\mathit{Count}(e_p, h_q)$ is the number of times the word h_q was present in the translation of sentences containing the word e_p in the parallel corpus.

3.1.2 DiceRoots

Dice Coefficient of the lemmatized forms of the source and target words. It is important to consider this feature for language pairs which do not have a large unsupervised sentence aligned corpora. Co-occurrence information can be learnt better after we lemmatize the words.

3.1.3 Dict

This feature tests whether there exists a dictionary entry from the source word e_p to the target word h_q. For English-Hindi, we used a medium-coverage dictionary (25000 words) available from IIIT - Hyderabad, India [2].

3.1.4 Null_POS

These parameters measures the likelihood of a source word with a particular part of speech tag[3] to be aligned to no word (Null) on the target language side. This feature was extremely useful because it models the cooccurence information of words with nulls which is not captured by the features *DiceWords* and *DiceRoots*. Here are some of the features of this type with extreme estimated parameter weights.

3.2 Lemmatized word pairs

The word pairs themselves are a good indicator of whether an alignment link exists between the word pair or not. Also, taking word-pairs as feature helps in the alignment of some of the most common words in both the languages. A variation of this feature was used by (Moore, 2005) in his paper.

[1] 50K sentence pairs originally collected as part of TIDES MT project and later refined at IIIT-Hyderabad, India.

[2] http://ltrc.iiit.ac.in/onlineServices/Dictionaries/Dict_Frame.html

[3] We have limited the number of POS tags by considering only the first alphabets of Penn Tags as our POS tag categories

Param.	*weight*		*Param.*	*weight*
Null_‘	0.2737		null_C	-0.7030
Null_U	0.1969		null_D	-0.6914
Null_L	0.1814		null_V	-0.6360
Null_.	0.0383		null_N	-0.5600
Null_:	0.0055		null_I	-0.4839

Table 1: Top Five Features each with Maximum and Minimum weights

Other parameters like the relative distance between the source word e_p and the target word h_q, $\mathit{RelDist}(e_p, h_q) = \mathit{abs}(j/|e| - k/|h|)$, which are mentioned as important features in the previous literature, did not perform well for the English-Hindi language pair. This is because of the predominant word-order variation between the sentences of English and Hindi (Refer Figure 1).

3.3 Structural Features (F_G)

The global features are used to model the properties of the entire alignment structure taken as a unit, between the source and the target sentence. In doing so, we have attempted to exploit the syntactic information available on both the source and the target sides of the corpus. The syntactic information on the target side is obtained by projecting the syntactic information of the source using the alignment links. Some of the features which we have used in our work are in the following subsection.

3.3.1 Overlap

This feature considers the instances in a sentence pair where a source word links to a target word which is a participant in more than one alignment links (has a fertility greater than one). This feature is used to encourage the source words to be linked to different words in the target language. For example, we would prefer the alignment in Figure 6 when compared to the alignment in Figure 5 even before looking at the actual words. This parameter captures such prior information about the alignment structure.

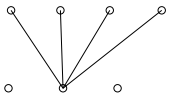

Figure 5: Alignment where many source words are linked to one target word

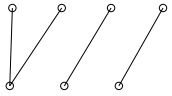

Figure 6: Alignment where the source words are aligned to many different target words

Formally, it is defined as

$$\mathit{Overlap}(\bar{a}) = \frac{\sum_{h_q \in T, \mathit{Fert}(h_q)>1} \mathit{Fert}^2(h_q)}{\sum_{h \in T} \mathit{Fert}(h)}$$

where T is the Hindi sentence. $\sum \mathit{Fert}^2(h_q)$ is measured in the numerator so that a more uniform distribution of target word fertilities be favored in comparison to others. The weight of *overlap* as estimated by our model is -6.1306 which indicates the alignments having a low overlap value are preferred.

3.3.2 NullPercent

This feature measures the percentage of words in target language sentence which are not aligned to any word in the source language sentence. It is defined as

$$\mathit{NullPercent} = \frac{|h_q|_{h_q \in T, \mathit{Fertility}(h_q)==0}}{|h|_{h \in T}}$$

3.3.3 Direction_DepPair

The following feature attempts to capture the first order interdependence between the alignment links of pairs of source sentence words which are connected by dependency relations. One way in which such an interdependence can be measured is by noting the order of the target sentence words linked to the child and parent of a source sentence dependency relation. Figures 7, 8 and 9 depict the various possibilities. The words in the source sentence are represented using their part-of-speech tags. These part-of-speech tags are also projected onto the target words. In the figures p is the parent and c is the part-of-speech of the child.

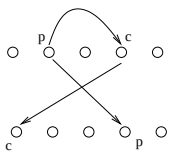

Figure 7: Target word linked to a child precedes the target word linked to a parent

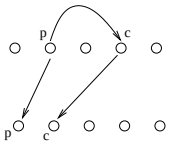

Figure 8: Target word linked to a parent precedes the target word linked to a child

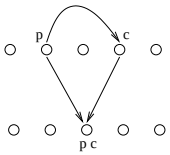

Figure 9: Parent and the child are both linked to same target word

The situation in Figure 9 is an indicator that the parent and child dependency pair might be part or whole of a multi-word expression on the source side. This feature thus captures the divergence between the source sentence dependency structure and the target language dependency structure (induced by taking the alignment as a constraint). Hence, in the test data, the alignments which do not express this divergence between the dependency trees are penalized. For example, the alignment in Figure 10 will be heavily penalized by the model during re-ranking step primarily for two reasons, 1) The word aligned to the preposition 'of' does not precede the word aligned to the noun 'king' and 2) The word aligned to the preposition 'to' does not succeed the word aligned to the noun 'king'.

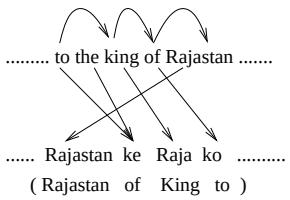

Figure 10: A simple example of an alignment that would be penalized by the feature Direction_DepPair

3.3.4 Direction_Bigram

This feature is a variation of the previous feature. In the previous feature, the dependency pair on the source side was projected to the target side to observe the divergence of the dependency pair. In this feature, we take a bigram instead of a dependency pair and observe its order in the target side. This feature is equivalent to the first-order features used in the related work.

There are three possibilities here, (1) The words of the bigram maintain their order when projected onto the target words, (2) The words of the bigram are reversed when projected, (3) Both the words are linked to the same word of the target sentence.

4 Online large margin training

For parameter optimization, we have used an online large margin algorithm called MIRA (McDonald et al., 2005) (Crammer and Singer, 2003). We will briefly describe the training algorithm that we have used. Our training set is a set of English-Hindi word aligned parallel corpus. Let the number of sentence pairs in the training data be t. We have $\{S_r, T_r, \hat{a}_r\}$ for training where $r \leq t$ is the index number of the sentence pair $\{S_r, T_r\}$ in the training set and $\hat{a}_r$ is the gold alignment for the pair $\{S_r, T_r\}$. Let W be the weight vector which has to be learnt, W_i be the weight vector after the end of i^{th} update. To avoid over-fitting, W is obtained by averaging over all the weight vectors W_i.

A generic large margin algorithm is defined follows for the training instances $\{S_r, T_r, \hat{a}_r\}$,

Initialize W_0, W, i
for $p = 1$ to *Iterations* **do**
  for $r = 1$ to *t* **do**
    Get K-Best predictions $\alpha_r = \{a_1, a_2 ... a_k\}$ for the training example $(S_r, T_r, \hat{a_r})$ using the current model W^i and applying step 1 and 2 of section 4. Compute W^{i+1} by updating W^i based on $(S_r, T_r, \hat{a_r}, \alpha_r)$.
    i = i + 1
    $W = W + W^{i+1}$
    $W = \frac{W}{Iterations * m}$
  end for
end for

The goal of MIRA is to minimize the change in W^i such that the score of the gold alignment $\hat{a}$ exceeds the score of each of the predictions in α by a margin which is equal to the number of mistakes in the predictions when compared to the gold alignment. One could choose a different loss function which assigns greater penalty for certain kinds of mistakes when compared to others.

Step 4 (*Get K-Best predictions*) in the algo-

rithm mentioned above can be substituted by the following optimization problem,

$$\begin{aligned} &\text{minimize } \|(W^{i+1} - W^i)\| \\ &\text{s.t. } \forall k,\, score(\hat{a}_r, S_r, T_r) - score(a_{q,k}, S_r, T_r) \\ &\qquad >= Mistakes(a_k, \hat{a}_r, S_r, T_r) \end{aligned}$$

For optimization of the parameters, ideally, we need to consider all the possible predictions and assign margin constraints based on every prediction. But, here the number of such classes is exponential and therefore we restrict ourselves to the $k - best$ predictions.

We estimate the parameters in two steps. In the first step, we estimate only the weights of the local parameters. After that, we keep the weights of local parameters constant and then estimate the weights of global parameters. It is important to decouple the parameter estimation to two steps. We also experimented estimating the parameters in one stage but as expected, it had an adverse impact on the parameter weights of local features which resulted in generation of poor k-best list after the first step while testing.

5 Experiments and Results

5.1 Data

We have used English-Hindi unsupervised data of 50000 sentence pairs[4]. This data was used to obtain the cooccurence statistics such as *DiceWords* and *DiceRoots* which we used in our model. This data was also used to obtain the predictions of GIZA++ (Implements the IBM models and the HMM model). We take the alignments of GIZA++ as baseline and evaluate our model for the English-Hindi language pair.

The supervised training data which is used to estimate the parameters consists of 4252 sentence pairs. The development data consists of 100 sentence pairs and the test data consists of 100 sentence pairs. This supervised data was obtained from IRCS, University of Pennsylvania. For training our model, we need to convert the many-to-many alignments in the corpus to one-to-one or may-to-one alignments. This is done by applying inverse operations of those performed during the post-processing step (section 2.3).

[4]Originally collected as part of TIDES MT project and later refined at IIIT-Hyderabad, India.

5.2 Experiments

We first obtain the predictions of GIZA++ to obtain the baseline accuracies. GIZA++ was run in four different modes 1) English to Hindi, 2) Hindi to English, 3) English to Hindi where the words in both the languages are lemmatized and 4) Hindi to English where the words are lemmatized. We then take the intersections of the predictions run from both the directions (English to Hindi and Hindi to English). Table 2 contains the results of experiments with GIZA++. As the recall of the alignment links of the intersection is very low for this dataset, further refinements of the alignments as suggested by (Och and Ney, 2003) were not performed.

Mode	*Prec.*	*Rec.*	*F-meas.*	*AER*
Normal: Eng-Hin	47.57	40.87	43.96	**56.04**
Normal: Hin-Eng	47.97	38.50	42.72	57.28
Normal: Inter.	88.71	27.52	42.01	57.99
Lemma.: Eng-Hin	53.60	44.58	48.67	**51.33**
Lemma.: Hin-Eng	53.83	42.68	47.61	52.39
Lemma.: Inter.	86.14	32.80	47.51	52.49

Table 2: GIZA++ Results

In Table 3, we observe that the best result (**51.33**) is obtained when GIZA++ is run after lemmatizing the words on the both sides of the unsupervised corpus. The best results obtained without lemmatizing is **56.04** when GIZA++ is run from English to Hindi.

The table 4 summarizes the results when we used only the local features in our model.

Features	*Prec.*	*Rec.*	*F-meas.*	*AER*
DiceRoots + *DiceWords*	41.49	38.71	40.05	**59.95**
+ *Null_POS*	42.82	38.29	40.43	59.57
+ *Dict.*	43.94	39.30	41.49	58.51
+ *Word pairs*	46.27	41.07	43.52	56.48

Table 3: Results using local features

We now add the global features. While estimating the parameter weights associated with the global features, we keep the weights of local features constant. We choose the appropriate beam size as 50 after testing with several values on the development set. We observed that the beam sizes (between 10 and 100) did not affect the alignment error rates very much.

Features	*Prec.*	*Rec.*	*F-meas.*	*AER*
Local feats.	46.27	41.07	43.52	56.48
Local feats. + *Overlap*	48.17	42.76	45.30	54.70
Local feats. + *Direc._Deppair*	47.93	42.55	45.08	54.92
Local feats. + *Direc._Bigram*	48.31	42.89	45.44	54.56
Local feats. + *All Global feats.*	48.81	43.31	45.90	**54.10**

Table 4: Results after adding global features

We see that by adding global features, we obtained an absolute increase of about 2.3 AER suggesting the usefulness of structural features which we considered. Also, the new AER is much better than that obtained by GIZA++ run without lemmatizing the words.

We now add the IBM Model-4 parameters (co-occurrence probabilities between source and target words) obtained using GIZA++ and our features, and observe the results (Table 6). We can see that structural features resulted in a significant decrease in AER. Also, the AER that we obtained is slightly better than the best AER obtained by the GIZA++ models.

Features	*Prec.*	*Rec.*	*F-meas.*	*AER*
IBM Model-4 Pars. + $LocalFeats$	48.85	43.98	46.29	52.71
IBM Model-4 Pars. + *All feats.*	48.95	50.06	49.50	**50.50**

Table 5: Results after combining IBM model-4 parameters with our features

6 Conclusion and Future Work

In this paper, we have proposed a discriminative re-ranking approach for word alignment which allows us to make use of structural features effectively. We have shown that by using the structural features, we have obtained a decrease of 2.3% in the absolute value of alignment error rate (AER). When we combine the prediction of IBM model-4 with our features, we have achieved an AER which is slightly better than the best AER of GIZA++ for the English-Hindi parallel corpus (a language pair with significant structural divergences). We expect to get large improvements when we add more number of relevant local and structural features. We also plan to design an appropriate dependency based decoder for machine translation to make good use of the parameters estimated by our model.

References

Phil Blunsom and Trevor Cohn. 2006. Discriminative word alignment with conditional random fields. In *Proceedings of the 21st COLING and 44th Annual Meeting of the ACL*, Sydney, Australia, July. ACL.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. In *Journal of Machine Learning Research*.

Alexander Fraser and Daniel Marcu. 2006. Semi-supervised training for statistical word alignment. In *Proceedings of the 21st COLING and 44th Annual Meeting of the ACL*, Sydney, Australia, July. Association for Computational Linguistics.

Simon Lacoste-Julien, Ben Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 112–119, New York City, USA, June. Association for Computational Linguistics.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-project dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver, British Columbia, Canada, October. Association of Computational Linguistics.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 81–88, Vancouver, British Columbia, Canada, October. Association of Computational Linguistics.

F. Och and H. Ney. 2003. A systematic comparisoin of various statistical alignment models. In *Computational Linguistics*.

Libin Shen. 2006. *Statistical LTAG Parsing*. Ph.D. thesis.

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

Stefan Vogel, Hermann Ney, and Christoph Tillmann. 1996. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th International Conference on Computational Linguistics*.