2006

Word Alignment via Quadratic Assignment

Simon Lacoste-Julien
University of California - Berkeley

Ben Taskar
University of Pennsylvania, taskar@cis.upenn.edu

Dan Klein
University of California - Berkeley

Michael Jordan
University of California - Berkeley

Simon Lacoste-Julien, Ben Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (HLT-NAACL '06). Association for Computational Linguistics, Stroudsburg, PA, USA, 112-119. DOI=10.3115/1220835.1220850 http://dx.doi.org/10.3115/1220835.1220850

Word Alignment via Quadratic Assignment

Abstract
Recently, discriminative word alignment methods have achieved state-of-the-art accuracies by extending the range of information sources that can be easily incorporated into aligners. The chief advantage of a discriminative framework is the ability to score alignments based on arbitrary features of the matching word tokens, including orthographic form, predictions of other models, lexical context and so on. However, the proposed bipartite matching model of Taskar et al. (2005), despite being tractable and effective, has two important limitations. First, it is limited by the restriction that words have fertility of at most one. More importantly, first order correlations between consecutive words cannot be directly captured by the model. In this work, we address these limitations by enriching the model form. We give estimation and inference algorithms for these enhancements. Our best model achieves a relative AER reduction of 25% over the basic matching formulation, outperforming intersected IBM Model 4 without using any overly compute-intensive features. By including predictions of other models as features, we achieve AER of 3:8 on the standard Hansards dataset.

Disciplines
Computer Sciences

Comments
Simon Lacoste-Julien, Ben Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics* (HLT-NAACL '06). Association for Computational Linguistics, Stroudsburg, PA, USA, 112-119. DOI=10.3115/1220835.1220850 http://dx.doi.org/10.3115/1220835.1220850

Word Alignment via Quadratic Assignment

Simon Lacoste-Julien
UC Berkeley, Berkeley, CA 94720
slacoste@cs.berkeley.edu

Ben Taskar
UC Berkeley, Berkeley, CA 94720
taskar@cs.berkeley.edu

Dan Klein
UC Berkeley, Berkeley, CA 94720
klein@cs.berkeley.edu

Michael I. Jordan
UC Berkeley, Berkeley, CA 94720
jordan@cs.berkeley.edu

Abstract

Recently, discriminative word alignment methods have achieved state-of-the-art accuracies by extending the range of information sources that can be easily incorporated into aligners. The chief advantage of a discriminative framework is the ability to score alignments based on arbitrary features of the matching word tokens, including orthographic form, predictions of other models, lexical context and so on. However, the proposed bipartite matching model of Taskar et al. (2005), despite being tractable and effective, has two important limitations. First, it is limited by the restriction that words have fertility of at most one. More importantly, first order correlations between consecutive words cannot be directly captured by the model. In this work, we address these limitations by enriching the model form. We give estimation and inference algorithms for these enhancements. Our best model achieves a relative AER reduction of 25% over the basic matching formulation, outperforming intersected IBM Model 4 without using any overly compute-intensive features. By including predictions of other models as features, we achieve AER of 3.8 on the standard Hansards dataset.

1 Introduction

Word alignment is a key component of most end-to-end statistical machine translation systems. The standard approach to word alignment is to construct directional generative models (Brown et al., 1990; Och and Ney, 2003), which produce a sentence in one language given the sentence in another language. While these models require sentence-aligned bitexts, they can be trained with no further supervision, using EM. Generative alignment models do, however, have serious drawbacks. First, they require extensive tuning and processing of large amounts of data which, for the better-performing models, is a non-trivial resource requirement. Second, conditioning on arbitrary features of the input is difficult; for example, we would like to condition on the orthographic similarity of a word pair (for detecting cognates), the presence of that pair in various dictionaries, the similarity of the frequency of its two words, choices made by other alignment systems, and so on.

Recently, Moore (2005) proposed a discriminative model in which pairs of sentences (e, f) and proposed alignments a are scored using a linear combination of arbitrary features computed from the tuples (a, e, f). While there are no restrictions on the form of the model features, the problem of finding the highest scoring alignment is very difficult and involves heuristic search. Moreover, the parameters of the model must be estimated using averaged perceptron training (Collins, 2002), which can be unstable. In contrast, Taskar et al. (2005) cast word alignment as a maximum weighted matching problem, in which each pair of words (e_j, f_k) in a sentence pair (e, f) is associated with a score $s_{jk}(e, f)$ reflecting the desirability of the alignment of that pair. Importantly, this problem is computationally tractable. The alignment for the sentence pair is the highest scoring matching under constraints (such as the constraint that matchings be one-to-one). The scoring model $s_{jk}(e, f)$ can be based on a rich feature set defined on word pairs (e_j, f_k) and their context, including measures of association, orthography, relative position, predictions of generative models, etc. The parameters of the model are estimated within the framework of large-margin estimation; in particular, the problem turns out to reduce to the

solution of a (relatively) small quadratic program (QP). The authors show that large-margin estimation is both more stable and more accurate than perceptron training.

While the bipartite matching approach is a useful first step in the direction of discriminative word alignment, for discriminative approaches to compete with and eventually surpass the most sophisticated generative models, it is necessary to consider more realistic underlying statistical models. Note in particular two substantial limitations of the bipartite matching model of Taskar et al. (2005): words have fertility of at most one, and there is no way to incorporate pairwise interactions among alignment decisions. Moving beyond these limitations—while retaining computational tractability—is the next major challenge for discriminative word alignment.

In this paper, we show how to overcome both limitations. First, we introduce a parameterized model that penalizes different levels of fertility. While this extension adds very useful expressive power to the model, it turns out not to increase the computational complexity of the aligner, for either the prediction or the parameter estimation problem. Second, we introduce a more thoroughgoing extension which incorporates first-order interactions between alignments of consecutive words into the model. We do this by formulating the alignment problem as a quadratic assignment problem (QAP), where in addition to scoring individual edges, we also define scores of pairs of edges that connect consecutive words in an alignment. The predicted alignment is the highest scoring quadratic assignment.

QAP is an NP-hard problem, but in the range of problem sizes that we need to tackle the problem can be solved efficiently. In particular, using standard off-the-shelf integer program solvers, we are able to solve the QAP problems in our experiments in under a second. Moreover, the parameter estimation problem can also be solved efficiently by making use of a linear relaxation of QAP for the min-max formulation of large-margin estimation (Taskar, 2004).

We show that these two extensions yield significant improvements in error rates when compared to the bipartite matching model. The addition of a fertility model improves the AER by 0.4. Modeling first-order interactions improves the AER by 1.8. Combining the two extensions results in an improvement in AER of 2.3, yielding alignments of better quality than intersected IBM Model 4. Moreover, including predictions of bi-directional IBM Model 4 and model of Liang et al. (2006) as features, we achieve an absolute AER of 3.8 on the English-French Hansards alignment task—the best AER result published on this task to date.

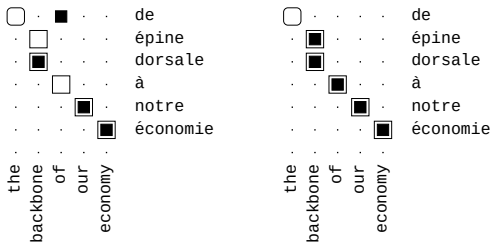
2 Models

We begin with a quick summary of the maximum weight bipartite matching model in (Taskar et al., 2005). More precisely, nodes $\mathcal{V} = \mathcal{V}^s \cup \mathcal{V}^t$ correspond to words in the "source" ($\mathcal{V}^s$) and "target" ($\mathcal{V}^t$) sentences, and edges $\mathcal{E} = \{jk : j \in \mathcal{V}^s, k \in \mathcal{V}^t\}$ correspond to alignments between word pairs.[1] The edge weights s_{jk} represent the degree to which word j in one sentence can be translated using the word k in the other sentence. The predicted alignment is chosen by maximizing the sum of edge scores. A matching is represented using a set of binary variables y_{jk} that are set to 1 if word j is assigned to word k in the other sentence, and 0 otherwise. The score of an assignment is the sum of edge scores: $s(\mathbf{y}) = \sum_{jk} s_{jk} y_{jk}$. For simplicity, let us begin by assuming that each word aligns to one or zero words in the other sentence; we revisit the issue of fertility in the next section. The maximum weight bipartite matching problem, $\arg\max_{\mathbf{y} \in \mathcal{Y}} s(\mathbf{y})$, can be solved using combinatorial algorithms for min-cost max-flow, expressed in a linear programming (LP) formulation as follows:

$$\begin{aligned} \max_{0 \le \mathbf{z} \le 1} \quad & \sum_{jk \in \mathcal{E}} s_{jk} z_{jk} \\ \text{s.t.} \quad & \sum_{j \in \mathcal{V}^s} z_{jk} \le 1, \ \forall k \in \mathcal{V}^t; \\ & \sum_{k \in \mathcal{V}^t} z_{jk} \le 1, \ \forall j \in \mathcal{V}^s, \end{aligned} \tag{1}$$

where the continuous variables z_{jk} are a relaxation of the corresponding binary-valued variables y_{jk}. This LP is guaranteed to have integral (and hence optimal) solutions for any scoring function $s(\mathbf{y})$ (Schrijver, 2003). Note that although the above LP can be used to compute alignments, combinatorial algorithms are generally more efficient. For

[1] The source/target designation is arbitrary, as the models considered below are all symmetric.



(a) (b)

Figure 2: An example fragment that requires fertility greater than one to correctly label. (a) The guess of the baseline M model. (b) The guess of the M+F fertility-augmented model.

example, in Figure 1(a), we show a standard construction for an equivalent min-cost flow problem. However, we build on this LP to develop our extensions to this model below. Representing the prediction problem as an LP or an integer LP provides a precise (and concise) way of specifying the model and allows us to use the large-margin framework of Taskar (2004) for parameter estimation described in Section 3.

For a sentence pair $\mathbf{x}$, we denote position pairs by $\mathbf{x}_{jk}$ and their scores as s_{jk}. We let $s_{jk} = \mathbf{w}^\top \mathbf{f}(\mathbf{x}_{jk})$ for some user provided feature mapping $\mathbf{f}$ and abbreviate $\mathbf{w}^\top \mathbf{f}(\mathbf{x}, \mathbf{y}) = \sum_{jk} y_{jk} \mathbf{w}^\top \mathbf{f}(\mathbf{x}_{jk})$. We can include in the feature vector the identity of the two words, their relative positions in their respective sentences, their part-of-speech tags, their string similarity (for detecting cognates), and so on.

2.1 Fertility

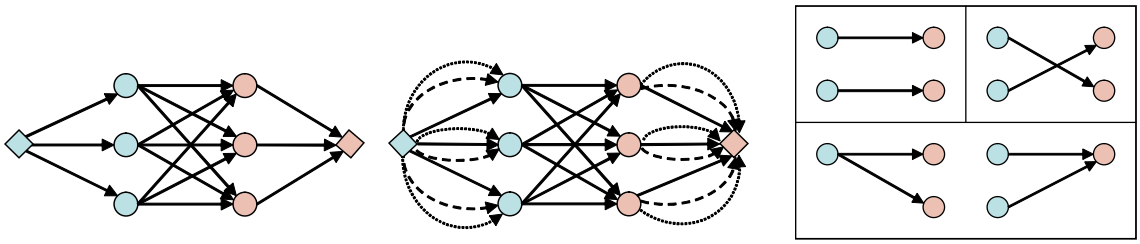
An important limitation of the model in Eq. (1) is that in each sentence, a word can align to at most one word in the translation. Although it is common that words have gold fertility zero or one, it is certainly not always true. Consider, for example, the bitext fragment shown in Figure 2(a), where *backbone* is aligned to the phrase *épine dorsal*. In this figure, outlines are gold alignments, square for sure alignments, round for possibles, and filled squares are target alignments (for details on gold alignments, see Section 4). When considering only the sure alignments on the standard Hansards dataset, 7 percent of the word occurrences have fertility 2, and 1 percent have fertility 3 and above; when considering the possible alignments high fertility is much more common—31 percent of the words have fertility 3 and above.

One simple fix to the original matching model is to increase the right hand sides for the constraints in Eq. (1) from 1 to D, where D is the maximum allowed fertility. However, this change results in an undesirable bimodal behavior, where maximum weight solutions either have all words with fertility 0 or D, depending on whether most scores $\mathbf{s}_{jk}$ are positive or negative. For example, if scores tend to be positive, most words will want to collect as many alignments as they are permitted. What the model is missing is a means for encouraging the common case of low fertility (0 or 1), while allowing higher fertility when it is licensed. This end can be achieved by introducing a penalty for having higher fertility, with the goal of allowing that penalty to vary based on features of the word in question (such as its frequency or identity).

In order to model such a penalty, we introduce indicator variables $z_{dj\bullet}$ (and $z_{d\bullet k}$) with the intended meaning: node j has fertility of at least d (and node k has fertility of at least d). In the following LP, we introduce a penalty of $\sum_{2 \le d \le D} s_{dj\bullet} z_{dj\bullet}$ for fertility of node j, where each term $s_{dj\bullet} \ge 0$ is the penalty increment for increasing the fertility from $d-1$ to d:

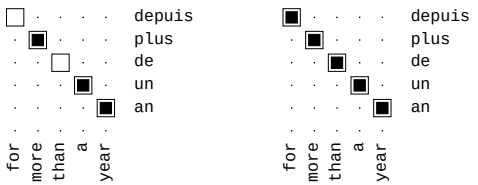
$$\max_{0 \le \mathbf{z} \le 1} \sum_{jk \in \mathcal{E}} s_{jk} z_{jk} \qquad (2)$$
$$- \sum_{j \in \mathcal{V}^s, 2 \le d \le D} s_{dj\bullet} z_{dj\bullet} - \sum_{k \in \mathcal{V}^t, 2 \le d \le D} s_{d\bullet k} z_{d\bullet k}$$
$$\text{s.t.} \quad \sum_{j \in \mathcal{V}^s} z_{jk} \le 1 + \sum_{2 \le d \le D} z_{d\bullet k}, \quad \forall k \in \mathcal{V}^t;$$
$$\sum_{k \in \mathcal{V}^t} z_{jk} \le 1 + \sum_{2 \le d \le D} z_{dj\bullet}, \quad \forall j \in \mathcal{V}^s.$$

We can show that this LP always has integral solutions by a reduction to a min-cost flow problem. The construction is shown in Figure 1(b). To ensure that the new variables have the intended semantics, we need to make sure that $s_{dj\bullet} \le s_{d'j\bullet}$ if $d \le d'$, so that the lower cost $z_{dj\bullet}$ is used before the higher cost $z_{d'j\bullet}$ to increase fertility. This restriction im-



(a) (b) (c)

Figure 1: (a) Maximum weight bipartite matching as min-cost flow. Diamond-shaped nodes represent flow source and sink. All edge capacities are 1, with edges between round nodes (j, k) have cost $-s_{jk}$, edges from source and to sink have cost 0. (b) Expanded min-cost flow graph with new edges from source and to sink that allow fertility of up to 3. The capacities of the new edges are 1 and the costs are 0 for solid edges from source and to sink, $s_{2j\bullet}, s_{2\bullet k}$ for dashed edges, and $s_{3j\bullet}, s_{3\bullet k}$ for dotted edges. (c) Three types of pairs of edges included in the QAP model, where the nodes on both sides correspond to consecutive words.



(a) (b)

Figure 3: An example fragment with a monotonic gold alignment. (a) The guess of the baseline M model. (b) The guess of the M+Q quadratic model.

plies that the penalty must be monotonic and convex as a function of the fertility.

To anticipate the results that we report in Section 4, adding fertility to the basic matching model makes the target alignment of the *backbone* example feasible and, in this case, the model correctly labels this fragment as shown in Figure 2(b).

2.2 First-order interactions

An even more significant limitation of the model in Eq. (1) is that the edges interact only indirectly through the competition induced by the constraints. Generative alignment models like the HMM model (Vogel et al., 1996) and IBM models 4 and above (Brown et al., 1990; Och and Ney, 2003) directly model correlations between alignments of consecutive words (at least on one side). For example, Figure 3 shows a bitext fragment whose gold alignment is strictly monotonic. This monotonicity is quite common – 46% of the words in the hand-aligned data diagonally follow a previous alignment in this way. We can model the common local alignment configurations by adding bonuses for pairs of edges. For example, strictly monotonic alignments can be encouraged by boosting the scores of edges of the form $\langle (j, k), (j + 1, k + 1) \rangle$. Another trend, common in English-French translation (7% on the hand-aligned data), is the local inversion of nouns and adjectives, which typically involves a pair of edges $\langle (j, k + 1), (j + 1, k) \rangle$. Finally, a word in one language is often translated as a phrase (consecutive sequence of words) in the other language. This pattern involves pairs of edges with the same origin on one side: $\langle (j, k), (j, k+1) \rangle$ or $\langle (j, k), (j+1, k) \rangle$. All three of these edge pair patterns are shown in Figure 1(c). Note that the set of such edge pairs $\mathcal{Q} = \{jklm : |j - l| \leq 1, |k - m| \leq 1\}$ is of linear size in the number of edges.

Formally, we add to the model variables z_{jklm} which indicate whether both edge jk and lm are in the alignment. We also add a corresponding score s_{jklm}, which we assume to be non-negative, since the correlations we described are positive. (Negative scores can also be used, but the resulting formulation we present below would be slightly different.) To enforce the semantics $z_{jklm} = z_{jk} z_{lm}$, we use a pair of constraints $z_{jklm} \leq z_{jk}; z_{jklm} \leq z_{lm}$. Since s_{jklm} is positive, at the optimum, $z_{jklm} =$

$\min(z_{jk}, z_{lm})$. If in addition z_{jk}, z_{lm} are integral (0 or 1), then $z_{jklm} = z_{jk} z_{lm}$. Hence, solving the following LP as an integer linear program will find the optimal quadratic assignment for our model:

$$\begin{aligned} \max_{0 \le \mathbf{z} \le 1} \quad & \sum_{jk \in \mathcal{E}} s_{jk} z_{jk} + \sum_{jklm \in \mathcal{Q}} s_{jklm} z_{jklm} \qquad (3) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{V}^s} z_{jk} \le 1, \ \forall k \in \mathcal{V}^t; \\ & \sum_{k \in \mathcal{V}^t} z_{jk} \le 1, \ \forall j \in \mathcal{V}^s; \\ & z_{jklm} \le z_{jk}, \quad z_{jklm} \le z_{lm}, \ \forall jklm \in \mathcal{Q}. \end{aligned}$$

Note that we can also combine this extension with the fertility extension described above.

To once again anticipate the results presented in Section 4, the baseline model of Taskar et al. (2005) makes the prediction given in Figure 3(a) because the two missing alignments are atypical translations of common words. With the addition of edge pair features, the overall monotonicity pushes the alignment to that of Figure 3(b).

3 Parameter estimation

To estimate the parameters of our model, we follow the large-margin formulation of Taskar (2004). Our input is a set of training instances $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, where each instance consists of a sentence pair $\mathbf{x}_i$ and a target alignment $\mathbf{y}_i$. We would like to find parameters $\mathbf{w}$ that predict correct alignments on the training data: $\mathbf{y}_i = \arg\max_{\bar{\mathbf{y}}_i \in \mathcal{Y}_i} \mathbf{w}^\top \mathbf{f}(\mathbf{x}_i, \bar{\mathbf{y}}_i)$ for each i, where $\mathcal{Y}_i$ is the space of matchings for the sentence pair $\mathbf{x}_i$.

In standard classification problems, we typically measure the error of prediction, $\ell(\mathbf{y}_i, \bar{\mathbf{y}}_i)$, using the simple 0-1 loss. In structured problems, where we are jointly predicting multiple variables, the loss is often more complex. While the F-measure is a natural loss function for this task, we instead chose a sensible surrogate that fits better in our framework: weighted Hamming distance, which counts the number of variables in which a candidate solution $\bar{\mathbf{y}}$ differs from the target output $\mathbf{y}$, with different penalty for false positives (c^+) and false negatives (c^-):

$$\ell(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{jk} \left[c^+ (1 - y_{jk}) \bar{y}_{jk} + c^- (1 - \bar{y}_{jk}) y_{jk} \right].$$

We use an SVM-like hinge upper bound on the loss $\ell(\mathbf{y}_i, \bar{\mathbf{y}}_i)$, given by $\max_{\bar{\mathbf{y}}_i \in \mathcal{Y}_i} [\mathbf{w}^\top \mathbf{f}_i(\bar{\mathbf{y}}_i) + \ell_i(\bar{\mathbf{y}}_i) - \mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i)]$, where $\ell_i(\bar{\mathbf{y}}_i) = \ell(\mathbf{y}_i, \bar{\mathbf{y}}_i)$, and $\mathbf{f}_i(\bar{\mathbf{y}}_i) = \mathbf{f}(\mathbf{x}_i, \bar{\mathbf{y}}_i)$. Minimizing this upper bound encourages the true alignment $\mathbf{y}_i$ to be optimal with respect to $\mathbf{w}$ for each instance i:

$$\min_{||\mathbf{w}|| \le \gamma} \sum_i \max_{\bar{\mathbf{y}}_i \in \mathcal{Y}_i} [\mathbf{w}^\top \mathbf{f}_i(\bar{\mathbf{y}}_i) + \ell_i(\bar{\mathbf{y}}_i)] - \mathbf{w}^\top \mathbf{f}_i(\mathbf{y}_i),$$

where γ is a regularization parameter.

In this form, the estimation problem is a mixture of continuous optimization over $\mathbf{w}$ and combinatorial optimization over $\mathbf{y}_i$. In order to transform it into a more standard optimization problem, we need a way to efficiently handle the *loss-augmented inference*, $\max_{\bar{\mathbf{y}}_i \in \mathcal{Y}_i} [\mathbf{w}^\top \mathbf{f}_i(\bar{\mathbf{y}}_i) + \ell_i(\bar{\mathbf{y}}_i)]$. This optimization problem has precisely the same form as the prediction problem whose parameters we are trying to learn — $\max_{\bar{\mathbf{y}}_i \in \mathcal{Y}_i} \mathbf{w}^\top \mathbf{f}_i(\bar{\mathbf{y}}_i)$ — but with an additional term corresponding to the loss function. Our assumption that the loss function decomposes over the edges is crucial to solving this problem. We omit the details here, but note that we can incorporate the loss function into the LPs for various models we described above and "plug" them into the large-margin formulation by converting the estimation problem into a quadratic problem (QP) (Taskar, 2004). This QP can be solved using any off-the-shelf solvers, such as MOSEK or CPLEX.[2] An important difference that comes into play for the estimation of the quadratic assignment models in Equation (3) is that inference involves solving an integer linear program, not just an LP. In fact the LP is a relaxation of the integer LP and provides an upper bound on the value of the highest scoring assignment. Using the LP relaxation for the large-margin QP formulation is an approximation, but as our experiments indicate, this approximation is very effective. At testing time, we use the integer LP to predict alignments. We have also experimented with using just the LP relaxation at testing time and then independently rounding each fractional edge value, which actually incurs no loss in alignment accuracy, as we discuss below.

[2] When training on 200 sentences, the QP we obtain contains roughly 700K variables and 300K constraints and is solved in roughly 10 minutes on a 2.8 GHz Pentium 4 machine. Aligning the whole training set with the fbw formulation takes a few seconds, whereas using the integer programming (for the QAP formulation) takes 1-2 minutes.

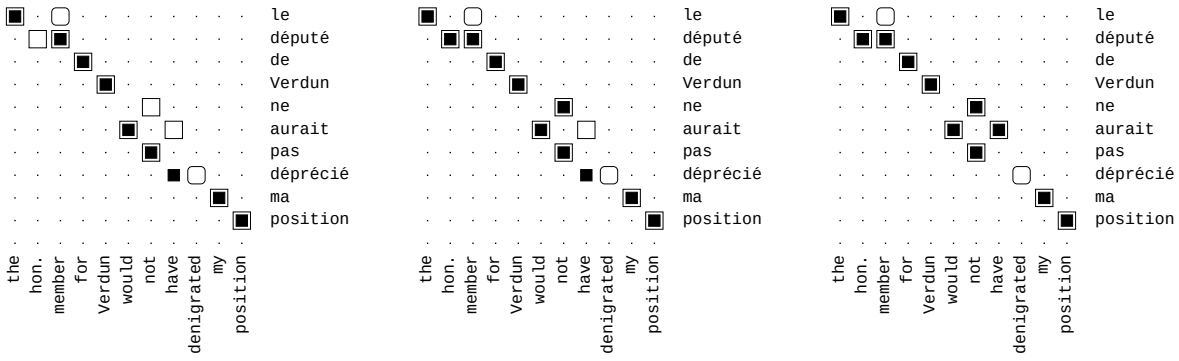


Figure 4: An example fragment with several multiple fertility sure alignments. (a) The guess of the M+Q model with maximum fertility of one. (b) The guess of the M+Q+F quadratic model with fertility two permitted. (c) The guess of the M+Q+F model with lexical fertility features.

4 Experiments

We applied our algorithms to word-level alignment using the English-French Hansards data from the 2003 NAACL shared task (Mihalcea and Pedersen, 2003). This corpus consists of 1.1M automatically aligned sentences, and comes with a validation set of 37 sentence pairs and a test set of 447 sentences. The validation and test sentences have been hand-aligned (see Och and Ney (2003)) and are marked with both *sure* and *possible* alignments. Using these alignments, *alignment error rate* (AER) is calculated as:

$$\left(1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}\right) \times 100\%.$$

Here, A is a set of proposed index pairs, S is the sure gold pairs, and P is the possible gold pairs. For example, in Figure 4, proposed alignments are shown against gold alignments, with open squares for sure alignments, rounded open squares for possible alignments, and filled black squares for proposed alignments.

The input to our algorithm is a small number of labeled examples. In order to make our results more comparable with Moore (2005), we split the original set into 200 training examples and 247 test examples. We also trained on only the first 100 to make our results more comparable with the experiments of Och and Ney (2003), in which IBM model 4 was tuned using 100 sentences. In all our experiments, we used a structured loss function that penalized false negatives 10 times more than false positives, where the value of 10 was picked by using a validation set. The regularization parameter γ was also chosen using the validation set.

4.1 Features and results

We parameterized all scoring functions s_{jk}, $s_{dj\bullet}$, $s_{d\bullet k}$ and s_{jklm} as weighted linear combinations of feature sets. The features were computed from the large unlabeled corpus of 1.1M automatically aligned sentences.

In the remainder of this section we describe the improvements to the model performance as various features are added. One of the most useful features for the basic matching model is, of course, the set of predictions of IBM model 4. However, computing these features is very expensive and we would like to build a competitive model that doesn't require them. Instead, we made significant use of IBM model 2 as a source of features. This model, although not very accurate as a predictive model, is simple and cheap to construct and it is a useful source of features.

The Basic Matching Model: Edge Features In the basic matching model of Taskar et al. (2005), called M here, one can only specify features on pairs of word tokens, i.e. alignment edges. These features

include word association, orthography, proximity, etc., and are documented in Taskar et al. (2005). We also augment those features with the predictions of IBM Model 2 run on the training and test sentences. We provided features for model 2 trained in each direction, as well as the intersected predictions, on each edge. By including the IBM Model 2 features, the performance of the model described in Taskar et al. (2005) on our test set (trained on 200 sentences) improves from 10.0 AER to 8.2 AER, outperforming unsymmetrized IBM Model 4 (but not intersected model 4).

As an example of the kinds of errors the baseline M system makes, see Figure 2 (where multiple fertility cannot be predicted), Figure 3 (where a preference for monotonicity cannot be modeled), and Figure 4 (which shows several multi-fertile cases).

The Fertility Model: Node Features To address errors like those shown in Figure 2, we increased the maximum fertility to two using the parameterized fertility model of Section 2.1. The model learns costs on the second flow arc for each word via features not of edges but of single words. The score of taking a second match for a word w was based on the following features: a bias feature, the proportion of times w's type was aligned to two or more words by IBM model 2, and the bucketed frequency of the word type. This model was called M+F. We also included a lexicalized feature for words which were common in our training set: whether w was ever seen in a multiple fertility alignment (more on this feature later). This enabled the system to learn that certain words, such as the English *not* and French verbs like *aurait* commonly participate in multiple fertility configurations.

Figure 5 show the results using the fertility extension. Adding fertility lowered AER from 8.5 to 8.1, though fertility was even more effective in conjunction with the quadratic features below. The M+F setting was even able to correctly learn some multiple fertility instances which were not seen in the training data, such as those shown in Figure 2.

The First-Order Model: Quadratic Features With or without the fertility model, the model makes mistakes such as those shown in Figure 3, where atypical translations of common words are not chosen despite their local support from adjacent edges. In the quadratic model, we can associate features with pairs of edges. We began with features which identify each specific pattern, enabling trends of monotonicity (or inversion) to be captured. We also added to each edge pair the fraction of times that pair's pattern (monotonic, inverted, one to two) occurred according each version of IBM model 2 (forward, backward, intersected).

Figure 5 shows the results of adding the quadratic model. M+Q reduces error over M from 8.5 to 6.7 (and fixes the errors shown in Figure 3). When both the fertility and quadratic extensions were added, AER dropped further, to 6.2. This final model is even able to capture the diamond pattern in Figure 4; the adjacent cycle of alignments is reinforced by the quadratic features which boost adjacency. The example in Figure 4 shows another interesting phenomenon: the multi-fertile alignments for *not* and *député* are learned even without lexical fertility features (Figure 4b), because the Dice coefficients of those words with their two alignees are both high. However the surface association of *aurait* with *have* is much higher than with *would*. If, however, lexical features are added, *would* is correctly aligned as well (Figure 4c), since it is observed in similar periphrastic constructions in the training set.

We have avoided using expensive-to-compute features like IBM model 4 predictions up to this point. However, if these are available, our model can improve further. By adding model 4 predictions to the edge features, we get a relative AER reduction of 27%, from 6.5 to 4.5. By also including as features the posteriors of the model of Liang et al. (2006), we achieve AER of 3.8, and 96.7/95.5 precision/recall.

It is comforting to note that in practice, the burden of running an integer linear program at test time can be avoided. We experimented with using just the LP relaxation and found that on the test set, only about 20% of sentences have fractional solutions and only 0.2% of all edges are fractional. Simple rounding[3] of each edge value in the LP solution achieves the same AER as the integer LP solution, while using about a third of the computation time on average.

[3]We slightly bias the system on the recall side by rounding 0.5 up, but this doesn't yield a noticeable difference in the results.

Model	Prec	Rec	AER
Generative			
IBM 2 (E→F)	73.6	87.7	21.7
IBM 2 (F→E)	75.4	87.0	20.6
IBM 2 (intersected)	90.1	80.4	14.3
IBM 4 (E→F)	90.3	92.1	9.0
IBM 4 (F→E)	90.8	91.3	9.0
IBM 4 (intersected)	98.0	88.1	6.5
Discriminative (100 sentences)			
Matching (M)	94.1	88.5	8.5
M + Fertility (F)	93.9	89.4	8.1
M + Quadratic (Q)	94.4	91.9	6.7
M + F + Q	94.8	92.5	6.2
M + F + Q + IBM4	96.4	94.4	4.5
Discriminative (200 sentences)			
Matching (M)	93.4	89.7	8.2
M + Fertility (F)	93.6	90.1	8.0
M + Quadratic (Q)	95.0	91.1	6.8
M + F + Q	95.2	92.4	6.1
M + F + Q + IBM4	96.0	95.0	4.4

Figure 5: AER on the Hansards task.

5 Conclusion

We have shown that the discriminative approach to word alignment can be extended to allow flexible fertility modeling and to capture first-order interactions between alignments of consecutive words. These extensions significantly enhance the expressive power of the discriminative approach; in particular, they make it possible to capture phenomena of monotonicity, local inversion and contiguous fertility trends—phenomena that are highly informative for alignment. They do so while remaining computationally efficient in practice both for prediction and for parameter estimation.

Our best model achieves a relative AER reduction of 25% over the basic matching formulation, beating intersected IBM Model 4 without the use of any compute-intensive features. Including Model 4 predictions as features, we achieve a further relative AER reduction of 32% over intersected Model 4 alignments. By also including predictions of another model, we drive AER down to 3.8. We are currently investigating whether the improvement in AER results in better translation BLEU score. Allowing higher fertility and optimizing a recall biased cost function provide a significant increase in recall relative to the intersected IBM model 4 (from 88.1% to 94.4%), with only a small degradation in precision. We view this as a particularly promising aspect of our work, given that phrase-based systems such as Pharaoh (Koehn et al., 2003) perform better with higher recall alignments.

References

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

M. Collins. 2002. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*.

P. Liang, B. Taskar, and D. Klein. 2006. Alignment by agreement. In *HLT-NAACL*.

R. Mihalcea and T. Pedersen. 2003. An evaluation exercise for word alignment. In *Proceedings of the HLT-NAACL 2003 Workshop, Building and Using parallel Texts: Data Driven Machine Translation and Beyond*, pages 1–6, Edmonton, Alberta, Canada.

Robert C. Moore. 2005. A discriminative framework for bilingual word alignment. In *Proc. HLT/EMNLP*.

F. Och and H. Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52.

A. Schrijver. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer.

B. Taskar, S. Lacoste-Julien, and D. Klein. 2005. A discriminative matching approach to word alignment. In *EMNLP*.

B. Taskar. 2004. *Learning Structured Prediction Models: A Large Margin Approach*. Ph.D. thesis, Stanford University.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING 16*, pages 836–841.