Model-Based Aligner Combination Using Dual Decomposition

John DeNero
Google Research
denero@google.com

Klaus Macherey
Google Research
kmach@google.com

Abstract

Unsupervised word alignment is most often modeled as a Markov process that generates a sentence $\boldsymbol{f}$ conditioned on its translation $\boldsymbol{e}$. A similar model generating $\boldsymbol{e}$ from $\boldsymbol{f}$ will make different alignment predictions. Statistical machine translation systems combine the predictions of two directional models, typically using heuristic combination procedures like *grow-diag-final*. This paper presents a graphical model that embeds two directional aligners into a single model. Inference can be performed via dual decomposition, which reuses the efficient inference algorithms of the directional models. Our bidirectional model enforces a one-to-one phrase constraint while accounting for the uncertainty in the underlying directional models. The resulting alignments improve upon baseline combination heuristics in word-level and phrase-level evaluations.

1 Introduction

Word alignment is the task of identifying corresponding words in sentence pairs. The standard approach to word alignment employs directional Markov models that align the words of a sentence $\boldsymbol{f}$ to those of its translation $\boldsymbol{e}$, such as IBM Model 4 (Brown et al., 1993) or the HMM-based alignment model (Vogel et al., 1996).

Machine translation systems typically combine the predictions of two directional models, one which aligns $\boldsymbol{f}$ to $\boldsymbol{e}$ and the other $\boldsymbol{e}$ to $\boldsymbol{f}$ (Och et al., 1999). Combination can reduce errors and relax the one-to-many structural restriction of directional models. Common combination methods include the union or intersection of directional alignments, as well as heuristic interpolations between the union and intersection like *grow-diag-final* (Koehn et al., 2003). This paper presents a model-based alternative to aligner combination. Inference in a probabilistic model resolves the conflicting predictions of two directional models, while taking into account each model's uncertainty over its output.

This result is achieved by embedding two directional HMM-based alignment models into a larger bidirectional graphical model. The full model structure and potentials allow the two embedded directional models to disagree to some extent, but reward agreement. Moreover, the bidirectional model enforces a one-to-one phrase alignment structure, similar to the output of phrase alignment models (Marcu and Wong, 2002; DeNero et al., 2008), unsupervised inversion transduction grammar (ITG) models (Blunsom et al., 2009), and supervised ITG models (Haghighi et al., 2009; DeNero and Klein, 2010).

Inference in our combined model is not tractable because of numerous edge cycles in the model graph. However, we can employ dual decomposition as an approximate inference technique (Rush et al., 2010). In this approach, we iteratively apply the same efficient sequence algorithms for the underlying directional models, and thereby optimize a dual bound on the model objective. In cases where our algorithm converges, we have a certificate of optimality under the full model. Early stopping before convergence still yields useful outputs.

Our model-based approach to aligner combination yields improvements in alignment quality and phrase extraction quality in Chinese-English experiments, relative to typical heuristic combinations methods applied to the predictions of independent directional models.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 420–429,
Portland, Oregon, June 19-24, 2011.

2 Model Definition

Our bidirectional model $\mathcal{G} = (\mathcal{V}, \mathcal{D})$ is a globally normalized, undirected graphical model of the word alignment for a fixed sentence pair $(\boldsymbol{e}, \boldsymbol{f})$. Each vertex in the vertex set $\mathcal{V}$ corresponds to a model variable V_i, and each undirected edge in the edge set $\mathcal{D}$ corresponds to a pair of variables (V_i, V_j). Each vertex has an associated potential function $\omega_i(v_i)$ that assigns a real-valued potential to each possible value v_i of V_i.[1] Likewise, each edge has an associated potential function $\mu_{ij}(v_i, v_j)$ that scores pairs of values. The probability under the model of any full assignment $\boldsymbol{v}$ to the model variables, indexed by $\mathcal{V}$, factors over vertex and edge potentials.

$$\mathrm{P}(\boldsymbol{v}) \propto \prod_{v_i \in \mathcal{V}} \omega_i(v_i) \cdot \prod_{(v_i, v_j) \in \mathcal{D}} \mu_{ij}(v_i, v_j)$$

Our model contains two directional hidden Markov alignment models, which we review in Section 2.1, along with additional structure that that we introduce in Section 2.2.

2.1 HMM-Based Alignment Model

This section describes the classic hidden Markov model (HMM) based alignment model (Vogel et al., 1996). The model generates a sequence of words $\boldsymbol{f}$ conditioned on a word sequence $\boldsymbol{e}$. We conventionally index the words of $\boldsymbol{e}$ by i and $\boldsymbol{f}$ by j. $\mathrm{P}(\boldsymbol{f}|\boldsymbol{e})$ is defined in terms of a latent alignment vector $\mathbf{a}$, where $a_j = i$ indicates that word position i of $\boldsymbol{e}$ aligns to word position j of $\boldsymbol{f}$.

$$\mathrm{P}(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\mathbf{a}} \mathrm{P}(\boldsymbol{f}, \mathbf{a}|\boldsymbol{e})$$

$$\mathrm{P}(\boldsymbol{f}, \mathbf{a}|\boldsymbol{e}) = \prod_{j=1}^{|\boldsymbol{f}|} \mathrm{D}(a_j | a_{j-1}) \mathrm{M}(f_j | e_{a_j}) \, . \quad (1)$$

In Equation 1 above, the emission model M is a learned multinomial distribution over word types. The transition model D is a multinomial over transition distances, which treats null alignments as a special case.

$$\mathrm{D}(a_j = 0 | a_{j-1} = i) = \mathrm{p}_o$$
$$\mathrm{D}(a_j = i' \neq 0 | a_{j-1} = i) = (1 - \mathrm{p}_o) \cdot c(i' - i) \, ,$$

where $c(i' - i)$ is a learned distribution over signed distances, normalized over the possible transitions from i. The parameters of the conditional multinomial M and the transition model c can be learned from a sentence aligned corpus via the expectation maximization algorithm. The null parameter p_o is typically fixed.[2]

The highest probability word alignment vector under the model for a given sentence pair $(\boldsymbol{e}, \boldsymbol{f})$ can be computed exactly using the standard Viterbi algorithm for HMMs in $O(|\boldsymbol{e}|^2 \cdot |\boldsymbol{f}|)$ time.

An alignment vector $\mathbf{a}$ can be converted trivially into a set of word alignment links $\mathcal{A}$:

$$\mathcal{A}_{\mathbf{a}} = \{(i, j) : a_j = i, i \neq 0\} \, .$$

$\mathcal{A}_{\mathbf{a}}$ is constrained to be many-to-one from $\boldsymbol{f}$ to $\boldsymbol{e}$; many positions j can align to the same i, but each j appears at most once.

We have defined a directional model that generates $\boldsymbol{f}$ from $\boldsymbol{e}$. An identically structured model can be defined that generates $\boldsymbol{e}$ from $\boldsymbol{f}$. Let $\mathbf{b}$ be a vector of alignments where $b_i = j$ indicates that word position j of $\boldsymbol{f}$ aligns to word position i of $\boldsymbol{e}$. Then, $\mathrm{P}(\boldsymbol{e}, \mathbf{b}|\boldsymbol{f})$ is defined similarly to Equation 1, but with $\boldsymbol{e}$ and $\boldsymbol{f}$ swapped. We can distinguish the transition and emission distributions of the two models by subscripting them with their generative direction.

$$\mathrm{P}(\boldsymbol{e}, \mathbf{b}|\boldsymbol{f}) = \prod_{j=1}^{|\boldsymbol{e}|} \mathrm{D}_{\boldsymbol{f} \to \boldsymbol{e}}(b_i | b_{i-1}) \mathrm{M}_{\boldsymbol{f} \to \boldsymbol{e}}(e_i | f_{b_i}) \, .$$

The vector $\mathbf{b}$ can be interpreted as a set of alignment links that is one-to-many: each value i appears at most once in the set.

$$\mathcal{A}_{\mathbf{b}} = \{(i, j) : b_i = j, j \neq 0\} \, .$$

2.2 A Bidirectional Alignment Model

We can combine two HMM-based directional alignment models by embedding them in a larger model

[1] Potentials in an undirected model play the same role as conditional probabilities in a directed model, but do not need to be locally normalized.

[2] In experiments, we set $\mathrm{p}_o = 10^{-6}$. Transitions from a null-aligned state $a_{j-1} = 0$ are also drawn from a fixed distribution, where $\mathrm{D}(a_j = 0 | a_{j-1} = 0) = 10^{-4}$ and for $i' \geq 1$,

$$\mathrm{D}(a_j = i' | a_{j-1} = 0) \propto 0.8^{\left(-\left|i' \cdot \frac{|\boldsymbol{f}|}{|\boldsymbol{e}|} - j\right|\right)} \, .$$

With small p_o, the shape of this distribution has little effect on the alignment outcome.

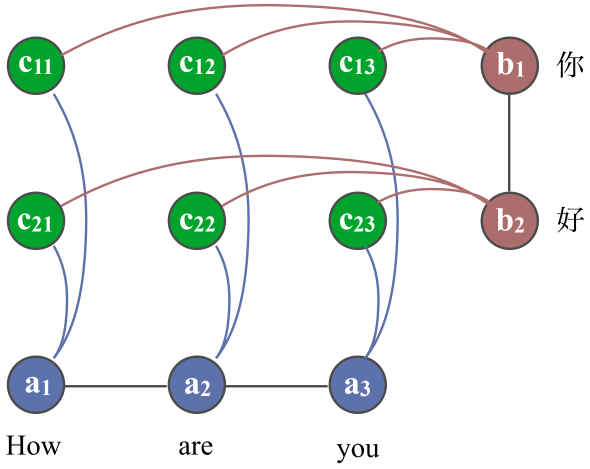

Figure 1: The structure of our graphical model for a simple sentence pair. The variables **a** are blue, **b** are red, and **c** are green.

that includes all of the random variables of two directional models, along with additional structure that promotes agreement and resolves discrepancies.

The original directional models include observed word sequences $\boldsymbol{e}$ and $\boldsymbol{f}$, along with the two latent alignment vectors **a** and **b** defined in Section 2.1. Because the word types and lengths of $\boldsymbol{e}$ and $\boldsymbol{f}$ are always fixed by the observed sentence pair, we can define our model only over **a** and **b**, where the edge potentials between any a_j, f_j, and $\boldsymbol{e}$ are compiled into a vertex potential function $\omega_j^{(\mathbf{a})}$ on a_j, defined in terms of $\boldsymbol{f}$ and $\boldsymbol{e}$, and likewise for any b_i.

$$\omega_j^{(\mathbf{a})}(i) = \mathrm{M}_{\boldsymbol{e}\to\boldsymbol{f}}(f_j|e_i)$$
$$\omega_i^{(\mathbf{b})}(j) = \mathrm{M}_{\boldsymbol{f}\to\boldsymbol{e}}(e_i|f_j)$$

The edge potentials between **a** and **b** encode the transition model in Equation 1.

$$\mu_{j-1,j}^{(\mathbf{a})}(i,i') = \mathrm{D}_{\boldsymbol{e}\to\boldsymbol{f}}(a_j = i'|a_{j-1} = i)$$
$$\mu_{i-1,i}^{(\mathbf{b})}(j,j') = \mathrm{D}_{\boldsymbol{f}\to\boldsymbol{e}}(b_i = j'|b_{i-1} = j)$$

In addition, we include in our model a latent boolean matrix **c** that encodes the output of the combined aligners:

$$\mathbf{c} \in \{0,1\}^{|\boldsymbol{e}|\times|\boldsymbol{f}|} \; .$$

This matrix encodes the alignment links proposed by the bidirectional model:

$$\mathcal{A}_{\mathbf{c}} = \{(i,j) : c_{ij} = 1\} \; .$$

Each model node for an element $c_{ij} \in \{0,1\}$ is connected to a_j and b_i via *coherence edges*. These edges allow the model to ensure that the three sets of variables, **a**, **b**, and **c**, together encode a coherent alignment analysis of the sentence pair. Figure 1 depicts the graph structure of the model.

2.3 Coherence Potentials

The potentials on coherence edges are not learned and do not express any patterns in the data. Instead, they are fixed functions that promote consistency between the integer-valued directional alignment vectors **a** and **b** and the boolean-valued matrix **c**.

Consider the assignment $a_j = i$, where $i = 0$ indicates that word f_j is null-aligned, and $i \geq 1$ indicates that f_j aligns to e_i. The coherence potential ensures the following relationship between the variable assignment $a_j = i$ and the variables $c_{i'j}$, for any $i' \in [1, |\boldsymbol{e}|]$.

- If $i = 0$ (null-aligned), then all $c_{i'j} = 0$.
- If $i > 0$, then $c_{ij} = 1$.
- $c_{i'j} = 1$ only if $i' \in \{i-1,\; i,\; i+1\}$.
- Assigning $c_{i'j} = 1$ for $i' \neq i$ incurs a cost $e^{-\alpha}$.

Collectively, the list of cases above enforce an intuitive correspondence: an alignment $a_j = i$ ensures that c_{ij} must be 1, adjacent neighbors may be 1 but incur a cost, and all other elements are 0.

This pattern of effects can be encoded in a potential function $\mu^{(\mathbf{c})}$ for each coherence edge. These edge potential functions takes an integer value i for some variable a_j and a binary value k for some $c_{i'j}$.

$$\mu^{(\mathbf{c})}_{(a_j, c_{i'j})}(i,k) = \left\{ \begin{array}{ll} 1 & i = 0 \wedge k = 0 \\ 0 & i = 0 \wedge k = 1 \\ \hline 1 & i = i' \wedge k = 1 \\ 0 & i = i' \wedge k = 0 \\ \hline 1 & i \neq i' \wedge k = 0 \\ e^{-\alpha} & |i - i'| = 1 \wedge k = 1 \\ 0 & |i - i'| > 1 \wedge k = 1 \end{array} \right. \quad (2)$$

Above, potentials of 0 effectively disallow certain cases because a full assignment to $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ is scored by the product of all model potentials. The potential function $\mu^{(\mathbf{c})}_{(b_i, c_{ij'})}(j,k)$ for a coherence edge between **b** and **c** is defined similarly.

2.4 Model Properties

We interpret $\mathbf{c}$ as the final alignment produced by the model, ignoring $\mathbf{a}$ and $\mathbf{b}$. In this way, we relax the one-to-many constraints of the directional models. However, all of the information about how words align is expressed by the vertex and edge potentials on $\mathbf{a}$ and $\mathbf{b}$. The coherence edges and the link matrix $\mathbf{c}$ only serve to resolve conflicts between the directional models and communicate information between them.

Because directional alignments are preserved intact as components of our model, extensions or refinements to the underlying directional Markov alignment model could be integrated cleanly into our model as well, including lexicalized transition models (He, 2007), extended conditioning contexts (Brunning et al., 2009), and external information (Shindo et al., 2010).

For any assignment to $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ with non-zero probability, $\mathbf{c}$ must encode a one-to-one phrase alignment with a maximum phrase length of 3. That is, any word in either sentence can align to at most three words in the opposite sentence, and those words must be contiguous. This restriction is directly enforced by the edge potential in Equation 2.

3 Model Inference

In general, graphical models admit efficient, exact inference algorithms if they do not contain cycles. Unfortunately, our model contains numerous cycles. For every pair of indices (i, j) and (i', j'), the following cycle exists in the graph:

$$\begin{aligned} &c_{ij} \rightarrow b_i \rightarrow c_{ij'} \rightarrow a_{j'} \rightarrow \\ &\quad c_{i'j'} \rightarrow b_{i'} \rightarrow c_{i'j} \rightarrow a_j \rightarrow c_{ij} \end{aligned}$$

Additional cycles also exist in the graph through the edges between a_{j-1} and a_j and between b_{i-1} and b_i. The general phrase alignment problem under an arbitrary model is known to be NP-hard (DeNero and Klein, 2008).

3.1 Dual Decomposition

While the entire graphical model has loops, there are two overlapping subgraphs that are cycle-free. One subgraph $\mathcal{G}_\mathbf{a}$ includes all of the vertices corresponding to variables $\mathbf{a}$ and $\mathbf{c}$. The other subgraph $\mathcal{G}_\mathbf{b}$ includes vertices for variables $\mathbf{b}$ and $\mathbf{c}$. Every edge in the graph belongs to exactly one of these two subgraphs.

The dual decomposition inference approach allows us to exploit this sub-graph structure (Rush et al., 2010). In particular, we can iteratively apply exact inference to the subgraph problems, adjusting their potentials to reflect the constraints of the full problem. The technique of dual decomposition has recently been shown to yield state-of-the-art performance in dependency parsing (Koo et al., 2010).

3.2 Dual Problem Formulation

To describe a dual decomposition inference procedure for our model, we first restate the inference problem under our graphical model in terms of the two overlapping subgraphs that admit tractable inference. Let $\mathbf{c}^{(\mathbf{a})}$ be a copy of $\mathbf{c}$ associated with $\mathcal{G}_\mathbf{a}$, and $\mathbf{c}^{(\mathbf{b})}$ with $\mathcal{G}_\mathbf{b}$. Also, let $f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})})$ be the unnormalized log-probability of an assignment to $\mathcal{G}_\mathbf{a}$ and $g(\mathbf{b}, \mathbf{c}^{(\mathbf{b})})$ be the unnormalized log-probability of an assignment to $\mathcal{G}_\mathbf{b}$. Finally, let $\mathcal{I}$ be the index set of all (i, j) for $\mathbf{c}$. Then, the maximum likelihood assignment to our original model can be found by optimizing

$$\max_{\mathbf{a},\mathbf{b},\mathbf{c}^{(\mathbf{a})},\mathbf{c}^{(\mathbf{b})}} f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})}) + g(\mathbf{b}, \mathbf{c}^{(\mathbf{b})}) \tag{3}$$

$$\textit{such that: } c_{ij}^{(\mathbf{a})} = c_{ij}^{(\mathbf{b})} \;\; \forall\, (i, j) \in \mathcal{I} .$$

The Lagrangian relaxation of this optimization problem is $L(\mathbf{a}, \mathbf{b}, \mathbf{c}^{(\mathbf{a})}, \mathbf{c}^{(\mathbf{b})}, \mathbf{u}) =$

$$f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})}) + g(\mathbf{b}, \mathbf{c}^{(\mathbf{b})}) + \sum_{(i,j)\in\mathcal{I}} u(i, j)(\mathbf{c}_{i,j}^{(\mathbf{a})} - \mathbf{c}_{i,j}^{(\mathbf{b})}) .$$

Hence, we can rewrite the original problem as

$$\max_{\mathbf{a},\mathbf{b},\mathbf{c}^{(\mathbf{a})},\mathbf{c}^{(\mathbf{b})}} \min_{\mathbf{u}} L(\mathbf{a}, \mathbf{b}, \mathbf{c}^{(\mathbf{a})}, \mathbf{c}^{(\mathbf{b})}, \mathbf{u}) .$$

We can form a dual problem that is an upper bound on the original optimization problem by swapping the order of min and max. In this case, the dual problem decomposes into two terms that are each local to an acyclic subgraph.

$$\begin{aligned} \min_{\mathbf{u}} \Bigg(&\max_{\mathbf{a},\mathbf{c}^{(\mathbf{a})}} \Bigg[f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})}) + \sum_{i,j} u(i, j) c_{ij}^{(\mathbf{a})} \Bigg] \\ &+ \max_{\mathbf{b},\mathbf{c}^{(\mathbf{b})}} \Bigg[g(\mathbf{b}, \mathbf{c}^{(\mathbf{b})}) - \sum_{i,j} u(i, j) c_{ij}^{(\mathbf{b})} \Bigg] \Bigg) \end{aligned} \tag{4}$$

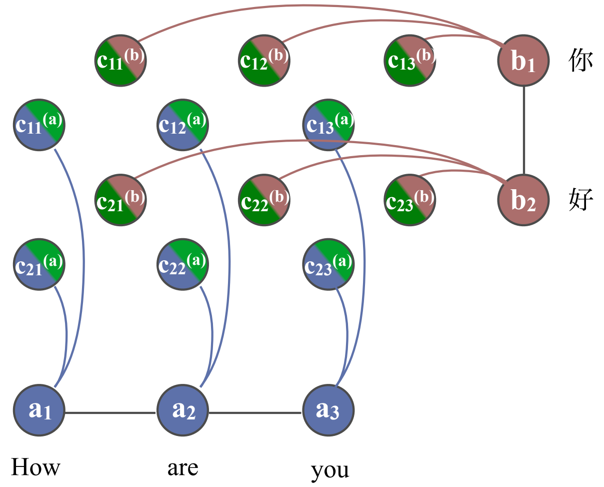

Figure 2: Our combined model decomposes into two acyclic models that each contain a copy of $\mathbf{c}$.

The decomposed model is depicted in Figure 2. As in previous work, we solve for the dual variable $\mathbf{u}$ by repeatedly performing inference in the two decoupled maximization problems.

3.3 Sub-Graph Inference

We now address the problem of evaluating Equation 4 for fixed $\mathbf{u}$. Consider the first line of Equation 4, which includes variables $\mathbf{a}$ and $\mathbf{c}^{(\mathbf{a})}$.

$$\max_{\mathbf{a},\mathbf{c}^{(\mathbf{a})}} \left[f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})}) + \sum_{i,j} u(i,j) c_{ij}^{(\mathbf{a})} \right] \tag{5}$$

Because the graph $\mathcal{G}_{\mathbf{a}}$ is tree-structured, Equation 5 can be evaluated in polynomial time. In fact, we can make a stronger claim: we can reuse the Viterbi inference algorithm for linear chain graphical models that applies to the embedded directional HMM models. That is, we can cast the optimization of Equation 5 as

$$\max_{\mathbf{a}} \left[\prod_{j=1}^{|\boldsymbol{f}|} \mathrm{D}_{\boldsymbol{e}\to\boldsymbol{f}}(a_j|a_{j-1}) \cdot \mathrm{M}'_j(a_j = i) \right] .$$

In the original HMM-based aligner, the vertex potentials correspond to bilexical probabilities. Those quantities appear in $f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})})$, and therefore will be a part of $\mathrm{M}'_j(\cdot)$ above. The additional terms of Equation 5 can also be factored into the vertex potentials of this linear chain model, because the optimal choice of each c_{ij} can be determined from a_j and the model parameters. If $a_j = i$, then $c_{ij} = 1$ according to our edge potential defined in Equation 2. Hence, setting $a_j = i$ requires the inclusion of the corresponding vertex potential $\omega_j^{(\mathbf{a})}(i)$, as well as $u(i,j)$. For $i' \neq i$, either $c_{i'j} = 0$, which contributes nothing to Equation 5, or $c_{i'j} = 1$, which contributes $u(i',j) - \alpha$, according to our edge potential between a_j and $c_{i'j}$.

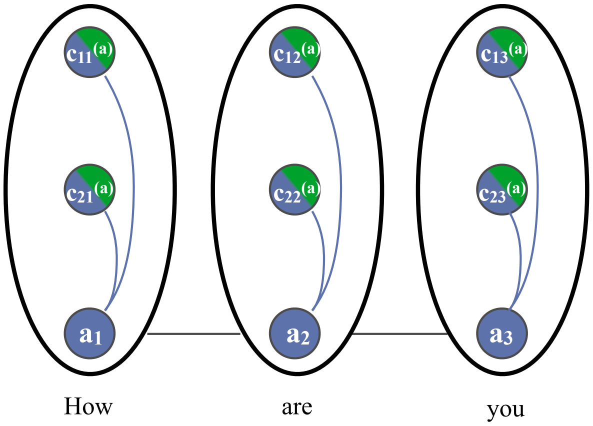

Figure 3: The tree-structured subgraph $\mathcal{G}_{\mathbf{a}}$ can be mapped to an equivalent chain-structured model by optimizing over $c_{i'j}$ for $a_j = i$.

Thus, we can capture the net effect of assigning a_j and then optimally assigning all $c_{i'j}$ in a single potential $\mathrm{M}'_j(a_j = i) =$

$$\omega_j^{(\mathbf{a})}(i) + \exp\left[u(i,j) + \sum_{j':|j'-j|=1} \max(0, u(i,j') - \alpha) \right]$$

Note that Equation 5 and f are sums of terms in log space, while Viterbi inference for linear chains assumes a product of terms in probability space, which introduces the exponentiation above.

Defining this potential allows us to collapse the source-side sub-graph inference problem defined by Equation 5, into a simple linear chain model that only includes potential functions M'_j and $\mu^{(\mathbf{a})}$. Hence, we can use a highly optimized linear chain inference implementation rather than a solver for general tree-structured graphical models. Figure 3 depicts this transformation.

An equivalent approach allows us to evaluate the

Algorithm 1 Dual decomposition inference algorithm for the bidirectional model

```
for $t = 1$ to max iterations do
    $r \leftarrow \frac{1}{t}$                                                        ▷ Learning rate
    $\mathbf{c}^{(\mathbf{a})} \leftarrow \arg\max f(\mathbf{a}, \mathbf{c}^{(\mathbf{a})}) + \sum_{i,j} u(i,j) c^{(\mathbf{a})}_{ij}$
    $\mathbf{c}^{(\mathbf{b})} \leftarrow \arg\max g(\mathbf{b}, \mathbf{c}^{(\mathbf{b})}) - \sum_{i,j} u(i,j) c^{(\mathbf{b})}_{ij}$
    if $\mathbf{c}^{(\mathbf{a})} = \mathbf{c}^{(\mathbf{b})}$ then
        return $\mathbf{c}^{(\mathbf{a})}$                                            ▷ Converged
    $\mathbf{u} \leftarrow \mathbf{u} + r \cdot (\mathbf{c}^{(\mathbf{b})} - \mathbf{c}^{(\mathbf{a})})$              ▷ Dual update
return combine($\mathbf{c}^{(\mathbf{a})}, \mathbf{c}^{(\mathbf{b})}$)                      ▷ Stop early
```

second line of Equation 4 for fixed $\mathbf{u}$:

$$\max_{\mathbf{b}, \mathbf{c}^{(\mathbf{b})}} \left[g(\mathbf{b}, \mathbf{c}^{(\mathbf{b})}) + \sum_{i,j} u(i,j) c^{(\mathbf{b})}_{ij} \right] . \quad (6)$$

3.4 Dual Decomposition Algorithm

Now that we have the means to efficiently evaluate Equation 4 for fixed $\mathbf{u}$, we can define the full dual decomposition algorithm for our model, which searches for a $\mathbf{u}$ that optimizes Equation 4. We can iteratively search for such a $\mathbf{u}$ via sub-gradient descent. We use a learning rate $\frac{1}{t}$ that decays with the number of iterations t. The full dual decomposition optimization procedure appears in Algorithm 1.

If Algorithm 1 converges, then we have found a $\mathbf{u}$ such that the value of $\mathbf{c}^{(\mathbf{a})}$ that optimizes Equation 5 is identical to the value of $\mathbf{c}^{(\mathbf{b})}$ that optimizes Equation 6. Hence, it is also a solution to our original optimization problem: Equation 3. Since the dual problem is an upper bound on the original problem, this solution must be optimal for Equation 3.

3.5 Convergence and Early Stopping

Our dual decomposition algorithm provides an inference method that is exact upon convergence.[3] When Algorithm 1 does not converge, the two alignments $\mathbf{c}^{(\mathbf{a})}$ and $\mathbf{c}^{(\mathbf{b})}$ can still be used. While these alignments may differ, they will likely be more similar than the alignments of independent aligners.

These alignments will still need to be combined procedurally (e.g., taking their union), but because they are more similar, the importance of the combination procedure is reduced. We analyze the behavior of early stopping experimentally in Section 5.

3.6 Inference Properties

Because we set a maximum number of iterations n in the dual decomposition algorithm, and each iteration only involves optimization in a sequence model, our entire inference procedure is only a constant multiple n more computationally expensive than evaluating the original directional aligners.

Moreover, the value of $\mathbf{u}$ is specific to a sentence pair. Therefore, our approach does not require any additional communication overhead relative to the independent directional models in a distributed aligner implementation. Memory requirements are virtually identical to the baseline: only $\mathbf{u}$ must be stored for each sentence pair as it is being processed, but can then be immediately discarded once alignments are inferred.

Other approaches to generating one-to-one phrase alignments are generally more expensive. In particular, an ITG model requires $O(|\boldsymbol{e}|^3 \cdot |\boldsymbol{f}|^3)$ time, whereas our algorithm requires only

$$O(n \cdot (|\boldsymbol{f}||\boldsymbol{e}|^2 + |\boldsymbol{e}||\boldsymbol{f}|^2)) .$$

Moreover, our approach allows Markov distortion potentials, while standard ITG models are restricted to only hierarchical distortion.

4 Related Work

Alignment combination normally involves selecting some $\mathcal{A}$ from the output of two directional models. Common approaches include forming the union or intersection of the directional sets.

$$\mathcal{A}_{\cup} = \mathcal{A}_{\mathbf{a}} \cup \mathcal{A}_{\mathbf{b}}$$
$$\mathcal{A}_{\cap} = \mathcal{A}_{\mathbf{a}} \cap \mathcal{A}_{\mathbf{b}} .$$

More complex combiners, such as the *grow-diag-final* heuristic (Koehn et al., 2003), produce alignment link sets that include all of $\mathcal{A}_{\cap}$ and some subset of $\mathcal{A}_{\cup}$ based on the relationship of multiple links (Och et al., 1999).

In addition, supervised word alignment models often use the output of directional unsupervised aligners as features or pruning signals. In the case

[3] This certificate of optimality is not provided by other approximate inference algorithms, such as belief propagation, sampling, or simulated annealing.

that a supervised model is restricted to proposing alignment links that appear in the output of a directional aligner, these models can be interpreted as a combination technique (Deng and Zhou, 2009). Such a model-based approach differs from ours in that it requires a supervised dataset and treats the directional aligners' output as fixed.

Combination is also related to agreement-based learning (Liang et al., 2006). This approach to jointly learning two directional alignment models yields state-of-the-art unsupervised performance. Our method is complementary to agreement-based learning, as it applies to Viterbi inference under the model rather than computing expectations. In fact, we employ agreement-based training to estimate the parameters of the directional aligners in our experiments.

A parallel idea that closely relates to our bidirectional model is posterior regularization, which has also been applied to the word alignment problem (Graça et al., 2008). One form of posterior regularization stipulates that the posterior probability of alignments from two models must agree, and enforces this agreement through an iterative procedure similar to Algorithm 1. This approach also yields state-of-the-art unsupervised alignment performance on some datasets, along with improvements in end-to-end translation quality (Ganchev et al., 2008).

Our method differs from this posterior regularization work in two ways. First, we iterate over Viterbi predictions rather than posteriors. More importantly, we have changed the output space of the model to be a one-to-one phrase alignment via the coherence edge potential functions.

Another similar line of work applies belief propagation to factor graphs that enforce a one-to-one word alignment (Cromières and Kurohashi, 2009). The details of our models differ: we employ distance-based distortion, while they add structural correspondence terms based on syntactic parse trees. Also, our model training is identical to the HMM-based baseline training, while they employ belief propagation for both training and Viterbi inference. Although differing in both model and inference, our work and theirs both find improvements from defining graphical models for alignment that do not admit exact polynomial-time inference algorithms.

Aligner Model	Intersection $\|\mathcal{A}_\cap\|$	Union $\|\mathcal{A}_\cup\|$	Agreement $\|\mathcal{A}_\cap\|/\|\mathcal{A}_\cup\|$
Baseline	5,554	10,998	50.5%
Bidirectional	7,620	10,262	74.3%

Table 1: The bidirectional model's dual decomposition algorithm substantially increases the overlap between the predictions of the directional models, measured by the number of links in their intersection.

5 Experimental Results

We evaluated our bidirectional model by comparing its output to the annotations of a hand-aligned corpus. In this way, we can show that the bidirectional model improves alignment quality and enables the extraction of more correct phrase pairs.

5.1 Data Conditions

We evaluated alignment quality on a hand-aligned portion of the NIST 2002 Chinese-English test set (Ayan and Dorr, 2006). We trained the model on a portion of FBIS data that has been used previously for alignment model evaluation (Ayan and Dorr, 2006; Haghighi et al., 2009; DeNero and Klein, 2010). We conducted our evaluation on the first 150 sentences of the dataset, following previous work. This portion of the dataset is commonly used to train supervised models.

We trained the parameters of the directional models using the agreement training variant of the expectation maximization algorithm (Liang et al., 2006). Agreement-trained IBM Model 1 was used to initialize the parameters of the HMM-based alignment models (Brown et al., 1993). Both IBM Model 1 and the HMM alignment models were trained for 5 iterations on a 6.2 million word parallel corpus of FBIS newswire. This training regimen on this data set has provided state-of-the-art unsupervised results that outperform IBM Model 4 (Haghighi et al., 2009).

5.2 Convergence Analysis

With $n = 250$ maximum iterations, our dual decomposition inference algorithm only converges 6.2% of the time, perhaps largely due to the fact that the two directional models have different one-to-many structural constraints. However, the dual decompo-

Model	Combiner	Prec	Rec	AER
Baseline	union	57.6	80.0	33.4
	intersect	**86.2**	62.7	27.2
	grow-diag	60.1	78.8	32.1
Bidirectional	union	63.3	**81.5**	29.1
	intersect	77.5	75.1	**23.6**
	grow-diag	65.6	80.6	28.0

Table 2: Alignment error rate results for the bidirectional model versus the baseline directional models. "grow-diag" denotes the grow-diag-final heuristic.

Model	Combiner	Prec	Rec	F1
Baseline	union	**75.1**	33.5	46.3
	intersect	64.3	43.4	51.8
	grow-diag	68.3	37.5	48.4
Bidirectional	union	63.2	44.9	52.5
	intersect	57.1	**53.6**	**55.3**
	grow-diag	60.2	47.4	53.0

Table 3: Phrase pair extraction accuracy for phrase pairs up to length 5. "grow-diag" denotes the grow-diag-final heuristic.

sition algorithm does promote agreement between the two models. We can measure the agreement between models as the fraction of alignment links in the union $\mathcal{A}_\cup$ that also appear in the intersection $\mathcal{A}_\cap$ of the two directional models. Table 1 shows a 47% relative increase in the fraction of links that both models agree on by running dual decomposition (bidirectional), relative to independent directional inference (baseline). Improving convergence rates represents an important area of future work.

5.3 Alignment Error Evaluation

To evaluate alignment error of the baseline directional aligners, we must apply a combination procedure such as union or intersection to $\mathcal{A}_\mathbf{a}$ and $\mathcal{A}_\mathbf{b}$. Likewise, in order to evaluate alignment error for our combined model in cases where the inference algorithm does not converge, we must apply combination to $\mathbf{c}^{(\mathbf{a})}$ and $\mathbf{c}^{(\mathbf{b})}$. In cases where the algorithm does converge, $\mathbf{c}^{(\mathbf{a})} = \mathbf{c}^{(\mathbf{b})}$ and so no further combination is necessary.

We evaluate alignments relative to hand-aligned data using two metrics. First, we measure alignment error rate (AER), which compares the proposed alignment set $\mathcal{A}$ to the sure set $\mathcal{S}$ and possible set $\mathcal{P}$ in the annotation, where $\mathcal{S} \subseteq \mathcal{P}$.

$$\text{Prec}(\mathcal{A}, \mathcal{P}) = \frac{|\mathcal{A} \cap \mathcal{P}|}{|\mathcal{A}|}$$

$$\text{Rec}(\mathcal{A}, \mathcal{S}) = \frac{|\mathcal{A} \cap \mathcal{S}|}{|\mathcal{S}|}$$

$$\text{AER}(\mathcal{A}, \mathcal{S}, \mathcal{P}) = 1 - \frac{|\mathcal{A} \cap \mathcal{S}| + |\mathcal{A} \cap \mathcal{P}|}{|\mathcal{A}| + |\mathcal{S}|}$$

AER results for Chinese-English are reported in Table 2. The bidirectional model improves both precision and recall relative to all heuristic combination techniques, including *grow-diag-final* (Koehn et al., 2003). Intersected alignments, which are one-to-one phrase alignments, achieve the best AER.

Second, we measure phrase extraction accuracy. Extraction-based evaluations of alignment better coincide with the role of word aligners in machine translation systems (Ayan and Dorr, 2006). Let $\text{R}_5(\mathcal{S}, \mathcal{P})$ be the set of phrases up to length 5 extracted from the sure link set $\mathcal{S}$ and possible link set $\mathcal{P}$. Possible links are both included and excluded from phrase pairs during extraction, as in DeNero and Klein (2010). Null aligned words are never included in phrase pairs for evaluation. Phrase extraction precision, recall, and F1 for $\text{R}_5(\mathcal{A}, \mathcal{A})$ are reported in Table 3. Correct phrase pair recall increases from 43.4% to 53.6% (a 23.5% relative increase) for the bidirectional model, relative to the best baseline.

Finally, we evaluated our bidirectional model in a large-scale end-to-end phrase-based machine translation system from Chinese to English, based on the alignment template approach (Och and Ney, 2004). The translation model weights were tuned for both the baseline and bidirectional alignments using lattice-based minimum error rate training (Kumar et al., 2009). In both cases, *union* alignments outperformed other combination heuristics. Bidirectional alignments yielded a modest improvement of 0.2% BLEU[4] on a single-reference evaluation set of sentences sampled from the web (Papineni et al., 2002).

[4] BLEU improved from 29.59% to 29.82% after training IBM Model 1 for 3 iterations and training the HMM-based alignment model for 3 iterations. During training, link posteriors were symmetrized by pointwise linear interpolation.

As our model only provides small improvements in alignment precision and recall for the *union* combiner, the magnitude of the BLEU improvement is not surprising.

6 Conclusion

We have presented a graphical model that combines two classical HMM-based alignment models. Our bidirectional model, which requires no additional learning and no supervised data, can be applied using dual decomposition with only a constant factor additional computation relative to independent directional inference. The resulting predictions improve the precision and recall of both alignment links and extraced phrase pairs in Chinese-English experiments. The best results follow from combination via *intersection*.

Because our technique is defined declaratively in terms of a graphical model, it can be extended in a straightforward manner, for instance with additional potentials on **c** or improvements to the component directional models. We also look forward to discovering the best way to take advantage of these new alignments in downstream applications like machine translation, supervised word alignment, bilingual parsing (Burkett et al., 2010), part-of-speech tag induction (Naseem et al., 2009), or cross-lingual model projection (Smith and Eisner, 2009; Das and Petrov, 2011).

References

Necip Fazil Ayan and Bonnie J. Dorr. 2006. Going beyond AER: An extensive analysis of word alignments and their impact on MT. In *Proceedings of the Association for Computational Linguistics*.

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of the Association for Computational Linguistics*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*.

Jamie Brunning, Adria de Gispert, and William Byrne. 2009. Context-dependent alignment models for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

David Burkett, John Blitzer, and Dan Klein. 2010. Joint parsing and alignment with weakly synchronized grammars. In *Proceedings of the North American Association for Computational Linguistics and IJCNLP*.

Fabien Cromières and Sadao Kurohashi. 2009. An alignment algorithm using belief propagation and a structure-based distortion model. In *Proceedings of the European Chapter of the Association for Computational Linguistics and IJCNLP*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the Association for Computational Linguistics*.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of the Association for Computational Linguistics*.

John DeNero and Dan Klein. 2010. Discriminative modeling of extraction sets for machine translation. In *Proceedings of the Association for Computational Linguistics*.

John DeNero, Alexandre Bouchard-Côté, and Dan Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yonggang Deng and Bowen Zhou. 2009. Optimizing word alignment combination for phrase table training. In *Proceedings of the Association for Computational Linguistics*.

Kuzman Ganchev, Joao Graça, and Ben Taskar. 2008. Better alignments = better translations? In *Proceedings of the Association for Computational Linguistics*.

Joao Graça, Kuzman Ganchev, and Ben Taskar. 2008. Expectation maximization and posterior constraints. In *Proceedings of Neural Information Processing Systems*.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised ITG models. In *Proceedings of the Association for Computational Linguistics*.

Xiaodong He. 2007. Using word-dependent transition models in HMM-based word alignment for statistical machine. In *ACL Workshop on Statistical Machine Translation*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Terry Koo, Alexander M. Rush, Michael Collins, Tommi Jaakkola, and David Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Shankar Kumar, Wolfgang Macherey, Chris Dyer, and Franz Josef Och. 2009. Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Association for Computational Linguistics*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

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

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*.

Franz Josef Och, Christopher Tillman, and Hermann Ney. 1999. Improved alignment models for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*.

Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hiroyuki Shindo, Akinori Fujino, and Masaaki Nagata. 2010. Word alignment with synonym regularization. In *Proceedings of the Association for Computational Linguistics*.

David A. Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the Conference on Computational linguistics*.