Predicate Argument Alignment using a Global Coherence Model

Travis Wolfe, Mark Dredze, and Benjamin Van Durme
Johns Hopkins University
Baltimore, MD, USA

Abstract

We present a joint model for predicate argument alignment. We leverage multiple sources of semantic information, including temporal ordering constraints between events. These are combined in a max-margin framework to find a globally consistent view of entities and events across multiple documents, which leads to improvements over a very strong local baseline.

1 Introduction

Natural language understanding (NLU) requires analysis beyond the sentence-level. For example, an *entity* may be mentioned multiple times in a discourse, participating in various events, where each event may itself be referenced elsewhere in the text. Traditionally the task of *coreference resolution* has been defined as finding those entity mentions within a single document that co-refer, while *cross-document coreference resolution* considers a wider discourse context across many documents, yet still pertains strictly to entities.

Predicate argument alignment, or entity-event cross-document coreference resolution, enlarges the set of possible co-referent elements to include the mentions of situations in which entities participate. This expanded definition drives practitioners towards a more complete model of NLU, where systems must not only consider who is mentioned, but also what happened. However, despite the drive towards an expanded notion of discourse, models typically are formulated with strong notions of local-independence: viewing a multi-document task as one limited to individual pairs of sentences. This creates a mis-match between the goals of such work – considering entire documents – with the systems – consider individual sentences.

In this work, we consider a system that takes a document level view in considering coreference for entities and predictions: the task of predicate argument linking. We treat this task as a global inference problem, leveraging multiple sources of semantic information identified at the document level. Global inference for this problem is mostly unexplored, with the exception of Lee et al. (2012) (discussed in § 8). Especially novel here is the use of document-level temporal constraints on events, representing a next step forward on the path to full understanding.

Our approach avoids the pitfalls of local inference while still remaining fast and exact. We use the pairwise features of a very strong predicate argument aligner (Wolfe et al., 2013) (competitive with the state-of-the-art (Roth, 2014)), and add quadratic factors that constrain local decisions based on global document information. These global factors lead to superior performance compared to the previous state-of-the-art. We release both our code and data.[1]

2 Model

Consider the two sentences from the document pair shown in Figure 1. These sentences describe the same event, although with different details. The source sentence has four predicates and four arguments, while the target has three predicates and three arguments. In this case, one of the predicates from each sentence aligns, as do three of the arguments. We also show additional information potentially helpful to determining alignments: temporal relations between the predicates. The goal of predicate argument alignment is to assign these links indicating coreferent predicates and arguments across a document pair (Roth and Frank, 2012).

Previous work by Wolfe et al. (2013) formulated

[1] `https://github.com/hltcoe/parma2`

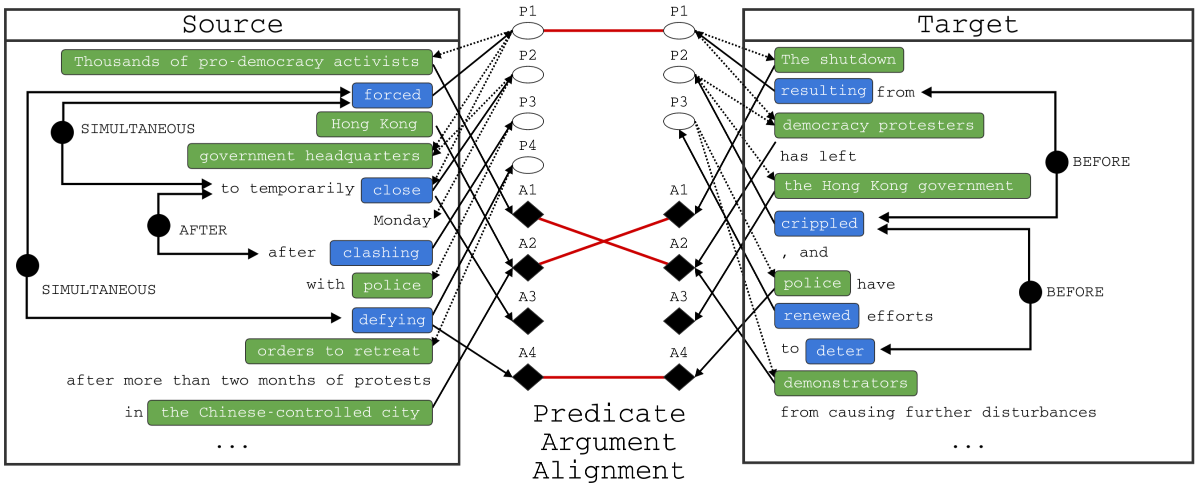

Figure 1: An example analysis and predicate argument alignment task between a source and target document. Predicates appear as hollow ovals, have blue mentions, and are aligned considering their arguments (dashed lines). Arguments, in black diamonds with green mentions, represent a document-level entity (coreference chain), and are aligned using their predicate structure and mention-level features. The alignment choices appear in the middle in red. Temporal relation information is lifted into the global inference over alignments.

this as a binary classification problem: given a pair of arguments or predicates, construct features and score the pair, where scores above threshold indicate links. A binary classification framework has advantages: it's fast since individual decisions can be made quickly, but it comes at the cost of global information across links. The result may be links that conflict in their interpretation of the document. Figure 1 makes clear that jointly considering all links at once can aid individual decisions, for example, by including temporal ordering of predicates.

The global nature of this task is similar to word alignment for machine translation (MT). Many systems consider alignment links between words individually, selecting the best link for each word independently of the other words in the sentence. Just as with an independent linking strategy in predicate argument alignment, this can lead to inconsistencies in the output. Lacoste-Julien et al. (2006) introduced a model that jointly resolved word alignments based on the introduction of quadratic variables, factors that depend on two alignment decisions which characterize patterns that span word-word links. Their approach achieved improved results even in the presence of little training data.

We present a global predicate argument alignment model based on considering quadratic interactions between alignment variables to captures patterns we expect in coherent discourse. We introduce factors which are comprised of a binary variable, multiple quadratic constraints on that variable, and features that determine the cost associated with that variable in order to characterize the dependence between alignment decisions.

While the mathematical framework we use is similar to Lacoste-Julien et al. (2006), predicate argument alignment greatly differs from word alignment; thus our joint factors are based on different sources of regularity. Word alignment favors monotonicity in word order, but this effect is very weak in predicate argument alignment: aligned items can be spread throughout a document, and are often nested, gapped, or shuffled. Instead, we encode assumptions about consistency of temporal relations between coreferent events, coherence between predicates and arguments that appear in both documents, and fertility (to prevent over-alignment). We also note that our setting has much less data than typical word alignment tasks, as well as richer features that utilize semantic resources.

Notation An alignment between an item indexed by i in the source document and j in the target document is represented by variable $z_{ij} \in \{0, 1\}$, where $z_{ij} = 1$ indicates that items i and j are aligned. In some cases, we will explicitly indicate when the two items are predicates as z^p_{ij}; an argument alignment will be z^a_{ij}. We represent all alignments for a document pair as matrix $\mathbf{z}$.

For clarity, we omit any variable representing observed data when discussing feature functions; alignment variables are endowed with this information. For each pair of items we use "local" feature functions $\mathbf{f}(\cdot)$ and corresponding parameters w, which capture the similarity between two items without the context of other alignments.

$$s_{ij} = w \cdot \mathbf{f}(z_{ij}) \tag{1}$$

where s_{ij} is the score of linking items i and j.

Using only local features, our system would greedily select alignments. To capture global aspects we add joint factors that capture effects between alignment variables. Each joint factor ϕ is comprised of a constrained binary variable z_ϕ associated with features $\mathbf{f}(\phi)$ that indicates when the factor is active. Together with parameters w these form additional scores s_ϕ for the objective:

$$s_\phi = w \cdot \mathbf{f}(\phi) \tag{2}$$

The full linear scoring function on alignments sums over both local similarity and joint factors:

$$\sum_{ij} s_{ij} z_{ij} + \sum_{\phi \in \Phi} s_\phi z_\phi. \tag{3}$$

Lastly, it is convenient to describe the local feature functions and their corresponding alignment variable as factors with no constraints, and we will do so when describing the full score function.

3 Local Factors

Local factors encode features based on the mention pair, which include a wide variety of similarity measures, e.g. whether two headwords appear as synonyms in WordNet, gender agreement based on possessive pronouns. We adopt the features of Wolfe et al. (2013), a strong baseline system which doesn't use global inference.[2] These features are built on top of a variety of semantic resources (PPDB (Ganitkevitch et al., 2013), WordNet (Miller, 1995), FrameNet (Baker et al., 1998)) and methods for comparing mentions (tree edit distance (Yao et al., 2013), string transducer (Andrews et al., 2012)).

4 Joint Factors

Our goal is to develop joint factors that improve over the feature rich local factors baseline by considering global information.

Fertility A common mistake when making independent classification decisions is to align many source items to a single target item. While each link looks promising on its own, they clearly cannot all be right. Empirically, the training set reveals that many to one alignments are uncommon; thus many to one predictions are likely errors. We add a fertility factor for predicates and arguments, where fertility is defined as the number of links to an item. Higher fertilities are undesired and are thus penalized. Formally, for matrix $\mathbf{z}$, the fertility of a row i or column j is the sum of that row or column. We discuss fertility in terms of rows below.

We include two types of fertility factors. First, factor ϕ_{fert1} distinguishes between rows with at least one link from those with none. For row i, we add one instance of the linear factor ϕ_{fert1} with constraints

$$z_{\phi_{\text{fert1}}} \geq z_{ij} \; \forall j \tag{4}$$

The cost associated with $z_{\phi_{\text{fert1}}}$, which we will refer to as s_{fert1}, will be incurred any time an item is mentioned in both documents. For data sets with many singletons, s_{fert1} more strongly penalizes non-singleton rows, reflecting this pattern in the training data. We make s_{fert1} parametric, where the features of the ϕ_{fert1} factor allow us to learn different weights for predicates and arguments, as well as the size of the row, i.e. number of items in the pairing.

The second fertility factory ϕ_{fert2} considers items with a fertility greater than one, penalizing items for having too many links. Its binary variable has the

[2]Some features inspect the apparent predicate argument structure, based on things like dependency parses, but the model may not inspect more than one of its own decisions (joint factors) while scoring an alignment.

quadratic constraints:

$$z_{\phi_{\text{fert2}}} \geq z_{ij} z_{ik} \ \forall j < k \tag{5}$$

This factor penalizes rows that have fertility of at least two, but does not distinguish beyond that. An alternative would be to introduce a factor for every pair of variables in a row, each with one constraint. This would heavily penalize fertilities greater than two. We found that the resulting quadratic program took longer to solve and gave worse results.

Since documents have been processed to identify in-document coreference chains, we do not expect multiple arguments from a source document to align to a single target item. For this reason, we expect ϕ_{fert2} for arguments to have a large negative weight. In contrast, since predicates do not form chains, we may have multiple source predicates for one target.

We note an important difference between our fertility factor compared with Lacoste-Julien et al. (2006). We parameterize fertility for only two cases (1 and 2) whereas they consider fertility factors from 2 to D. We do not parameterize fertilities higher than two because they are not common in our dataset and come at a high computational cost.

The features $\mathbf{f}(\phi)$ for both ϕ_{fert1} and ϕ_{fert2} are an intercept feature (which always fires), indicator features for whether this row corresponds to an argument or a predicate, and a discretized feature for how many alignments are in this row.

Predicate Argument Structure We expect structure among links that involve a predicate and its associated arguments. Therefore, we add joint factors that consider a predicate and its associated alignments: the predicate argument structure. We determine this structure from a dependency parse, though the idea is general to any semantic binding, e.g. FrameNet or Propbank style parses. Given a coherent discourse, there are several expected types of patterns in the PAS; we add factors for these.

Predicate-centric We begin with a predicate-centric factor, which views scores an alignment between predicates based on their arguments, i.e. the two predicates share the same arguments. Ideally, two predicates can only align when their arguments are coreferent. However, in practice we may incorrectly resolve argument links, or there may be implicit arguments that do not appear as syntactic dependencies of the predicate trigger. Therefore, we settle for a weaker condition, that there should be *some* overlap in the arguments of two coreferent predicates.

For every predicate alignment z^p_{ij}, we add a factor ϕ_{psa} whose score s_{psa} is a penalty for having no argument overlap; predicates share arguments (`psa`). To constrain the variable of ϕ_{psa}, we add a quadratic constraint that considers every possible pair of argument alignments that might overlap:

$$z_{\phi_{\text{psa}}} \geq z^p_{ij}\big(1 - \max_{\substack{k \in \text{args}(p_i) \\ l \in \text{args}(p_j)}} z^a_{kl}\big) \tag{6}$$

where $\text{args}(p_i)$ finds the indices of all arguments governed by the predicate p_i.

Entity-centric We expect similar behavior from arguments (entities). If an entity appears in two documents, it is likely that this entity will be mentioned in the context of a common predicate, i.e. arguments share predicates (`asp`). For a given argument alignment z^a_{ij} we add quadratic constraints so that $z_{\phi_{\text{asp}}}$ represents a penalty for two arguments not sharing a single predicate:

$$z_{\phi_{\text{asp}}} \geq z^a_{ij}\big(1 - \max_{\substack{k \in \text{preds}(a_i) \\ l \in \text{preds}(a_j)}} z^p_{kl}\big) \tag{7}$$

where $\text{preds}(a_i)$ finds the indices of all predicates that govern any mention of argument a_i.

The features $\mathbf{f}(\phi)$ for both `psa` and `asp` are an intercept feature and a bucketed count of the size of $\text{args}(p_i) \times \text{args}(p_j)$ or $\text{preds}(a_i) \times \text{preds}(a_j)$ respectively.

Temporal Information Temporal ordering, in contrast to textual ordering, can indicate when predicates cannot align: we expect aligned predicates in both documents to share the same temporal relations. SemEval 2013 included a task on predicting temporal relations between events (UzZaman et al., 2013). Many systems produced partial relations of events in a document based on lexical aspect and tense, as well as discourse connectives like "during" or "after". We obtain temporal relations with CAEVO, a state-of-the-art sieve-based system (Chambers et al., 2014).

TimeML (Pustejovsky et al., 2003), the format for specifying temporal relations, defines relations between predicates (e.g. *immediately before* and *simultaneous*), each with an inverse (e.g. *immediately after* and *simultaneous* respectively). We will refer to a relation as R and its inverse as R^{-1}. Suppose we had p_a and p_b in the source document, p_x and p_y in the target document, and $p_a R_1 p_b$, $p_x R_2 p_y$. Given this configuration the following alignments conflict with the in-doc relations:

z_{ax}	z_{by}	z_{ay}	z_{bx}	In-Doc Relations
*	*	1	1	$R_1 = R_2$
1	1	*	*	$R_1 = R_2^{-1}$

where 1 means there is a link and * means there is a link or no link (wildcard). The simplest example that fits this pattern is: 'a before b', 'x before y', 'a corefers with y', and 'b corefers with x' implies a conflict.

We introduce a factor that penalizes these conflicting configurations. In every instance where the predicted temporal relation for a pair of predicate alignments matches one of the conflict patterns above, we add a factor using $z_{\phi_{\text{temp}}}$:

$$\begin{aligned} z_{\phi_{\text{temp}}} &\geq z_{ay} z_{bx} \\ &\quad \text{if } p_a R_1 p_b, p_x R_2 p_y, R_1 = R_2 \\ z_{\phi_{\text{temp}}} &\geq z_{ax} z_{by} \\ &\quad \text{if } p_a R_1 p_b, p_x R_2 p_y, R_1 = R_2^{-1} \end{aligned} \tag{8}$$

Thus $s_{\phi_{\text{temp}}}$ is the cost of disagreeing with the in-doc temporal relations. This is a general technique for incorporating relational information into coreference decisions. It only requires specifying when two relations are incompatible, e.g. `spouseOf` and `siblingOf` are incompatible relations (in most states). We leave this for future work.

Since CAEVO gives each relation prediction a probability, we incorporate this into the feature by indicating the probability of a conflict *not* arising:

$$\mathbf{f}(\phi_{\text{temp}}) = \log\left(1 - p(R_1)p(R_2) + \epsilon\right) \tag{9}$$

ϵ avoids large negative values since CAEVO probabilities are not perfectly calibrated. We use $\epsilon = 0.1$, allowing feature values of at most -2.3.

Summary The objective is a linear function over binary variables. There is a local similarity score coefficient on every alignment variable, and a joint factor similarity score on every quadratic variable. These quadratic variables are constrained by products of the original alignment variables. Decoding an alignment requires solving this quadratically constrained integer program; in practice is can be solved quickly without relations.

```
def train(alignments):
  w = init_weights()
  working_set = set()
  while True:
    xi = solve_ILP(w, working_set)
    c = most_violated_constraint(w, alignments)
    working_set.add(c)
    if hinge(c, w) < xi:
      break

def most_violated_constraint(w, alignments):
  delta_features = vector()
  loss = 0
  for z in alignments:
    z_mv = make_ILP(z)
    for phi in factors:
      costs = dot(w, phi.features)
      z_mv.add_terms(costs, phi.vars)
      z_mv.add_constraints(phi.constraints)
    solve_ILP(z_mv)
    mu = (z.size + k) / (avg_z_size + k)
    delta_features += mu * (f(z) - f(z_mv))
    loss += mu * Delta(z, z_mv)
  return Constraint(delta_features, loss)

def hinge(c, w):
  return max(0, c.loss - dot(w, c.delta_features))
```

Figure 2: Learning algorithm (caching and ILP solver not shown). The sum in each constraint is performed once when finding the constraint, and implicitly thereafter.

5 Inference

Learning We use the supervised structured SVM formulation of Joachims et al. (2009). As is common in structure prediction we use margin rescaling and 1 slack variable, with the structural SVM objective:

$$\begin{gathered} \min_w ||w||_2^2 + C\xi \\ \text{s.t. } \xi \geq 0 \\ \xi + \sum_{i=1}^{N} w \cdot f(z_i) \geq \sum_{i=1}^{N} w \cdot f(\hat{z}_i) + \Delta(z_i, \hat{z}_i) \\ \forall \hat{z}_i \in \mathcal{Z}_i \end{gathered} \tag{10}$$

where $\mathcal{Z}_i$ is the set of all possible alignments that have the same shape as z_i.

The score function for an alignment uses three types of terms: weights, features, and alignment variables. When we decode, we take the product of the weights and the features to get the costs for the ILP (e.g. $s_\phi = w \cdot \mathbf{f}(\phi)$). When we optimize our SVM objective, we take the product of the alignment variables and the features to get modified features for the SVM:

$$f(z) = \sum_{ij} z_{ij}\mathbf{f}(z_{ij}) + \sum_{\phi \in \Phi} z_\phi \mathbf{f}(\phi) \qquad (11)$$

Since we cannot iterate over the exponentially many margin constraints, we solve for this optimization using the cutting-plane learning algorithm. This algorithm repeatedly asks the "separation oracle" for the most violated SVM constraint, which finds this constraint by solving:

$$\arg\max_{\hat{z_1} \ldots \hat{z_N}} \sum_i w \cdot f(\hat{z_i}) + \Delta(z_i, \hat{z_i}) \qquad (12)$$

subject to the constraints defined by the joint factors. When the separation oracle returns a constraint that is not violated or is already in the working set, then we have a guarantee that we solved the original SVM problem with exponentially many constraints. This is the most time-consuming aspect of learning, but since the problem decomposes over document alignments, we cache solutions on a per document alignment basis. With caching, we only call the separation oracle around 100-300 times.

We implement the separation oracle using an ILP solver, CPLEX,[3] due to complexity of the discrete optimization problem: there are 2^{m^n} possible alignments for and $m \times n$ alignment grid. In practice this is solved very efficiently, taking less than a third of a second per document alignment on average. We would like Δ to be F1, but we need a decomposable loss to include it in a linear objective (Taskar et al., 2003). Instead, we use Hamming loss as a surrogate, as in Lacoste-Julien et al. (2006).

Our training data is heavily biased towards negative examples, performing poorly on F1 since precision and recall are unbalanced. We use an asymmetric version of Hamming loss that incurs c_{FP} cost for predicting an alignment for two unaligned items and c_{FN} for predicting no alignment for two aligned items. We fixed $c_{FP} = 1$ and tuned $c_{FN} \in \{1, 2, 3, 4\}$ on dev data. Additionally we found it useful to tune the scale of the loss function across $\{\frac{1}{2}, 1, 2, 4\}$. Previous work, such as Joachims et al. (2009), use a hand-chosen constant for the scale of the Hamming loss, but we observe some sensitivity in this parameter and choose to optimize it.

Decoding Following Wolfe et al. (2013), we tune the threshold for classification τ on dev data to maximize F1 (via linesearch). For SVMs τ is typically fixed at 0: this is not necessarily good practice when your training loss differs from test loss (Hamming vs F1). In our case this extra parameter is worth allocating a portion of training data to enable tuning. Tuning τ addresses the same problem as using an asymmetric Hamming loss, but we found that doing both led to better results.[4] Since we are using a global scoring function rather than a set of classifications, τ is implemented as a test-time unary factor on every alignment.

6 Experiments

Data We consider two datasets for evaluation. The first is a cross-document entity and event coreference resolution dataset called the Extended Event Coref Bank (EECB) created by Lee et al. (2012) and based on a corpus from Bejan and Harabagiu (2010). The dataset contains clusters of news articles taken from Google News with annotations about coreference over entities and events. Following the procedure of Wolfe et al. (2013), we select the first document in every cluster and pair it with every other document in the cluster.

The second dataset (RF) comes from Roth and Frank (2012). The dataset contains pairs of news articles that describe the same news story, and are annotated for predicate links between the document pairs. Due to the lack of annotated arguments, we can only report predicate linking performance and the `psa` and `asp` factors do not apply. Lastly, the size of the RF data should be noted as it is much smaller than EECB: the test set has 60 document pairs and the dev set has 10 document pairs.

[3] `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

[4] Only tuning τ performed almost as well as tuning τ and the Hamming loss, but not tuning τ performed much worse than only tuning the Hamming loss at train time.

Both datasets are annotated with parses and in-document coreference labels provided by the toolset of Napoles et al. (2012)[5] and are available with our code release. Due to the small data size, we use k-fold cross validation for both datasets. We choose $k = 10$ for RF due to its very small size (more folds give more training examples) and $k = 5$ on EECB to save computation time (amount of training data in EECB is less of a concern). Hyperparameters were chosen by hand using using cross validation on the EECB dataset using F1 as the criteria (rather than Hamming). Figures report averages across these folds.

Systems Following Roth and Frank (2012) and Wolfe et al. (2013) we include a *Lemma* baseline for identifying alignments which will align any two predicates or arguments that have the same lemmatized head word.[6] The *Local* baseline uses the same features as Wolfe et al., but none of our joint factors. In addition to running our joint model with all factors, we measure the efficacy of each individual factor by evaluating each with the local features.

For evaluation we use a generous version of F1 that is defined for alignment labels composed of sure, G_s, and possible links, G_p and the system's proposed links H (following Cohn et al. (2008), Roth and Frank (2012) and Wolfe et al. (2013)).

$$P = \frac{|H \cap G_p|}{|H|} \quad R = \frac{|H \cap G_s|}{|G_s|} F = \frac{2PR}{P+R}$$

Note that the EECB data does not have a sure and possible distinction, so $G_s = G_p$, resulting in standard F1. In addition to F1, we separately measure predicate and argument F1 to demonstrate where our model makes the largest improvements.

We performed a one-sided paired-bootstrap test where the null hypothesis was that the joint model was no better than the *Local* baseline (described in Koehn (2004)). Cases where $p < 0.05$ are bolded.

[5] `https://github.com/cnap/anno-pipeline`

[6] The lemma baseline is obviously sensitive to the lemmatizer used. We used the Stanford CoreNLP lemmatizer (Manning et al., 2014) and found it yielded slightly better results than previously reported as the lemma baseline (Roth and Frank, 2012), so we used it for all systems to ensure fairness and that the baseline is as strong as it could be.

7 Results

Results for EECB and RF are reported in Table 7. As previously reported, using just local factors (features on pairs) improves over lemma baselines (Wolfe et al., 2013). The joint factors make statistically significant gains over local factors in almost all experiments. Fertility factors provide the largest improvements from any single constraint. A fertility penalty actually allows the pairwise weights to be more optimistic in that they can predict more alignments for reasonable pairs, allowing the fertility penalty to ensure only the best is chosen. This penalty also prevents the "garbage collecting" effect that arises for instances that have rare features (Brown et al., 1993).

Temporal constraints are relatively sparse, appearing just 2.8 times on average. Nevertheless, it was very helpful across all experiments, though only statistically significantly on the RF dataset. This is one of the first results to demonstrate benefits of temporal relations affecting an downstream task. Perhaps surprisingly, these improvements result from a a temporal relation system that has relatively poor absolute performance. Despite this, improvements are possibly due to the orthogonal nature of temporal information; no other feature captures this signal. This suggests that future work on temporal relation prediction may yield further improvements and deserves more attention as a useful feature for semantic tasks in NLP.

The predicate-centric factors improved performance significantly on both datasets. For the predicate-centric factor, when a predicate was aligned there is a 72.3% chance that there was at least one argument aligned as well, compared to only 14.1% of case of non-aligned predicates. As mentioned before, the reason the former number isn't 100% is primarily due to implicit arguments and errors in argument identification. The argument-centric features helped almost as much as the predicate-centric version, but the improvements were not significant on the EECB dataset. Running the same diagnostic as the predicate-centric feature reveals similar support: in 57.1% of the cases where an argument was aligned, at least one predicate it partook in was aligned too, compared to 7.6% of cases for non-aligned arguments. Both the

	EECB								
	F1	P	R	Arg F1	Arg P	Arg R	Pred F1	Pred P	Pred R
Lemma	68.1	79.3 *	59.6	61.7	79.1 *	50.6	75.0	87.3 *	65.7
Local	73.0	75.8	**70.5**	67.7	76.3	**60.8**	78.7	81.4	76.2
+Fertility	77.1 *	83.9 *	71.3	66.6	80.9 *	56.6	82.8 *	87.4 *	**78.7** *
+Predicate-centric	74.1 *	80.7 *	68.6	67.4	81.6 *	57.3	79.7 *	85.0 *	75.1
+Argument-centric	73.7	81.2 *	67.5	66.8	**83.0** *	55.9	79.3	85.1 *	74.3
+Temporal	73.7	78.2 *	69.7	**67.9**	80.6 *	58.7	79.0	82.1	76.1
+All Factors	**77.5** *	**86.3** *	70.3	65.8	83.1 *	54.5	**83.7** *	**89.7** *	78.4 *

	RF		
	Pred F1	Pred P	Pred R
Lemma	52.4	47.6	58.2 *
Local	58.1	**63.5**	53.6
+Fertility	**60.0**	57.4	**62.4** *
+Predicate-centric	NA	NA	NA
+Argument-centric	NA	NA	NA
+Temporal	59.0	57.4	60.6 *
+All factors	59.4	56.9	62.2 *

Figure 3: Cross validation results for EECB (above) (Lee et al., 2012) and RF (left) (Roth and Frank, 2012). Statistically significant improvements from Local marked * ($p < 0.05$ using a one-sided paired-bootstrap test) and best results are bolded.

predicate- and argument-centric improve similarly across both predicates and arguments on EECB.

While each of the joint factors all improve over the baselines on RF, the full model with all the joint factors does not perform as well as with some factors excluded. Specifically, the fertility model performs the best. We attribute this small gap to lack of training data (RF only contains 64 training document pairs in our experiments), as this is not a problem on the larger EECB dataset.

Additionally, the joint models seem to trade precision for recall on the RF dataset compared to the *Local* baseline. Note that both models are tuned to maximize F1, so this tells you more about the shape of the ROC curve as opposed to either models' ability to achieve either high precision or recall. Since we don't see this behavior on the EECB corpus, it is more likely that this is a property of the data than the model.

8 Related Work

The task of predicate argument linking was introduced by Roth and Frank (2012), who used a graph parameterized by a small number of semantic features to express similarities between predicates and used min-cuts to produce an alignment. This was followed by Wolfe et al. (2013), who gave a locally-independent, feature-rich log-linear model that utilized many lexical semantic resources, similar to the sort employed in RTE challenges.

Lee et al. (2012) considered a similar problem but sought to produce *clusters* of entities and events rather than an alignment between two documents with the goal of improving coreference resolution. They used features which consider previous event and entity coreference decisions to make future coreference decisions in a greedy manner. This differs from our model which is built on non-greedy joint inference, but much of the signal indicating when two mentions corefer or are aligned is similar.

In the context of in-document coreference resolution, Recasens et al. (2013) sought to overcome the problem of opaque mentions[7] by finding high-precision paraphrases of entities by pivoting off verbs mentioned in similar documents. We address the issue of opaque mentions not by building a paraphrase table, but by jointly reasoning about entities that participate in coreferent events (c.f. §4); the approaches are complementary.

In this work we incorporate ordering information of events. Though we consider it an upstream task, there is a line of work trying to predict temporal relations between events (Pustejovsky et al., 2003; Mani et al., 2006; Chambers et al., 2014). Our results indicate this is a useful source of information, one of the first results to show an improvement from this

[7] A lexically disparate description of an entity.

type of system (Glavaš and Šnajder, 2013).

We utilize an ILP to improve upon a pipelined system, similar to Roth and Yih (2004), but our work differs in that we do not use piecewise-trained classifiers. Our local similarity scores are calibrated according to a global objective by propagating the gradient back from the loss to every parameter in the model. When using piecewise training, local classifiers must focus more on recall (in the spirit of Weiss and Taskar (2010)) than they would for an ordinary classification task with no global objective. Our method trains classifiers jointly with a global convex objective. While our training procedure requires decoding an integer program, the parameters we learn are globally optimal.

9 Conclusion

We presented a max-margin quadratic cost model for predicate argument alignment, seeking to exploit discourse level semantic features to improve on previous, locally independent approaches. Our model includes factors that consider fertility of predicates and arguments, the predicate argument structure present in coherent discourses, and soft constraints on predicate coreference determined by a temporal relation classifier. We have shown that this model significantly improves upon prior work which uses extensive lexical resources but without the benefit of joint inference. Additionally, this is one of the first demonstrations of the benefits of temporal relation identification. Overall, this work demonstrates the benefits of considering global document information as part of natural language understanding.

Future work should extend the problem formulation of predicate argument alignment to consider *incremental* linking: starting with a pair of documents, perform linking, and then continue to add in documents over time. This problem formulation would capture the evolution of a breaking news story, which closely matches the type of data (news articles) considered in this work (EECB and RF datasets). This formulation ties into existing work on news summarization, topic detection and tracking, an multi-document NLU. This goes hand with work on better intra-document relation prediction methods, such as the temporal relation model used in this work, to lead to better joint linking decisions.

References

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *EMNLP-CoNLL*, pages 344–355. ACL.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, ACL '98, pages 86–90, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1412–1422, Stroudsburg, PA, USA. Association for Computational Linguistics.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, Meredith J. Goldsmith, Jan Hajic, Robert L. Mercer, and Surya Mohanty. 1993. But dictionaries are data too. In *Proceedings of the Workshop on Human Language Technology*, HLT '93, pages 202–205, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Comput. Linguist.*, 34(4):597–614, December.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

Goran Glavaš and Jan Šnajder. 2013. Recognizing identical events with graph kernels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 797–803, Sofia, Bulgaria, August. Association for Computational Linguistics.

Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural svms. *Mach. Learn.*, 77(1):27–59, October.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Simon Lacoste-Julien, Benjamin Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In Robert C. Moore, Jeff A. Bilmes, Jennifer Chu-Carroll, and Mark Sanderson, editors, *HLT-NAACL*. The Association for Computational Linguistics.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA. Association for Computational Linguistics.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 753–760. Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

George A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *AKBC-WEKEX Workshop at NAACL 2012*, June.

James Pustejovsky, Jos Castao, Robert Ingria, Roser Saur, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. Timeml: Robust specification of event and temporal expressions in text. In *in Fifth International Workshop on Computational Semantics (IWCS-5)*.

Marta Recasens, Matthew Can, and Daniel Jurafsky. 2013. Same referent, different words: Unsupervised mining of opaque coreferent mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906, Atlanta, Georgia, June. Association for Computational Linguistics.

Michael Roth and Anette Frank. 2012. Aligning predicate argument structures in monolingual comparable texts: a new corpus for a new task. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 218–227, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *In Proceedings of CoNLL-2004*, pages 1–8.

Michael Roth. 2014. *Inducing Implicit Arguments via Cross-document Alignment: A Framework and its Applications*. Ph.D. thesis, Heidelberg University, June.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. MIT Press.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

David Weiss and Benjamin Taskar. 2010. Structured prediction cascades. *Journal of Machine Learning Research - Proceedings Track*, 9:916–923.

Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Bellar, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathann Weese, Tan Xu, and Xuchen Yao. 2013. Parma: A predicate argument aligner. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, July.

Xuchen Yao, Benjamin Van Durme, Chris Callison-burch, and Peter Clark. 2013. Answer extraction as sequence tagging with tree edit distance. In *In North American Chapter of the Association for Computational Linguistics (NAACL*.