Object Consolodation by Graph Partitioning with a Conditionally-Trained Distance Metric

Andrew McCallum[†]
[†]Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003 USA
mccallum@cs.umass.edu

Ben Wellner[†*]
[*]The MITRE Corporation
202 Burlington Road
Bedford, MA 01730 USA
wellner@cs.umass.edu

ABSTRACT

Coreference analysis, also known as record linkage, object consolidation or identity uncertainty, is a difficult and important problem in natural language processing, databases, citation matching and many other tasks. This paper introduces several discriminative, conditional-probability models for coreference analysis, all examples of undirected graphical models. Unlike many historical approaches to coreference, the models presented here are relational—they do not assume that pairwise coreference decisions should be made independently from each other. Unlike other relational models of coreference that are generative, the conditional model here can incorporate a great variety of features of the input without having to be concerned about their dependencies—paralleling the advantages of conditional random fields over hidden Markov models. We present positive results on noun coreference in two standard text data sets.

1. INTRODUCTION

In many domains—including computer vision, databases and natural language processing—we find multiple views, descriptions, or names for the same underlying object. Correctly resolving these references is a necessary precursor to further processing and understanding of the data. In computer vision, solving object correspondence is necessary for counting or tracking. In databases, performing record linkage or de-duplication creates a clean set of data that can be accurately mined. In natural language processing, coreference analysis finds the nouns, pronouns and general noun phrases that refer to the same entity, enabling the extraction of relations among entities as well as more complex propositions.

Consider, for example, the text in a news article that discusses the entities *George Bush*, *Colin Powell*, and *Donald Rumsfeld*. The article contains multiple mentions of Colin Powell by different strings—"Secretary of State Colin Powell," "he," "Mr. Powell," "the Secretary"—and also refers to the other two entities with sometimes overlapping strings. The coreference task is to use the content and context of all the mentions to determine how many entities are in the article, and which mention corresponds to which entity.

This task is most frequently solved by examining individual pair-wise distance measures between mentions independently of each other. For example, database record-linkage and citation reference matching has been performed by learning a pairwise distance metric between records, and setting a distance threshold below which records are merged [13, 11, 2, 4]. Coreference in NLP has also been performed with distance thresholds or pairwise classifiers [12, 7, 19, 15].

But these distance measures are inherently noisy and the answer to one pair-wise coreference decision may not be independent of another. For example, if we measure the distance between all of the three possible pairs among three mentions, two of the distances may be below threshold, but one above—an inconsistency due to noise and imperfect measurement. For example, "Mr. Powell" may be correctly coresolved with "Powell," but particular grammatical circumstances may make the model incorrectly believe that "Powell" is coreferent with a nearby occurrence of "she". Inconsistencies might be better resolved if the coreference decisions are made in *dependent relation* to each other, and in a way that accounts for the values of the multiple distances, instead of a threshold on single pairs independently.

Recently [16] have proposed a formal, relational approach to the problem of identity uncertainty using a type of Bayesian network called a Relational Probabilistic Model [6]. A great strength of this model is that it explicitly captures the dependence among multiple coreference decisions. However, it is a generative model of the entities, mentions and all their features, and thus has difficulty using many features that are highly overlapping, non-independent, at varying levels of granularity, and with long-range dependencies. For example, we might wish to use features that capture the phrases, words and character n-grams in the mentions, the appearance of keywords anywhere in the document, the parse-tree of the current, preceding and following sentences, as well as 2-d layout information. To produce accurate generative probability distributions, the dependencies between these features should be captured in the model; but doing so can lead to extremely complex models in which parameter estimation is nearly impossible.

Similar issues arise in sequence modeling problems. In this area significant recent success has been achieved by replacing a generative model—hidden Markov models—with a conditional model—conditional random fields (CRFs) [10]. CRFs have reduced part-of-speech tagging errors by 50% on out-of-vocabulary words in comparison with HMMs, matched champion noun phrase segmentation results [18], and significantly improved the segmentation of tables in government reports [17]. Relational Markov networks [20] are similar models, and have been shown to significantly improve classification of Web pages.

This paper introduces three conditional undirected graphical models for identity uncertainty. The models condition on the mentions, and generate the coreference decisions, (and in some cases also generate attributes of the entities). In the first most general model, the dependency structure is unrestricted, and the number of underlying entities explicitly appears in the model structure. The second and third models have no structural dependence on the number of entities, and fall into a class of Markov random fields in which inference corresponds to graph partitioning [3].

We show experimental results using the third model on a noun coreference problem in two different standard newswire text domains: broadcast news stories from the DARPA Automatic Content Extraction (ACE) program, and newswire articles from the MUC-6 corpus. In both domains we take advantage of the ability to use arbitrary, overlapping features of the input, including multiple grammatical features, string equality, substring, and acronym matches. Using the same features, in comparison with an alternative natural language processing technique, we reduce error by 33% and 28% in the two domains on proper nouns and by 10% on all nouns in the MUC-6 data.

2. THREE CONDITIONAL MODELS OF IDENTITY UNCERTAINTY

We now describe three conditional models of identity uncertainty, each progressively simpler and more specific than its predecessor. All three are based on conditionally-trained, undirected graphical models also known as Markov networks or Markov random fields. These models excel at capturing interdependent data in which causality among attributes is not apparent. We begin by introducing notation for mentions, entities and attributes of entities, then in the following subsections describe the likelihood, inference and estimation procedures for the specific models.

Let $\mathbf{E} = (E_1, ...E_m)$ be a collection of classes or "entities". Let $\mathbf{X} = (X_1, ...X_n)$ be a collection of random variables over observations or "mentions"; and let $\mathbf{Y} = (Y_1, ...Y_n)$ be a collection of random variables over integer identifiers, unique to each entity, specifying to which entity a mention refers. Thus the y's are integers ranging from 1 to m, and if $Y_i = Y_j$, then mention X_i is said to refer to the same underlying entity as X_j. For example, some particular entity e_4, *U.S. Secretary of State, Colin L. Powell*, may be mentioned multiple times in a news article that also contains mentions of other entities: x_6 may be "Colin Powell"; x_9 may be "he"; x_{17} may be "the Secretary of State." In this case, the unique integer identifier for this entity, e_4, is 4, and $y_6 = y_9 = y_{17} = 4$.

Furthermore, entities may have attributes. Let $\mathbf{A}$ be a random variable over the collection of all attributes for all entities. Borrowing the notation of Relational Markov Networks [20], we write the random variable over the attributes of entity E_s as $E_s.\mathbf{A} = \{E_s.A_1, E_s.A_2, E_s.A_3, ...\}$. For example, these three attributes may be *gender, birth year*, and *surname*. Continuing the above example, then $e_4.a_1$ = MALE, $e_4.a_2$ = 1937, and $e_4.a_3$ = Powell. One can interpret the attributes as the values that should appear in the fields of a database record for the given entity. Attributes such as *surname* may take on one of the finite number of values that appear in the mentions of the data set.

We may examine many features of the mentions, $\mathbf{x}$, but since a conditional model doesn't generate them, we don't need random variable notation for them. Separate measured features of the mentions and entity-assignments, $\mathbf{y}$, are captured in different feature functions, $f(\cdot)$, over cliques in the graphical model. Although the functions may be real-valued, typically they are binary. The parameters of the model are associated with these different feature functions. Details and example feature functions and parameterizations are given for the three specific models below.

The task is then to find the most likely collection of entity-assignments, $\mathbf{y}$, (and optionally also the most likely entity attributes, $\mathbf{a}$), given a collection of mentions and their context, $\mathbf{x}$. A generative probabilistic model of identity uncertainty is trained to maximize $P(\mathbf{Y}, \mathbf{A}, \mathbf{X})$. A conditional probabilistic model of identity uncertainty is instead trained to maximize $P(\mathbf{Y}, \mathbf{A}|\mathbf{X})$, or simply $P(\mathbf{Y}|\mathbf{X})$.

2.1 Model 1: Groups of nodes for entities

First we consider an extremely general undirected graphical model in which there is a single node for all of the mentions, $\mathbf{x}$,[1] a node for the entity-assignment of each mention, y, and a node for each of the attributes of each entity, $e.a$. These nodes are connected by edges in some unspecified structure, where an edge indicates that the values of the two connected random variables are dependent on each other.

The parameters of the model are defined over cliques in this graph. Typically the parameters on many different cliques would be tied in patterns that reflect the nature of the repeated relational structure in the data. Patterns of tied parameters are common in many graphical models, including HMMs and other finite state machines [10], where they are tied across different positions in the input sequence, and by more complex patterns based on SQL-like queries, as in Markov Relational Networks [20]. Following the nomenclature of the later, these parameter-tying-patterns are called *clique templates*; each particular instance of a template in the graph we call a *hit*.

For example, one clique template may specify a pattern consisting of two mentions, their entity-assignment nodes, and an entity's *surname* attribute node. The hits would consist of all possible combinations of such nodes. Multiple feature functions could then be run over each hit. One feature function might have value 1 if, for example, both mentions were assigned to the same entity as the surname node, and if the surname value appears as a substring in both mention strings (and value 0 otherwise).

The Hammersley-Clifford theorem stipulates that the probability of a particular set of values on the random variables in an undirected graphical model is a normalized product of potential functions over cliques of the graph. Our cliques will be the hits, $\mathbf{h} = \{h, ...\}$, resulting from a set of clique templates, $\mathbf{t} = \{t, ...\}$. In typical fashion, we will write the probability distribution in exponential form, with each po-

[1]Even though there are many mentions in $\mathbf{x}$, because we are not generating them, we can represent them as a single node. This helps show that feature functions can ask arbitrary questions about various large and small subsets of the mentions and their context. We will still use x_i to refer to the content and context of the ith mention.

tential function calculated as a dot-product of feature functions, f, and learned parameters, λ,

$$P(\mathbf{y}, \mathbf{a}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_{t \in \mathbf{t}} \sum_{h_t \in \mathbf{h}_t} \sum_{l} \lambda_l f_l(\mathbf{y}, \mathbf{a}, \mathbf{x} : h_t)\right),$$

where $(\mathbf{y}, \mathbf{a}, \mathbf{x} : h_t)$ indicates the subset of the entity-assignment, attribute, and mention nodes selected by the clique template hit h_t; and $Z_{\mathbf{x}}$ is a normalizer to make the probabilities over all $\mathbf{y}$ sum to one (also known as the partition function).

The parameters, λ, can be learned by maximum likelihood from labeled training data. Calculating the partition function, $Z_{\mathbf{x}}$, is problematic because there are a very large number of possible $\mathbf{y}$'s and $\mathbf{a}$'s. Loopy belief propagation or Gibbs sampling sampling have been used successfully in other similar situations, *e.g.* [20].

However, note that both loopy belief propagation and Gibbs sampling only work over a graph with fixed structure. But in our problem the number of entities (and thus number of attribute nodes, and the domain of the entity-assignment nodes) is unknown. Inference in these models must determine for us the highest-probability number of entities.

In related work on a generative probabilistic model of identity uncertainty, [16], solve this problem by alternating rounds of Metropolis-Hastings sampling on a given model structure with rounds of Metropolis-Hastings to explore the space of new graph structures. Our desire to avoid the complexity and low scalability of this approach motivates our Model 2.

2.2 Model 2: Nodes for mention pairs, with attributes on mentions

To avoid the need to change the graphical model structure during inference, we now remove any parts of the graph that depend on the number of entities, m: (1) The per-mention entity-assignment nodes, Y_i, are random variables whose domain is over the integers 0 through m; we remove these nodes, replacing them with binary-valued random variables, Y_{ij}, over each pair of mentions, (X_i, X_j) (indicating whether or not the two mentions are coreferent); although it is not strictly necessary, we also restrict the clique templates to operate over no more than two mentions (for efficiency). (2) The per-entity attribute nodes A are removed and replaced with attribute nodes associated with each mention; we write $x_i.\mathbf{a}$ for the set of attributes on mention x_i.

Even though the clique templates are now restricted to pairs of mentions, this does not imply that pairwise coreference decisions are made independently of each other—they are still highly dependent. Many pairs will overlap with each other, and constraints will flow through these overlaps. This point is reiterated with an example in the next subsection.

Notice, however, that it is possible for the model as thus far described to assign non-zero probability to an inconsistent set of entity-assignments, $\mathbf{y}$. For example, we may have an "inconsistent triangle" of coreference decisions in which y_{ij} and y_{jk} are 1, while y_{ik} is 0. We can enforce the impossibility of all inconsistent configurations by adding inconsistency-checking functions $f_*(y_{ij}, y_{jk}, y_{ik})$ for all mention triples, with the corresponding λ_*'s fixed at negative infinity—thus assigning zero probability to them. (Note that this is simply a notational trick; in practice the inference implementation simply avoids any configurations of $\mathbf{y}$ that are inconsistent.) Thus we have

$$\begin{aligned} P(\mathbf{y}, \mathbf{a}|\mathbf{x}) &= \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_{i,j,l} \lambda_l f_l(x_i, x_j, y_{ij}, x_i.\mathbf{a}, x_j.\mathbf{a})\right. \\ &\qquad \left. + \sum_{i,j,k} \lambda_* f_*(y_{ij}, y_{jk}, y_{ik})\right). \end{aligned}$$

We can also enforce consistency among the attributes of coreferent mentions by similar means. There are many widely-used techniques for efficiently and drastically reducing the number of pair-wise comparisons, *e.g.* [13, 11]. In this case, we could also restrict $f_l(x_i, x_j, y_{ij}) \equiv 0, \forall y_{ij} = 0$.

2.3 Model 3: Nodes for mention pairs, graph partitioning with learned distance metric

When gathering attributes of entities is not necessary, we can avoid the extra complication of attributes by removing them from the model. What results is a straightforward, yet highly expressive, discriminatively-trained, undirected graphical model that can use rich feature sets and relational inference to solve identity uncertainty tasks. Determining the most likely number of entities falls naturally out of inference. The model is

$$\begin{aligned} P(\mathbf{y}|\mathbf{x}) &= \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_{i,j,l} \lambda_l f_l(x_i, x_j, y_{ij})\right. \qquad (1) \\ &\qquad \left. + \sum_{i,j,k} \lambda_* f_*(y_{ij}, y_{jk}, y_{ik})\right). \end{aligned}$$

Recently there has been interest in study of the equivalence between graph partitioning algorithms and inference in certain kinds of undirected graphical models, *e.g.* [3]. This graphical model is an example of such a case. With some thought, one can straightforwardly see that finding the highest probability coreference solution, $\mathbf{y}^\star = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$, exactly corresponds to finding the graph partitioning of a (different) graph in which the entities are the nodes and the edge weights are the (log) clique potentials on the pair of nodes $\langle x_i, x_j \rangle$ involved in their edge: $\sum_l \lambda_l f_l(x_i, x_j, y_{ij})$, where $f_l(x_i, x_j, 1) = -f_l(x_i, x_j, 0)$, and edge weights range from $-\infty$ to $+\infty$. Unlike classic mincut/maxflow binary partitioning, here the number partitions (corresponding to entities) is unknown, but a single optimal number of partitions exists; negative edge weights encourage more partitions.

Graph partitioning with negative edge weights is NP-hard, and we are forced to make use of approximation algorithms. Our current experiments use a variation of the minimizing-disagreements clustering algorithm in [1]. It works by measuring the degree of inconsistency incurred by including a node in a partition, and making repairs.

The resulting solution does not make pairwise coreference decisions independently of each other. It has a significant "relational" nature because the assignment of a node to a partition (or, mention to an entity) depends not just on a single low distance measurement to one other node, but on its low distance measurement to all nodes in the partition (and furthermore on its high distance measurement to all nodes of all other partitions). For example, the "Mr. Powell"/"Powell"/"she" problem discussed in the introduction would be prevented by this model because, although the distance between "Powell" and "she" might grammatically look low, the distance from "she" to another member of the same partition, ("Mr. Powell") is very high.

Interestingly, in our model, the distance measure between nodes is learned from labeled training data. That is, we use data, $\mathcal{D}$, in which the correct coreference partitions are known in order to learn a distance metric such that, when the same data is clustered, the correct partitions emerge. This is accomplished by maximum likelihood—adjusting the weights, λ, to maximize the product of Equation 1 over all instances $\langle \mathbf{x}, \mathbf{y} \rangle$ in the training set. Fortunately this objective function is concave—it has a single global maximum—and there are several applicable optimization methods to choose from, including gradient ascent, stochastic gradient ascent and conjugate gradient; all simply require the derivative of the objective function. The derivative of the log-likelihood, L, is

$$\frac{\partial L}{\partial \lambda_l} = \sum_{\langle \mathbf{x},\mathbf{y} \rangle \in \mathcal{D}} \left(\sum_{i,j,l} f_l(x_i, x_j, y_{ij}) - \sum_{\mathbf{y}'} P_\Lambda(\mathbf{y}'|\mathbf{x}) \sum_{i,j,l} f_l(x_i, x_j, y'_{ij}) \right),$$

where $P_\Lambda(\mathbf{y}'|\mathbf{x})$ is defined by Equation 1, using the current set of λ parameters, Λ, and $\sum_{\mathbf{y}'}$ is a sum over all possible partitionings.

The number of possible partitionings is exponential in the number of mentions, so for any reasonably-sized problem, we obviously must resort to approximate inference for the second expectation. A simple option is stochastic gradient ascent in the form of a voted perceptron [5]. Here we calculate the gradient for a single training instance at a time, and rather than use a full expectation in the second line, simply using the single most likely (or nearly most likely) partitioning as found by a graph partitioning algorithm, and make progressively smaller steps in the direction of these gradients while cycling through the instances, $\langle \mathbf{x}, \mathbf{y} \rangle$ in the training data. Neither the full sum, $\sum_{\mathbf{y}'}$, or the partition function, $Z_\mathbf{x}$, need to be calculated in this case. Further details are given in [5].

3. GRAPH PARTITIONING

In our model, inference, or more precisely, finding the highest probability set of coreference decisions, can be performed by graph partitioning in which the edges of the graph have weight $w_{ij} = \sum_l (\lambda_l f_l(x_i, x_j)$. The objective function is:

$$Obj = \sum_{ij} w_{ij} f(i,j)$$

$$\text{where} f(i,j) = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are in the same partition} \\ -1 & \text{otherwise} \end{cases}$$

This problem is a general case of the problem discussed in [1] where results for various graph partitioning algortihms are discussed when the edge weights are in $-1, 1$ or in $[-1, 1]$; in our case edge weights are in $[-\infty, +\infty]$. The algorithm for *minimizing disagreements* discussed there works as follows. Choose a vertex, v, at random. Form an initial cluster, C around v which consists of all the neighbors whose incident edge with v is $+1$. Then, randomly choose vertices from C and remove them if the ratio of positive edges to negative edges with other vertices within C is below some threshold. This continues until no vertices in C can be removed. A second pass considers all the remaining vertices outside C and adds them to the cluster if the ratio of positive negative incident edges with vertices in C is above a second threshold. At this point C is decided and its vertices removed from the graph.

As mentioned in the previous section, we use a variant of the above algrotihm with a couple of key differences. First, the selection of an initial vertex from which to form the initial cluster is chosen with probability proportional to the sum of the absolute value of each of its incident edges, instead of at random. The motivation behind this is to choose initial vertices whose incident edge-weights have the highest variance. These are the vertices that the model, in some sense, has the most confidence in. The initial cluster C is populated with neighboring vertices whose incident edges are positive. The second major difference is to select the vertices to remove with probability proportional to the sum of incident edges with other vertices within C. This is in contrast to choosing which vertices to remove at random. If the sum is negative for the chosen vertex, it is removed. This process tends to remove the worst (i.e. most incompatible with C) vertices first. The order of removal is imporant as C is updated at each iteration. The second phase adds vertices to C, but in this case C is not updated after each addition and so the order of addition is irrelevant. In summary, we have modified the criterion for vertex addition and removal, and we have introduced stochastic processes which tend to pick initial vertices that have high incident edge-variance and pick the worst vertices for removal first.

We use this algorithm to produce a sample of partitions for a given graph. Because a partition can be quickly evaluated using the objective function above, we can simply select the best partition of a large number of samples. This algorithm has running time $O(n^2)$ where n is the number of vertices. In practice, we have found that our largest graphs of 350 vertices or so take roughly 2-3 seconds to partition with times much shorter on smaller graphs. This makes samples of 1000s of partitions feasible on all but the largest of graphs.

This algorithm tends to work well in practice on the graphs involved in the coreference task. Along the way we have explored a number of existing partitioning algorithms as well as some original modifications to such algorithms. Many popular graph partitioning algorithms work when edge weights are greater than zero or simply binary. They typically try to produce *balanced* bi-partitions over a graph. This amounts to finding the minimal cut of the graph with a bias towards each partition having the same number of vertices. We explored using these algorithms to find coarse partitions within the graph and then made use of an exhaustive partitioning algorithm once the sub-graphs were small enough. We also developed a variation of the Fidduccia-Mattheyses partitioning algorithm that works for negative edge-weights. In all of these algorithms and variations we recursively bisect the graph to form our partitions. The recursion stops when the bisection of the graph results in a drop in the objective function for that sub-graph. In all cases, we found that the more traditional algorithms and our variations on them performed better than our algorithm here on *randomly* generated graphs that have roughly equal numbers of posi-

```
Given a graph, $G$
  • choose a vertex, $v_i \in G$ with probability proportional to the sum: $\sum_j |w_{i,j}|$
  • form an initial cluster, $C$, that includes $v_i$ and all vertices connected to $v_i$ with edge-weight greater than zero.
  • repeat
      – select $v_j \in C$ with probability proportional to $\sum_{k \in C} w_{ij}$
          ∗ if $\sum_k w_{jk} f(j,k) < 0$ (where $f(j,k) = 1$ if $v_k \in C$ and $f(i,j) = -1$ otherwise)
          ∗ then remove $v_j$
      – until $\forall v_j \in C \sum_k w_{jk} f(j,k) \geq 0$
  • for all $v_j \notin C$ such that $\sum_k w_{jk} f(j,k) > 0$ (where $f(j,k) = 1$ if $v_k \in C$ and $f(i,j) = -1$ otherwise)
      – add $v_j$ to $C$
  • let $G = G \setminus C$
  • repeat until $G$ is empty
```

Figure 1: Stochastic graph partitioning algorithm.

tive and negative edges. However, our algorithm perfromed better on the graphs for the coreference task discussed here where negative edges greatly outnumber positive ones. From inspection, the recursive bisectioning of the graph appears to cause disastrous errors as large coreference clusters can be split early on due to a few low-valued edge-weights.

The performance of the graph partitioning algorithm is crucial to our approach. Besides having high accuracy (i.e. the ability to find a partitioning close to the optimal one) it is desirable to have a stable algorithm in the sense that it will tend to have low variance in performance - i.e. the difference between the value of the objective function for the found partition and the objective value for the optimal partition has low variance on different graphs. The reason for this is clear: we compute our model expectations using the best partitioning found and the updating of model parameters is very sensistive to this. An erratic partitioning algorithm - even one with good expected performance - may therefore make learning within the voted perceptron difficult or slow to converge.

Many extensions are possible to our approach. One possible extension is to run the algorithm as is and then refine partitions by moving vertices to other partitions or making them singletons based on various criteria. By defining a set of moves possible at each step, we can make use of techniques such as simluated annealing. Altogether different approaches might make use of linear programming approximations of an integer programming formulation of the objective function [9]. Finally, more combinations of various techniques are worth exploring. For example, finding a good bi-partitioning first using known fast methods and then switching to a different, perhaps more epensive, algorithm may prove useful on very large graphs.

4. EXPERIMENTS WITH NOUN COREFERENCE

We test our approach to identity uncertainty without attributes (Model 3) on natural language noun coreference, using two different data sets: the DARPA MUC-6 corpus, and a set of 117 stories from the broadcast news portion of

	ACE (Proper)	MUC-6 (Proper)	MUC-6 (All)
best-previous-match	90.98	88.83	70.41
single-link-threshold	91.65	88.90	60.83
Model 3	93.96	91.59	73.42

Table 1: F1 results on three data sets.

the DARPA ACE data set. Both data sets have annotated coreferences. We pre-process both data sets with the Brill part-of-speech tagger.

We compare our Model 3 against two other techniques representing typical approaches to the problem of identity uncertainty. The first is single-link clustering with a threshold, (*single-link-threshold*), which is universally used in database record-linkage and citation reference matching [13, 2, 11, 4]. It forms partitions by simply collapsing the spanning trees of all mentions with pairwise distances below some threshold. For each experiment, the threshold was selected by cross validation.

The second technique, which we call *best-previous-match*, has been used in natural language processing applications [14, 7, 15]. It works by scanning linearly through a document, and associating each mention with its best-matching predecessor—best as measured with a single pairwise distance.

In our experiments, both single-link-threshold and best-previous-match implementations use a distance measure based on a binary maximum entropy classifier—matching the practice of [14] and [4].

We use an identical feature set for all techniques, including our Method 3. The features, typical of those used in many other NLP coreference systems, are modeled after those in [15]. They include tests for string and substring matches, acronym matches, parse-derived head-word matches, gender, WordNet subsumption, sentence distance, distance in the parse tree; etc., and are detailed in an accompanying technical report. They are quite non-independent, and operate at multiple levels of granularity.

Table 1 shows standard MUC-style F1 scores for three experiments. In the first two experiments, we consider only proper nouns, and perform five-fold cross validation. In the third experiment, we perform the standard MUC evaluation, including all nouns—pronouns, common and proper—and use the standard 30/30 document train/test split; furthermore, as in [8], we consider only mentions that have one or more coreferents. Model 3 out-performs both the single-link-threshold and the best-previous-match techniques, reducing error by 28% over single-link-threshold on the ACE proper noun data, by 24% on the MUC-6 proper noun data, and by 10% over the best-previous-match technique on the full MUC-6 task. Historically, these data sets have been heavily studied, and even small gains have been celebrated.

Our overall results on MUC-6 are slightly better (with unknown statistical significance) than the best published results of which we are aware with a matching experimental design, [8], who reach 72.3% using the same training and test data.

5. RELATED WORK AND CONCLUSIONS

There has been much related work on identity uncertainty in various specific fields. Traditional work in de-duplication for databases or reference-matching for citations measure the distance between two records by some metric, and then collapse all records at a distance below a threshold, *e.g.* [13, 11]. This method is not relational, that is, it does not account for the inter-dependent relations among multiple decisions to collapse. Most recent work in the area has focused on learning the distance metric [2, 4] not the clustering method.

Natural language processing has had similar emphasis and lack of emphasis respectively. Pairwise coreference learned distance measures have used decision trees [12, 15], SVMs [21], maximum entropy classifiers [14], and generative probabilistic models [7]. But all use thresholds on a single pairwise distance, or the maximum of a single pairwise distance to determine if or where a coreferent merge should occur.

[16] introduce a generative probability model for identity uncertainty based on Probabilistic Relational Networks networks [6]. Our work is an attempt to gain some of the same advantages that CRFs have over HMMs by creating conditional models of identity uncertainty. The models presented here, as instances of conditionally-trained undirected graphical models, are also instances of relational Markov networks [20] and conditional Random fields [10]. [20] briefly discuss clustering of dyadic data, such as people and their movie preferences, but not identity uncertainty or inference by graph partitioning.

Identity uncertainty is a significant problem in many fields. In natural language processing, it is not only especially difficult, but also extremely important, since improved coreference resolution is one of the chief barriers to effective data mining of text data. Natural language data is a domain that has particularly benefited from rich and overlapping feature representations—representations that lend themselves better to conditional probability models than generative ones [10, 5, 14]. Hence our interest in conditional models of identity uncertainty.

Acknowledgments

We thank Andrew Ng, Jon Kleinberg, David Karger, Avrim Blum and Fernando Pereira for helpful and insightful discussions. This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant numbers N66001-99-1-8912 and N66001-02-1-8903, and DARPA under contract number F30602-01-2-0566.

6. REFERENCES

[1] N. Bansal, S. Chawala, and A. Blum. Correlation clustering. In *The 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 238–247, 2002.

[2] M. Bilenko and R. J. Mooney. Learning to combine trained distance metrics for duplicate detection in databases. Technical Report AI 02-296, Artificial Intelligence Lab, University of Texas at Austin, February 2002.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV (1)*, pages 377–384, 1999.

[4] W. Cohen and J. Richman. Learning to match and cluster entity names. In *Proceedings of KDD-2002, 8th International Conference on Knowledge Discovery and Data Mining*, 2002.

[5] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms, 2002.

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.

[7] N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–171, 1998.

[8] S. Harabagiu, R. Bunescu, and S. Maiorano. Text and knowledge mining for coreference resolution. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL-2001)*, pages 55–62, Carnegie Mellon University, Pittsburgh PA, June 2001.

[9] N. Immorlica, 2003. Personal Communication.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, pages 282–289, 2001.

[11] A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Knowledge Discovery and Data Mining*, pages 169–178, 2000.

[12] J. F. McCarthy and W. G. Lehnert. Using decision trees for coreference resolution. In *IJCAI*, pages 1050–1055, 1995.

[13] A. E. Monge and C. Elkan. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Research Issues on Data Mining and Knowledge Discovery*, 1997.

[14] T. Morton. Coreference for NLP applications. In *Proceedings ACL*, 1997.

[15] V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In *Fortieth Anniversary Meeting of the Association for Computational Linguistics (ACL-02)*, 2002.

[16] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *Advances in Neural Information Processing (NIPS)*, 2003.

[17] D. Pinto, A. McCallum, X. Lee, and W. B. Croft. Table extraction using conditional random fields. In *Proceedings of the 26th ACM SIGIR*, 2003.

[18] F. Sha and F. Pereira. Shallow parsing with conditional random fields. Technical Report CIS TR MS-CIS-02-35, University of Pennsylvania, 2003.

[19] W. M. Soon, H. T. Ng, and D. C. Y. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[20] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, 2002.

[21] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research (submitted)*, 2003.