An Entity Based Model for Coreference Resolution

Michael Wick [#1], Aron Culotta [*2], Khashayar Rohanimanesh [#3], Andrew McCallum [#4]

[#] *Computer Science Department, University of Massachusetts, Amherst*
Amherst, MA, United States of America
[1] mwick@cs.umass.edu
[2] culotta@cs.umass.edu
[3] khash@cs.umass.edu
[4] mccallum@cs.umass.edu

May 26, 2009

Abstract

Recently, many advanced machine learning approaches have been proposed for coreference resolution; however, all of the discriminatively-trained models reason over *mentions* rather than *entities*. That is, they do not explicitly contain variables indicating the "canonical" values for each attribute of an entity (e.g., name, venue, title, etc.). This *canonicalization* step is typically implemented as a post-processing routine to coreference resolution prior to adding the extracted entity to a database. In this paper, we propose a discriminatively-trained model that jointly performs coreference resolution and canonicalization, enabling features over hypothesized entities. We validate our approach on two different coreference problems: newswire anaphora resolution and research paper citation matching, demonstrating improvements in both tasks and achieving an error reduction of up to 62% when compared to a method that reasons about mentions only.

1 Introduction

Coreference resolution is the problem of clustering mentions (or records) into sets referring to the same underlying entity (e.g., person, places, organizations). Over the past several years, increasingly powerful supervised machine learning techniques have been developed to solve this problem. Initial solutions treated it as a set of independent binary classifications, one for each pair of mentions [1, 2]. Next, relational probability models were developed to capture the dependency between each of these classifications [3, 4]; however the parameterization of these methods still consists of features over pairs of mentions. Finally, methods have been developed to enable arbitrary features over entire clusters of mentions [5, 6, 7].

With few exceptions (e.g., [5]), all of the coreference systems above reason about *mentions*, not *entities*. That is, an entity is defined as simply a concatenation of its mentions. In this paper, we propose a coreference system that explicitly models the attributes of the underlying entity.

Entity A	
Attribute	**Value**
Type	Person
First Name	Stephen
Last Name	Harper
Title	Prime Minister
Country	U.S.
Gender	Female

Figure 1: An entity represented by a canonical record

Modeling these attribute variables allows the coreference system to harness information about the compatibility of an entity as a whole, rather than the sum of its parts. Consider the entity in Figure 1. Even without knowing anything about Prime Minister Harper, it is clear that this entity is not cohesive. For example, the United States has the office of President—not Prime Minister; furthermore, *S*tephen is more likely to be a male name than a female name. Exploiting these dependencies between entity attributes allows the model to better understand the cohesiveness of an underlying coreference cluster.

Given a set of coreferent mentions, we use the term *canonicalization* to refer to the process of generating a standardized representation of the referent entity. The canonical entity should contain any relevant information present in each of the mentions while avoiding artifacts introduced by extraction or coreference. Ultimately, the canonical entity should be robust to outliers since it represents the commonality shared between the individual mentions.

Typically, canonicalization is performed as a post-

name
reginald smith
R. Smith
R. Smith
reggie smith
R. Smith
R.B. Smith
Reginald Smiht
Reginald B. Smith
canonical name: Reginald B. Smith

Table 1: The name attribute for a set of coreferent mentions and the resulting canonical attribute value

processing step to coreference, just before placing an extracted entity into a database. This is unfortunate since these records are densely packed with information and contain few errors, making them valuable pieces of evidence for coreference. In fact, we would like to explicitly use the canonicalization process to construct our entity-level representations.

As another example of how the canonical entity can assist coreference, consider the list of names in Table 1; all the names refer to some real-world entity *Reginald Smith*. The list of extracted names includes lowercase strings, abbreviations of the first name, and typos. The most frequent representation is R. Smith, which unfortunately lacks important information about the first and middle name.

Imagine trying to link this set of mentions with an "R. Smith" entity from another database. Since R. Smith is the most frequently occurring representation, it is likely to be the dominant piece of evidence in the feature computations. Ideally, we would like to know that Reginald B. Smith is the canonical name, which would allow us the freedom to place more emphasis on features involving that particular representation. Indeed, as we demonstrate in Section 7 we find that allowing our model to learn a different set of parameters for a canonical representations yields substantial performance improvements.

The motivation for this paper is that by more closely integrating coreference and canonicalization, we may be able to reduce coreference errors. By performing this integration, we can enable the coreference system to reason about entities (generated by canonicalization), rather than simply sets of mentions.

To this end, we present a discriminative model that jointly predicts coreference and canonicalization. Since coreference is a broad set of problems that encompasses everything from web-people disambiguation to anaphora resolution, we evaluate our approach on two variations of the task: newswire coreference and citation matching. We are able demonstrate performance improvements in both domains, particularly in citation matching where we achieve 62% reduction in coreference error.

The remainder of this paper is organized as follows: In the sequel we present a formal definition of the canonicalization problem and present a solution based on string edit-distance. Then, in Sections 3 - 5 we present a joint model for coreference and canonicalization and introduce approximate learning and prediction algorithms. We also present empirical comparisons on the CORA citation matching dataset and supplement this with additional experiments on people entities from the ACE anaphora resolution corpus. In Section 11 we provide a further discussion of related work. Finally, we discuss our results and suggest directions for future work.

2 Canonicalization by String Edit-Distance

In this section, we formalize the canonicalization problem and present a solution based on string edit-distance.

Given a collection of citation mentions (or newswire documents annotated with a set of entity mentions) $\mathbf{m} = \{m_1 \dots m_n\}$, coreference resolution is the problem of clustering $\mathbf{m}$ into sets of mentions that all refer to the same underlying object (e.g., research paper in the citation or ACE entity in the newswire case). Let $\mathbf{m}^j = \{m_i \dots m_k\}$ be a set of coreferent mentions, where each mention has a set of attribute-value pairs $\{\langle a_1, v_1\rangle \dots \langle a_p, v_p\rangle\}$. Canonicalization is the task of constructing a representative set of attributes for $\mathbf{m}^j$. For example, suppose we have discovered the following three coreferent mentions:

first	last	title
Bill	Clinton	president
William		President of USA
William	Clinton	

We may want to generate the following canonical entity:

first	last	title
William	Clinton	President of USA

Often, canonicalization is performed upon placing an entity into a relational database, either for further processing or browsing by a user. Therefore, canonicalization should create a set of attributes that are both complete and accurate. Efficiency is another motivation for canonicalization — it may be infeasible to store and reason about all mentions to each entity in the database.

In many databases systems, canonicalization is enforced manually with a set of rules, a tedious and error-prone process. However, simple automated solutions are often insufficient. For example, one can simply return the most common or longest string for each attribute value, but noise in automatically extracted values and biases in the frequency of mention strings can lead to unexpected errors. For example, the abbreviated name (R. Smith) may be most frequent.

In this paper, we perform automatic canonicalization by using a tunable string edit-distance between attribute values to find strings for each attribute with the least distance to other values in the cluster. This differs from the system presented in Culotta et al. [8]: our system finds canonical values for each attribute separately, whereas in [8] the canonical entity must match exactly one of the existing mentions.

Let $D : v_i \times v_j \mapsto \mathcal{R}^+$ be the string edit distance between two attribute values. Given a set of coreferent mentions $\mathbf{m}^j$, we define the average edit distance of attribute value v_i as

$$A(v_i) = \frac{\sum_{v_k \in \mathbf{m}^j} D(v_i, v_k)}{|\mathbf{m}^j|} \tag{2.1}$$

Given this metric, we determine the centroid of the set of attribute values and select it as the canonical string. Intuitively, this is the string with the minimum average distance to every other string in the set and therefore engenders the commonality amongst the strings. To construct an entire canonical entity record, we select a canonical value for each attribute and combine them. In this way, it is possible (and likely) that a canonical record contains attribute values from many records and not just one.

For D, we use the well-known Levenshtein distance: a weighted sum of the number of character insertions, deletions, and replacements required to transform one string into another [9]. The recursive definition of the Levenshtein distance for strings s^n and t^m with length n and m is the following:

$$D(s^n, t^m) = \min \begin{cases} c_r(s_n, t_m) + D(s^{n-1}, t^{m-1}) \\ c_i + D(s^{n-1}, t^m) \\ c_d + D(s^n, t^{m-1}) \end{cases} \tag{2.2}$$

where $c_r(s_n, t_m)$ is the *replacement cost* for swapping character s_n with character t_m, c_i is the *insertion cost*, and c_d is the *deletion cost*. We can further define the replacement cost as

$$c_r(s_n, t_m) = \begin{cases} c_r^{\neq} & \text{if } s_n \neq t_m \\ c_r^{=} & \text{if } s_n = t_m \end{cases} \tag{2.3}$$

That is, $c_r^{\neq}$ is the cost of replacing one character with another, and $c_r^{=}$ is the cost of copying a character from one string to the next. We refer to $c_r^{\neq}$ as the *substitution cost*, and $c_r^{=}$ as the *copy cost*.

As the value of the edit distance costs greatly effects the output of the system. For example, if c_i is small, then abbreviated strings will have a small distance to their expanded version. Abbreviated strings will therefore have lower values of $A(v_i)$.

When labeled data is available, rather than requiring the user to manually tune these costs, we set their values automatically to maximize performance on a labeled training set. We discuss this further in Section 5. However, with no labeled data, the small number (four) of parameters can be tuned by hand and also the default parameters (insert=1, delete=1,modify=1,copy=0) yield reasonable results.

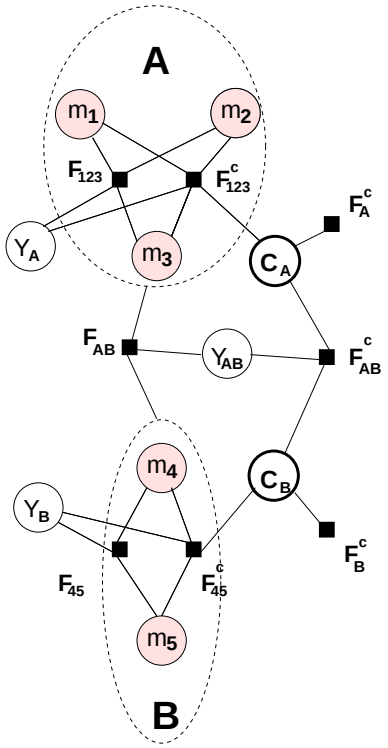

Figure 2: Factor graph for the joint coreference and canonicalization model. This example shows two clusters indicated with dotted lines. Cluster A has three observed mentions $\{m_1, m_2, m_3\}$ and cluster B has two observed mentions $\{m_4, m_5\}$. The unshaded Y nodes represent unobserved binary coreference variables. There are also unobserved entity variables, one per cluster (C_A, C_B).

3 A Joint Model of Coreference and Canonicalization

In this section, we present a graphical model to combine canonicalization and coreference resolution.

Let $\mathbf{m} = \{m_1 \ldots m_n\}$ be a vector of entity mention variables and let I be an element of the *index set* of $\mathbf{m}$ (the set of all indices of $\mathbf{m}$). We define a set of binary random variables $\mathbf{y} = \{\ldots y_I \ldots\}$, where y_I indicates whether the set of mentions referenced by I are coreferent. In Figure 2, there are two variables of this type, Y_A and Y_B, corresponding to the two clusters. There is also another type of coreference variable that represents whether mentions across two different clusters are coreferent, for example Y_{AB} is *o*ne if clusters A and B refer to the same entity and *z*ero otherwise.

To model canonicalization, we also introduce a vector of attribute variables $\mathbf{c} = \{c_1 \dots c_k\}$. Each cluster (mention set) has an associated attribute variable c_i, an assignment to which indicates the canonical attributes for that cluster. For example, an assignment to c_i may be *first name*=Reggie, *last name*=Smith.

To learn a predictive model of the $\mathbf{y}$ and $\mathbf{c}$ variables, we define a conditional random field [10, 11]. We introduce two types of *factors* (or, *compatibility functions*). Factors $f_I(y_I, c_I, m_I, \Lambda)$ are *coreference factors* that return a positive real-valued number indicating the compatibility of an assignment to coreference variable y_I and the set of attribute variables c_I. These functions are parameterized by Λ, a vector of real-valued weights of the CRF. We use the standard log-linear form of these factor functions:

$$f_I(y_I, c_I, m_I, \Lambda) = \exp\left(\sum_k \lambda_k \phi_k(y_I, c_I, m_I)\right) \tag{3.4}$$

where $\lambda_k \in \Lambda$, and ϕ_k is a positive real-valued *feature function* characterizing its arguments.

There are coreference factors that correspond to a single cluster (for example F_{123} in Figure 2), which represent the compatibility of the mentions in that cluster, but there are also factors that represent the compatibility across two clusters (for example F_{AB}). Intuitively, a highly probable coreference clustering has the property that factors between clusters have low affinity scores, while factors over each cluster have high affinity scores. Additionally, there are factors that incorporate information from the canonical variables. These are discussed in more detail below.

Factors $\mathbf{f}^c = \{f_1^c \dots f_k^c\}$ are *canonicalization factors* indicating the compatibility of an assignment to the attribute variables. To determine the canonical form of a cluster, it is necessary to examine the attributes of all mentions in that cluster. Let l(j) by the set of mentions in cluster j, then the canonicalization factors take the form $f_j^c(c_j, \mathbf{m}, l(j), \Theta)$, where these factors are parameterized by Θ.

For the canonicalization method described in Section 2, Θ corresponds to the string edit-distance costs, and f^c corresponds to the (inverse) of the average edit distance. These canonicalization factors are defined analogously to the coreference factors in Equation 3.4:

$$f_j^c(y_j, \mathbf{m}, c_{I(j)}, \Theta) = \exp\left(\sum_k \theta_k \psi_k(y_j, \mathbf{m}, c_{I(j)})\right) \tag{3.5}$$

Since these factors include functions of the canonical entity variables, they enable features that measure the cohesiveness of an entity's attributes (F_A^c, in Figure 2) as well as its compatibility with other entities (e.g., F_{AB}^c). Additionally, entity variables can be compared to the observed mentions for further expressive power (e.g., F_{123}^c).

Algorithm 1 Prediction Algorithm

```
Input:
  Observed mentions m,
  Learned parameters (Λ, Θ)
Initialize m to singleton clusters
while not converged do
  for all Pairs of clusters ⟨m_I, m_J⟩ do
    Merge m_I, m_J
    Perform canonicalization to generate canonical attributes for m_I, m_J
    Score this new assignment with Equation 3.6
  end for
  Merge clusters with highest score
  Canonicalize the newly created cluster
end while
```

Given the coreference and canonicalization factord, we can now define the conditional distribution over $\mathbf{y}$ and $\mathbf{c}$:

$$p(\mathbf{y}, \mathbf{c}|\mathbf{m}; \Lambda, \Theta) \propto \prod_{I \subseteq \mathcal{P}(\mathbf{m})} f_I(\mathbf{y}_I, c_I, m_I, \Lambda) \prod_{j=1}^{n} f_j^c(c_j, \mathbf{m}, I(j), \Theta) \tag{3.6}$$

where the product of factor functions can be converted into probabilities by summing over all assignments to $\mathbf{y}$ and $\mathbf{c}$.

Figure 2 displays the *factor graph* for this CRF, where shaded circles are observed variables, unshaded circles are predicted variables, and black boxes are factors. Edges connect each factor to its variable arguments.

Observe that even with only a handful of entity mentions, the corresponding graphical model is quite complex. Indeed, the connectivity of the graph and the high-arity of the factors makes exact inference intractable for real-world datasets. In the following sections, we describe approximate learning and prediction methods for this model.

4 Prediction

Given parameters (Λ, Θ) and a set of mentions $\mathbf{m}$, prediction is the problem of finding the assignment to $\mathbf{c}$ and $\mathbf{a}$ that maximizes Equation 3.6. Previous work in coreference has demonstrated that graph partitioning and agglomerative clustering algorithms have provided reasonable approximations [1, 12]. We therefore extend an agglomerative clustering algorithm to jointly perform coreference and canonicalization.

The algorithm proceeds by alternating between coreference predictions and canonicalization predictions. The mentions are initialized to singleton clusters. A coreference step is made by scoring all possible merges of existing clusters by Equation 3.6. To compute the model score for each cluster, the canonical attributes $\mathbf{a}$ are predicted using the learned canonicalizer. This enables the model score to account for

the canonical attributes that would be generated by this step.

After the highest scoring coreference step is chosen, the canonical attributes for the newly formed clusters are generated as described in Section 2. The entire prediction algorithm is summarized in Algorithm 1.

5 Parameter Estimation

Parameter estimation is the problem of setting (Λ, Θ). Since prediction is typically a subroutine of parameter estimation algorithms, exact solutions to this problem are also intractable.

Our approximation is the following: First, we optimize Θ (the edit-distance costs) on data labeled for both coreference and canonicalization. (Note that the "label" for canonicalization is the set of attributes that should be selected from a set of coreferent mentions.) We perform exhaustive search over a fixed set of real-valued settings of Θ to find the setting that maximizes canonicalization accuracy.

Second, given the learned Θ, we set Λ (the coreference parameters) by sampling pairs of (possibly incomplete) entities $(\mathbf{m}^i, \mathbf{m}^j)$ and training a logistic-regression classifier to predict whether the two entities should be merged into a single one. The features used in this classifier include both traditional coreference features (e.g., string match, syntactic information), as well as the canonicalization features generated by running the canonicalizer with the weights learned from the previous step.

6 Citation Matching Experiments

6.1 CORA Dataset For our citation matching experiments, we use the CORA corpus, a collection of research paper citations and authors, to evaluate our approach. The corpus contains 1295 citations referring to 134 different research papers for an average cluster size of roughly ten citations per cluster.

We focus our experiments on the citation matching task using the following attributes of a citation:

- venue
- publication date
- publisher
- publication title
- volume
- page numbers

The attribute values in CORA are imperfect and contain a variety of errors including human-introduced typos, as well as extraction errors from automated segmentation algorithms. For example, a researcher's name may be incorrectly segmented and become part of the title instead of being contained in the list of authors.

Furthermore, the citations were originally created by different authors and come from a variety publication venues with different citation formats. For example, page numbers may be written "pp 22-33" or "pages 22-33", and dates: "2003" or "jan 03" or "01/03". Additionally, there are a wide variety of ways to include information about the venue. Some citations contain "in the proceedings of the 23rd..." or "in proc. of twenty-third annual..." while others may omit that information entirely and choose not to include the annual conference number.

These various sources of error and heterogeneity make CORA an ideal and realistic testing-ground for canonizalwise coreference.

6.2 Coreference Features We use first order logic quantified features similar to recent coreference systems [7, 13]. Comparisons for each citation pair in a cluster are aggregated to produce features over entire clusters. The comparisons (or extractors) can be catagorized as either real-valued or boolean-valued.

Pairwise boolean extractors are aggregated over a cluster in the following ways:

- **forall** $\forall$ quantifier in first order logic
- **exists** $\exists$ quantifier in first order logic
- **average** average number of times the feature is on
- **majority** true if the feature is true for a majority of pairs in the cluster
- **minority** true if the feature is inactive for a minority of pairs in the cluster
- **bias** true if the pairwise extractor is relevant for a mention (for example, not all citations have volume numbers, rendering a pairwise comparison of "does volume numbers match" irrelevant)

An exhaustive list of our boolean-valued pairwise extractors is:

- **title strings match** checks if two titles are string identical
- **publication date match** checks if two publication dates are the same
- **venue match** checks if venue names are string identical
- **author list** checks if the list of authors are string identical
- **page numbers** checks if the page numbers are the same
- **volume** whether the two citations came from the same volume
- **publisher** whether two citations have the same publisher

In addition, we include real-valued features that are aggregations of comparisons between two citations. The aggregations for real-valued extractors are:

- **average** the average of all pairwise comparisons
- **max** the maximum value encountered in a cluster
- **min** the minimum value encountered in a cluster
- **bins** the above real-valued aggregations placed in bins

The types of real-valued extractors we use are cosine distance between tokens in an attribute string. These include:

- **TFIDF** cosine distance between two title strings
- **TFIDF** cosine distance between author list strings
- **TFIDF** cos distance between publication venue strings

Finally, we include features involving the canonical entities. The set of features we use for entities is identical to the set of features for mentions, only they are applied to entity records instead of mention records. For example, we may wish to compute the TFIDF token distance between two canonical title attributes taken from different entities (to determine if they are in fact the same entity).

More specifically, we include the following types of entity features:

- **entity to entity comparisons** the entire set of features applied to attribute strings from two entities. No aggregation occurs because there is only one canonical entity for each cluster and thus only one pairwise comparison.
- **entity to mention comparisons** feature extraction occurs between strings of an entity and strings in all the mentions in a cluster. Aggregation is possible (and used as described above).

6.3 Systems In this section we describe two coreference systems. The first system is a first-order coreference model over sets of mentions. This system includes features over entire clusters of citations by using different aggregations of pairwise comparisons between mentions as described in Section 6.2. Note, that this system does not include an explicit representation of an entity record.

The second system, which is the one we propose in this paper, is able to reason on the entity-level by performing coreference and canonicalization jointly. This enriched model enables features that examines the canonical records of entities. In our implementation, canonicalization is handled by computing the centroid of each attribute collection. This centroid-based approach was described in Section 2 and relies on a Levenshtein distance between string pairs. Recall that there are four costs associated with this function (insert, delete, substitute, and copy). We do not learn these parameters on this particular data corpus because we lack ground truth canonicalization labels; rather, we set them to the default values of (insert=1, delete=1, substitute=1, copy=0), which we have found to be reasonable in practice.

	System	Prec	Recall	F1
BCubed	coref+canon	**94.5**	**94.9**	**94.7**
	coref only	93.3	85.7	89.3
Pair F1	coref+canon	**95.7**	**93.6**	**94.7**
	coref only	93.0	79.7	85.8
MUC	coref+canon	98.0	**98.6**	**98.3**
	coref only	**98.1**	96.6	97.3

Table 2: Citation matching results on the CORA dataset. data.

7 Coreference Results (CORA)

We compare the centroid-based joint canonicalization-coreference model to the baseline of a cascaded approach (coreference followed by canonicalization).

For our experiments we performed three-fold cross validation using the same splits provided by Poon and Domingos [13]. We evaluate our systems using precision, recall and F1 according to three evaluation schemes: B-Cubed [14], pairwise comparisons, and MUC [15]. We report multiple evaluation metrics because each has its own set of advantages and disadvantages. For example, MUC and pairwise do not reward the system for correctly predicting singletons (entities with only one referring mention) while B-CUBED does. The results are summarized in Table 6.

The joint approach with centroid canonicalization achieves the highest F1 in all three evaluation metrics, particularly with the pairwise and B-Cubed measure. We notice a substantial boost in recall, suggesting that the canonicalization features are information-rich sources of evidence for coreference.

The current state-of-the-art model by Poon and Domingos [13] on the CORA dataset jointly models segmentation and coreference achieving 95.6% pairwise F1, which is only slightly higher than our 94.7%. We are very happy to see that we are competitive with this system since we are not explicitly modeling and correcting a particular source of error, rather we are mitigating the effect through canonicalization. We believe that canonicalization is more practical, because in general, errors in the data can arise from any number of sources, not just segmentation. We show in the next sections how our model is able to obtain improvements on the ACE data, which contains perfect segmentation, but contains more subtle errors derived from extraction heuristics.

8 Canonicalization Results (CORA)

Although we have no data on which to evaluate the performance of canonicalization, we do provide an example of an attribute that is correctly canonicalized by our system. The following are actual venue strings taken from citations

in a predicted coreference cluster. The canonical form as chosen by the centroid method is shown at the very bottom of Table 3.

Cluster of Venue Attribute Strings
1. in proceedings of the 21th acm symp. on theory of computing,
2. in acm symposium on the theory of computing,
3. in proceedings of the twenty first annual acm symposium on theory of computing,
4. in proceedings of 21th annual acm sympposium on theory of computing.
5. proceedings of the twenty-first annual acm symposium on theory of computing, acm,
6. in proc. 21st ann. acm symp. on theoretical computing
Canonical Venue String
in proceedings of the twenty first annual acm symposium on theory of computing,

Table 3: Example output of our canonicalization algorithm on the CORA dataset

In this example, no string is repeated twice so that simply picking the string that is most commonly occurring would require breaking a six-way tie. The centroid based approach correctly picks a string with no abbreviations and that contains all the relevant information about the publication venue (e.g., it is in the conference proceedings, it is the twenty first symposium and occurs yearly, it is associated with the ACM, and finally, it is on the "theory of computing").

9 Anaphora Resolution Experiments

In addition to citation matching experiments, we also test our methods on the Automatic Content Extraction 2005 (ACE) corpus. The ACE 2005 corpus is a collection of heterogeneous newswire documents taken from broadcast news, transcribed talk shows, radio, and newspaper articles. ACE contains several different entity types including: people (Bill Clinton), organizations (IBM), locations (northern Iraq), weapons (missiles), vehicles (trucks, busses), and geopolitical entities (the United Nations).

For these experiments we focus exclusively on the person entity type since it (1) contains a wide variety of interesting attributes that enable rich canonical entities, (2) is a particularly challenging entity type to resolve, (3) is commonly occurring throughout the corpus and contains many linking pronouns and common nouns.

9.1 Extracted Attributes Using heuristics, we extract the following attributes from the mention texts: first name, middle name, last name, title, gender and nationality.

Even though newswire documents are generally edited and error-free, the heuristics used for extracting attributes are imperfect and consequently these extractions may contain a variety of errors including those described in Table 5. An example of information that we might extract from a mention is shown below:

ACE mention text: U.S. President Bill Clinton
Extracted attributes:
- **Nationality:** U.S.
- **Title:** President
- **First Name:** Bill
- **Last Name:** Clinton
- **Gender:** male

9.2 Features for ACE As in the citation matching problem, we apply first order logic features to the anaphora resolution problem. We aggregate the following types of pairwise comparisons:

	System	Prec	Recall	F1
Pair F1	coref+canon	95.7	93.6	94.7
	coref only	93.0	79.7	85.8
	corf+segment (Poon)	97.0	94.3	95.6

Table 4: Citation matching results comparison with recent work.

Error Type	Example
missing tokens	leaving nationality blank because no nationality information was mentioned in the same sentence
extraneous tokens	mistakenly including a first name in the title, as in title="Senator John"
mis-recognized	mistaking someone's title for their first name as attributes in: first name = "Senator" last name="Kerry"
mis-recognized	mistaking a gender-ambiguous name as male values or female, e.g., "Robin" or "Clinton"

Table 5: Examples of extraction errors in ACE

• **string comparison features** checks if strings match or mismatch at a token level and lexicalize these tokens.
• **field comparison features** checks whether fields such as title are the same or not
• **intervening words** if words are in the same or adjacent sentences, then add all the words that occur between them
• **apposition** checks if mentions A and B are in apposition
• **sentence position** notes whether one of the mentions being compared is the first mention in that sentence.
• **sentence distance** whether mentions are in adjoining sentences, the same sentence, or certain thresholds apart.

All of the above features are aggregated over the pairs of mentions in the cluster by quantifying them (universally and existentially), as well as determining if the features are active for a majority of the pairs. Additionally, the minimum and maximum values are used for real-valued features, for example: *the minimum sentence distance between two mentions is greater than three*.

In addition to these pairwise aggregations, we also include features that summarize the mentions in the cluster:

• Cluster contains does/does not contain pronouns
• Cluster contains does/does not contain proper nouns
• Percentage of pronouns/proper nouns in cluster
• Percentage of mentions with title, first name, last name etc.

Finally, we include features involving the canonical entities. The following lists three types of such features and provides examples for each:

- **features of canonical entity:** these features only examine a single canonical entity at a time. for example, it is more likely that a person's title is 'President' than 'Prime Minister' if their nationality is American. Other features check if the first/last/middle name or nationality occurs in the canonical text of the mention.
- **features between entities:** these features compare two canonical entities and examine specifically string comparisons (string-identical and substring matches) of canonicalized number, gender, nationality, title, and text.
- **features between entities and mentions:** these features compare the records of individual mentions to the canonicalized records of the entire group. These include the number of matches and mismatches between the canonical field value and the mention field values.

Additionally, we suply a set of filters that limit the domains of the first-order logic quantifiers and aggregaters. For example, these filters allow for features such as "for all mention pairs in the cluster that are in the same sentence implies the gender matches". A list of such filters that limit the domain of quantifiers include:

• Both mentions are in the same sentence
• Both mentions are proper names
• First mention is a proper name, second is a pronoun
• Mentions are no farther than two sentences apart

9.3 ACE Coreference Systems In this section we describe three coreference systems for the ACE corpus, all of which produce canonical entities for each cluster. We also include a system that uses a naive canonicalization method to explore the consequences of picking poor representations for entities. Two of the systems approximate the canonicalization factors in our model (see Figure 2) effectively performing the two tasks jointly. The third ignores these factors and performs canonicalization as a post-processing step to coreference.

The first system, which is the model proposed in this work, uses a canonicalization method that selects each field separately, *constructing* the entity from multiple mentions. Two strategies are used to select the fields for this record: a centroid-based approach for string-valued fields (Section 2), and a voted approach for fields with finite domains. For example, gender can only take on the values of male, female or neuter and so the canonical gender just the most frequently occurring one.

The centroid based approach was described in Section 2 and relies on a Levenshtein distance between string pairs. Recall that there are four costs associated with this function (insert, delete, substitute, and copy). We automatically generate training data to learn these parameters from the ground truth coreference labels. This is accomplished by treating the text from the first mention in each gold standard coreference chain as the canonical value for that cluster. Even though the parameters are learned for the"text" attribute, they are applied to the other attributes including "title" and "nationality".

10 ACE Results

We compare the centroid-based entity generation approach to two baselines: a coreference system that is completely devoid of canonical entities, and one that selects them with a heuristic. The three systems are described below:

- **Centroid-based canonicalization and coreference (coref + cent):** jointly solves coreference and canonicalization by approximating canonicalization factors with a centroid model. Advantages of this model are

that it learns parameters from training data and constructs the canonical entity one field at a time.

- **Heuristic joint canonicalization and coreference (coref + heur):** another joint approach that reasons about the two tasks simultaneously. This model, however, deterministically selects the first mention in the coreference chain and designates it as the canonical mention. This is in contrast to the centroid based system which actually assembles the entity from multiple mentions.

- **Pipelined coreference and canonicalization (coref only):** this approach first performs coreference and then canonicalization as a post processing step. This model ignores the entity factors when making coreference decisions and is similar to previous approaches.

We randomly split the ACE documents into a training set of 336 documents and an evaluation set of 114. We evaluate our systems using precision, recall and F1 according to three evaluation schemes: B-Cubed [14], pairwise comparisons, and MUC [15]. The results are summarized in Table 6.

The joint approach with centroid canonicalization achieves the highest F1 in all three evaluation metrics, particularly the pairwise measure. The joint heuristic approach only slightly outperforms the pipeline system in pairwise F1, and is worse than the more sophisticated centroid method.

Both canonicalization methods yield a boost in recall rather than precision. One explanation is that the canonical entities may help to reduce the impact of errors. For example, a single incorrect mention may be enough to prevent other correct mentions from being incorporated into the cluster. However, that error will not contribute much to the canonical entity and so that mention has less of a bearing on the compatibility score with other clusters.

An actual example of how centroid-based canonicalization is improving recall is shown in in Figure 3. Notice how similar the three entities are in the center, a strong indication that there is one entity and not two. The pipeline system incorrectly predicts two entities because it does not take this similarity into account. Relying on mention-wise features is simply not enough: there are only a total of six pairwise comparisons that can be made between these clusters. Out of these six comparisons there is only one gender match (between He and Larry) and only two last name matches.

Unfortunately, the centroid system does not improve precision on the ACE data as it did in CORA; however, the heuristic canonicalizer actually harms it, consistently achieving the lowest score. Since all the fields of the canonical entity are derived from a single mention, a poor mention choice could be devastating. Consider the previous scenario of a cluster with a single error, but imagine that the errorful mention is chosen as the canonical entity. Such a situation would allow that mention to have too much influence on

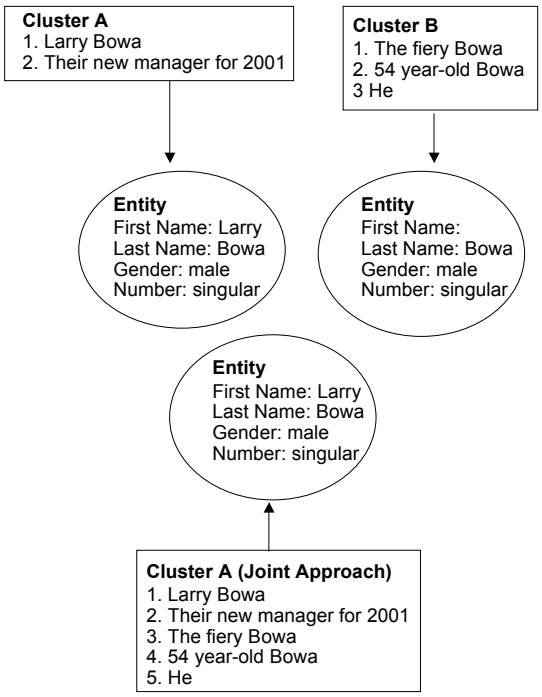

Figure 3: Shown in boxes are clusters of mentions and in circles are the corresponding canonical entities that have been generated from the clusters with a centroid based approach. The top two clusters were generated by the system that ignores the entity factors and erroneously placed them in separate clusters. The bottom cluster was generated by a system that jointly models the entities and correctly merged all the mentions into the same cluster. Notice the similarity between the three entities (circles): they are nearly identical.

other coreference decisions. This further supports the conjecture that features involving certain mentions should be weighted differently than identical features extracted from others.

One reason that our canonicalization system does not improve the results over the coreference-only baseline as drastically on the ACE data as on the CORA data is because ACE clusters are typically much smaller. Recall that our method selects the centroid string for each attribute. Since the centroid is defined as the string that is closest to each other string in a cluster, clusters of size two and smaller do not even have a valid centroid and a "best-guess" has to be made. Additionally, small clusters do not contain as many strings to choose amongst, making the canonical representation less accurate than its potential representation on a larger dataset like CORA

We would also like to compare our results to other coreference systems on ACE; however, it is difficult to

	System	Prec	Recall	F1
BCubed	coref+cent	**82.7**	**76.2**	**79.3**
	coref+heur	81.5	75.8	78.5
	coref only	82.5	75.8	78.5
Pair F1	coref+cent	**66.5**	**44.7**	**53.4**
	coref+heur	63.1	44.0	51.9
	coref only	64.9	40.1	50.0
MUC	coref+cent	**75.1**	72.5	**73.8**
	coref+heur	73.9	**72.8**	73.4
	coref only	74.9	70.9	72.8

Table 6: Person coreference results on ACE 2005 data.

compare these results directly since previous work does not break-down the results into different entity types, but report numbers averaged over people, organization, geo-political entities, etc. However, people entities are a particularly challenging and ubiquitous entity in the data corpus and our results for this entity type surpass many previous systems that average over all entity types. Ng and Cardie [16] achieve almost 70% F1 on the ACE corpus in contrast to our 79% on people, and Culotta et al. obtain just under 80%, but this includes non-people entities, such as geo-political, which tend to be easier to resolve. Another factor that makes comparisons difficult is that many systems omit important details such as which splits of the data the results were obtained on.

11 Related Work

Because the reliability of many real-world systems depend on underlying databases, it is important that the information contained in these databases is as complete and accurate as possible. Two important problems in this area are coreference and canonicalization. In this section we discuss previous research in these areas: first we present coreference work, and then we discuss canonicalization.

11.1 Coreference One important data-cleaning problem is coreference, which is the tasks of clustering mentions/records into real-world entities. Their are many variations of the coreference task, including web-people disambiguation [17], anaphora resolution [18], author disambiguation [19], and citation matching. In this paper we focus primarily on the anaphora resolution and citation matching tasks. However, we discuss relevant work from other areas of coreference because the techniques employed are relevant to our task.

Statistical approaches to coreference resolution can be broadly placed into two categories: generative models, which model the joint probability, and discriminative models that model that conditional probability. These models can be either supervised (uses labeled coreference data for learning) or unsupervised (no labeled data is used). Our model falls into the category of discriminative and supervised. We discuss the relevant work below in terms of these categories.

There has been a large body of work on coreference as a task in isolation. In particular, newswire coreference research has largely been promoted via the ACE and MUC corpora. Initial machine learning efforts on these datasets utilize pairwise similarity measure between mentions, limiting the expressiveness of the models [1, 20]. Other work has explored useful features for modeling the problem [21, 22]. We differ from these works in that we are proposing a new type of model that reasons about entities and furthermore, we do not study the problem of coreference in isolation.

Li et al. [23] propose an unupservised generative model that can identify a canonical string for a newswire entity within a particular document. However, since the canonical representation is selected rather than constructed field-by-field, they lose the ability to model dependencies between the attributes of a single canonical representation.

Haghighi and Klein [24] also propose an unsupervised Bayesian model for newswire coreference. In this generative model, each mention is drawn from a latent entity. However, since each attribute is a distribution over words, the model does not produce a single canonical representation for each entity, a vital step that would have to be performed post-hoc to store the entities in a first normal form relational database. In contrast our system produces canonical representations automatically as coreference is performed. Also, our model is discriminatively trained allowing it to capture arbitrary dependencies between features without the addition of extra edges in the graphical sense.

There has also been a line of related work on the *r*ecord linkage coreference problem dating back to the fifties and sixties with work by Newcombe et al. [25, 26, 27]. Citation matching has been studied in recent years with the burgeoning popularity of digital academic libraries. Examples of such research repositories include DBLife , REXA , , and others. Recent probabilistic approaches to citation matching include both directed (usually generative) [5, 28] and undirected (typically discriminative) graphical models [29, 30, 13] which we describe in more detail below.

Pasula et al. [5] and Milch et al. [28] propose Bayesian network based on logical clauses for modeling the citation matching task. The model implicitly represents entities with distributions specific to certain attributes such as title or venue. However, we believe that the flexibility of discriminatively-trained models is an advantage for the coreference tasks since they more naturally handle overlapping and co-dependencies between features. Also, their approaches do not explicitly result in canonical records as ours does.

Hall et al. [31, 32] also propose a directed model, but

for the task of venue coreference. Their approach incorporates distortion models between strings that discovers patterns of heterogeneous representations in a similar spirit to canonicalization. In contrast to previous unsupervised methods, they explicitly model dependencies between their two attributes of interest: venue and titles. Their results reveal that modeling this dependency is important; however, in a directed framework, adding additional dependencies between attributes requires blowing up the model. In the citation matching task, we can have up to a dozen attributes, and modeling all these cross-attribute dependencies begins to become prohibitively expensive. In contrast, because discriminative training methods model the conditional distribution, the complexity of our model stays the same when adding additional cross-field dependencies.

There has also been discriminatively trained methods in undirected graphical models. For example work by Culotta and McCallum [29, 30] describe a conditional random field that incorporates first-order quantified features for the citation matching problem. The work describes a methods for inference in weighted logic models where there are too many clauses to ground the network. Similarly, our model is too large to ground and we must use similar techniques. A major difference in our work is that we explicitly model entities and perform coreference and canonicalization jointly, whereas their work focuses exclusively on coreference in isolation. Poon and Domingos [13] achieve impressive results by jointly modeling citation matching with segmentation. However, their weighted logic model factorizes mention pairs, forcing the model to reason over mentions instead of entities. In contrast, our model allows first order logic features to be expressed over entire clusters, enabling us to model canonicalization and coreference simultaneously.

11.2 Canonicalization Once coereference (or record deduplication) has taken place, a choice may need to be made about which of the many possible records should be chosen to represent the entity in a database. This problem, canonicalization, has been largely under studied by the community, but recently [8] formalize the task and propose three types of solutions. However, that work only demonstrates results for one of the solution types: choosing a single a mention as the canonical entity. In the current work, we select a canonical string for each attribute and assemble them into a completely novel canonical entity that incorporates information from multiple mentions.

Goldberg and Senator [33] outline many of the issues and theoretical concerns with a system that combines coreference and canonicalization, which they refer to as *link formation* and *consolidation*. However, they do not implement or evaluate such a system. Recent work by Wick et al. [34] has demonstrated the advantage of canonicalization and person-coreference in a schema-matching setting, yielding promising results that should be further explored on other coreference tasks.

12 Conclusions and Future Work

We have described a coreference system that reasons about entities rather than just mentions. We motivated an approach based on the idea of canonicalization and demonstrate that jointly performing coreference and canonicalization yields fewer errors than a cascaded system that performs coreference and then canonicalization. Specifically, we modeled the joint problem with a conditional random field. We validated our approach on two different coreference problems: citation matching and anaphora resolution, demonstrating that our method is viable for a wide variety of entity disambiguation problems. On the CORA dataset, we noticed a large reduction in error that is competitive with the state-of-the-art on that corpus. We also achieved noticeable improvements on ACE data when compared with a mention-based system. Upon error analysis, we gathered anecdotal evidence that the improved performance is due to both the model's ability to reason about entities as well as its ability to mitigate error sources.

Additionally, we have applied approximate learning and inference methods that make this model applicable in practice. Despite the approximations, our joint systems outperform a version that does the two tasks independently in a cascade. We believe that future efforts in improving these approximations will lead to further improvements in coreference performance. For example, more advanced canonicalization methods can be developed to infer certain attributes from the others, even if that attribute does not exist among the mentions. Also, an even tighter integration between canonicalization and coreference can be obtained by considering the distribution over canonicalization decisions in each cluster.

We also believe that canonicalization may dampen the effects of outliers in certain problems such as coreference. However, additional experiments and more in depth error analysis need to be conducted to verify the conjecture. It may be worthwhile to explore the concept of canonicalization in other tasks such as record extraction and data mining.

Finally, we believe that canonical-wise coreference decisions potentially scale better than mention-wise decisions since the number of entities is upper bounded by the number of mentions, and in practice, there are much fewer entities than mentions. Future work can investigate fast methods for performing canonicalization that will in turn lead to coreference systems that scale to large amounts of data.

13 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by Lockheed Martin through prime contract No. FA8650-06-C-7605 from the Air Force

Office of Scientific Research, in part by UPenn NSF medium IIS-0803847, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0326249, and by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

[1] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Comput. Linguist.*, vol. 27, no. 4, pp. 521–544, 2001.

[2] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *ACL*, 2002.

[3] A. McCallum and B. Wellner, "Toward conditional models of identity uncertainty with application to proper noun coreference," in *IJCAI Workshop (II Web)*, 2003.

[4] Parag and P. Domingos, "Multi-relational record linkage," in *Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining*, Aug. 2004, pp. 31–48.

[5] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser, "Identity uncertainty and citation matching," in *Advances in Neural Information Processing Systems 15*. MIT Press, 2003.

[6] H. Daumé III and D. Marcu, "A large-scale exploration of effective global features for a joint entity detection and tracking model," in *HLT/EMNLP*, Vancouver, Canada, 2005.

[7] A. Culotta, M. Wick, and A. McCallum, "First-order probabilistic models for coreference resolution," in *HLT/NAACL*, 2007.

[8] A. Culotta, M. Wick, R. Hall, M. Marzilli, and A. McCallum, "Canonicalization of database records using adaptive similarity measures," in *KDD*, San Jose, CA, 2007.

[9] V. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals." *Doklady Akademii Nauk SSR*, vol. 163, no. 4, pp. 845–848, 1965.

[10] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[11] C. Sutton and A. McCallum, "An introduction to conditional random fields for relational learning," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds., 2007.

[12] C. Nicolae and G. Nicolae, "Bestcut: A graph algorithm for coreference resolution," in *EMNLP*, 2006.

[13] H. Poon and P. Domingos, "Joint inference in information extraction," in *AAAI*. Vancouver, Canada: AAAI Press, 2007, pp. 913–918.

[14] B. Amit and B. Baldwin, "Algorithms for scoring coreference chains," in *Proceedings of MUC7*, 1998.

[15] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, "A model-theoretic coreference scoring scheme," in *Proceedings of MUC6*, 1995, pp. 45–52.

[16] V. Ng, "Machine learning for coreference resolution: From local classification to global ranking," in *ACL*, 2005.

[17] J. Artiles, S. Sekine, and J. Gonzalo, "Web people search: results of the first evaluation and the plan for the second," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, 2008.

[18] J. F. McCarthy and W. G. Lehnert, "Using decision trees for coreference resolution," in *IJCAI*, 1995, pp. 1050–1055.

[19] A. Culotta, P. Kanani, R. Hall, M. Wick, and A. McCallum, "Author disambiguation using error-driven machine learning with a ranking loss function," in *(IIWeb-07)*, Vancouver, Canada, 2007.

[20] A. McCallum and B. Wellner, "Conditional models of identity uncertainty with application to noun coreference," in *NIPS17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005.

[21] M. Poesio, D. Day, R. Arstein, J. Duncan, V. Eidelman, C. Giuliano, R. Hall, J. Hitzeman, A. Jern, M. Kabadjov, G. Mann, P. McNamee, A. Moschitti, S. Ponzetto, J. Smith, J. Steinberger, M. Strube, J. Su, Y. Versley, X. Yang, and M. Wick, "Exploiting encyclypedic and lexical resources for entity disambiguation," Johns Hopkings University, Baltimore, Tech. Rep., 2007.

[22] Y. Versley, "Antecedent selection techniques for high-recall coreference resolution," in *EMNLP*, 2007.

[23] X. Li, P. Morie, and D. Roth, "Identification and tracing of ambiguous names: Discriminative and generative approaches," in *AAAI*, 2004, pp. 419–424.

[24] A. Haghighi and D. Klein, "Unsupervised coreference resolution in a nonparametric bayesian model," in *ACL*, 2007.

[25] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. James, "Automatic linkage of vital records," *Science*, vol. 130, pp. 954–9, 1959.

[26] H. B. Newcombe and J. M. Kennedy, "Record linkage: Making maximum use of the discriminating power of identifying information," *Comm. ACM*, vol. 5, pp. 563–566, 1962.

[27] H. B. Newcombe, "Record linkage: The design of efficient systems for linking records into individual and family histories," *Am. J. Human Genetics*, vol. 19, pp. 335–359, 1967.

[28] B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov, "BLOG: Probabilistic models with unknown objects," in *IJCAI*, 2005.

[29] A. Culotta and A. McCallum, "Practical markov logic containing first-order quantifiers with application to identity uncertainty," University of Massachusetts, Tech. Rep. IR-430, 2005.

[30] Aron, A. Culotta, and A. McCallum, "Practical markov logic containing first-order quantifiers with application to identity uncertainty," in *HLT Workshop*, 2006.

[31] Rob, R. Hall, C. Sutton, and A. McCallum, "Unsupervised coreference of publication venues," University of Massachusetts, Amherst, Amherst, MA, Tech. Rep., 2007.

[32] R. Hall, C. Sutton, and A. McCallum, "Unsupervised deduplication using cross-field dependencies," in *KDD*, Las Vegas, Nevada, 2008.

[33] H. G. Goldberg and T. E. Senator, "Restructuring databases for knowledge discovery by consolidation and link formation," in *KDD*, 1995.

[34] M. Wick, K. Rohanimanesh, K. Schultz, and A. McCallum, "A unified approach for schema matching, coreference, and canonicalization," in *KDD*, Las Vegas, Nevada, 2008.