# Coreference Resolution:
## Current Trends and Future Directions

Jonathan H. Clark and José P. González-Brenes

November 24, 2008

**Abstract**

Coreference resolution seeks to find the mentions in text that refer to the same real-world entity. This task has been well-studied in NLP, but until recent years, empirical results have been disappointing. Recent research has greatly improved the state-of-the-art. In this review, we focus on five papers that represent the current state-of-the-art and discuss how they relate to each other and how these advances will influence future work in this area.

# Contents

# 1 Introduction

Coreference resolution is the process in which we identify the noun phrases that are referring to a same real-world **entity** (Ng, 2008). In this context, such noun phrases are called **mentions**, or just **anaphoric noun phrases**. Mentions can be either named, nominal or pronominal (Luo, 2007). For example, table 1 illustrates an example of mentions of the entity "Joe Smith" (Lin, 2008).

Table 1: (Source: ACE Annotation Guidelines for Entities). Example of mentions of the real-world entity "Joe Smith"

| | |
|---|---|
| Name Mention: | Joe Smith, Mr. Smith |
| Nominal Mention: | the guy wearing a blue shirt |
| Pronoun Mentions: | he, him |

Until recently, statistical approaches treated coreference resolution as a binary classification problem, in which the probability of two mentions from the text $i$ and $j$ having a coreferential outcome can be calculated from data by estimating the probability of Denis and Baldridge (2007):

$$P_C(COREF|\langle i,j\rangle) \tag{1}$$

If $\langle i,j\rangle$ is interpreted as an ordered pair, then we are enforcing an **asymmetric interpretation** (Nguyen and Kim, 2008), where $i$ is an antecedent preceding in the text the anaphora $j$. This interpretation of coreference resolution is very similar to **anaphora resolution**, where we try to find the antecedent $i$ of a pronominal $j$. Similarly, if $\langle i,j\rangle$ is interpreted as an unordered pair in which $i$ and $j$ are simply coreference, but no particular direction of anaphora is specified, then we have a **symmetric interpretation**.

It is straightforward to calculate $P_C$ in equation 1 using feature functions using nothing more than the pair $\langle i,j\rangle$. However, **pairwise formulations** like this imply a strong independence assumption that makes impossible to represent features on the entire cluster of mentions that refer to a same entity (Denis and Baldridge, 2007; Culotta et al., 2007). **Cluster-based** features are desired to enforce characteristics of the entity, for instance avoiding having an entity described with only pronominal mentions. It is still an active area of research how to convert the set of classifications of pairwise models into clusters of mentions where each cluster refers to the same entity(Culotta et al., 2007).

Another problem on this formulation is that the identification of anaphoric noun phrases is done as part of the coreference resolution process, and it is possible that an anaphor that is not coreferential with any other mention in the text might be assigned one by the model (Denis and Baldridge, 2007; Luo, 2007).

In this review, we give a detailed discussion of five recent papers that represent recent trends in coreference resolution to address the problems discussed above. We begin by describing the basic techniques employed by each of the papers in Section 2. We discuss the linguistic considerations involved in designing features in Section 3. We then discuss how to train a model that can learn from annotated corpora in Section 4, we particularly focus on how to sample training instances. Having shown how a model can be trained, we present inference mechanisms for coreference with an emphasis on the more difficult case of

unsupervised systems in Section 5. Finally, we offer a few conclusions based on the successes and failures of these systems.

## 2 Summary of Reviewed Papers

In this review, we focus on several recent works that represent the state of the art in coreference resolution. During these past two years, the performance of state-of-the-art systems has increased dramatically, as shown in the 10 point F1 gain in (Culotta et al., 2007) over the previous system on the same data set[1] (Ng, 2005). A few themes that run through this literature are:

1. Models that allow the representation of more complex features, such as cluster-based features and apposition
2. An interest in resolving anaphoricity jointly with coreference
3. Unsupervised methods

These themes will be discussed in their own right in later sections, but first we briefly summarize individually the papers that we will review in this paper.

### 2.1 First-Order Probabilistic Models for Coreference Resolution

Culotta et al. (2007) makes the key observation that traditional noun phrase coreference solution systems represent features only of pairs of noun phrases as shown in their baseline pairwise model: Given a pair of noun phrases $x_{ij} = \{x_i, x_j\}$, let the binary random variable $y_{ij} = 1$ indicate that $x_i$ and $x_j$ are coreferent. Let $F = f_k(x_{ij}, y)$ be a set of features, each with an associated parameter $\lambda_k$. Let $Z_{x_{ij}}$ be a normalizer that sums over the two values of $y_{ij}$. Then Culotta's pairwise model is:

$$p(y_{ij}|x_{ij}) = \frac{1}{Z_{x_{ij}}} \exp \sum_k \lambda_k f_k(x_{ij}, y_{ij}) \tag{2}$$

The paper then goes on to describe a model that softens this independence assumption by allowing features based on first-order logic. The major change in this model is that we are now given a *set* of noun phrases $x^j = \{x_i\}$. Let $y_j$ be true when $\forall i : x_i \in x^j$. $Z_{x^j}$ is a normalizer that sums over the two values of $y_j$. Then Culotta's first-order model is:

$$p(y_j|x^j) = \frac{1}{Z_{x^j}} \exp \sum_k \lambda_k f_k(x^j, y_j) \tag{3}$$

By creating a model over clusters instead of over pairs, the number of $y$ variables grow exponentially in the number of noun phrase of the documents. To overcome this difficulty, they propose sampling methods at training time and do inference using greedy clustering at testing, by incrementally instantiating $y$ variables as needed during prediction.

Training is performed by sampling uniformly from positive and negative examples and using the Margin Infused Relaxed Algorithm (MIRA) to give positive examples a higher

[1] Culotta et al. (2007) do note that the test-train splits may differ slightly

rank than the negative examples so that true coreferent mentions within bad clusters are not unjustly penalized. This rank-based method is used in combination with an error-driven training technique that targets only the negative cluster examples that would have otherwise been created by their clustering procedure.

By using first-order features, Culotta et al allows to model constraints like "do not prefer entities whose only mentions are pronouns". Incidentally, this gives them the power of Markov Logic Networks, which are discussed later in Section 2.4. Culotta also introduces an error-driven, rank-based training technique that targets carefully chosen negative examples when moving the decision boundary to classify positive examples.

### 2.2 Unsupervised Coreference Resolution in a Nonparametric Bayesian Model

Haghighi and Klein (2007) proposed a hierarchical Dirichlet Process to find the referents of mentions within a document. They extend their solution to find coreferents across documents with the entities being shared across the corpus. The number of clusters are determined by the inference (see discussion on inference in section 5.2.3). Their work was the first unsupervised approach to report performance "in the same range" of fully supervised approaches for coreference resolution.

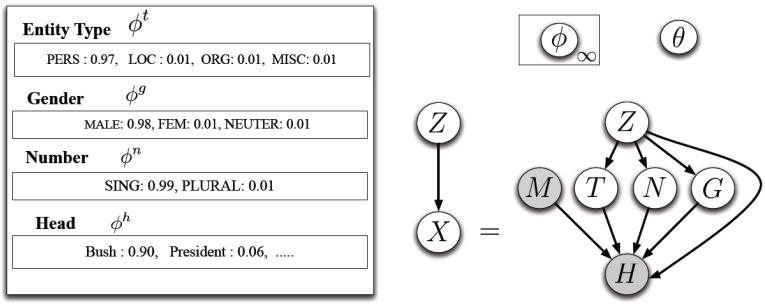


Figure 1: Taken from Haghighi et al: (a) A Haghighi and Klein entity and its features. (b) Graphical model representation of their infinite mixture model. The shaded nodes indicate observed variables.

Figure 1 uses random variable Z to refer to the random variable that takes the value of the the index of an entity. Let $X$ be the collection of variables associated with a mention in the model (namely entity type $T$, number$N$, gender $G$, head $H$, mention $M$).

### 2.3 Unsupervised Models for Coreference Resolution

Ng (2008) conceptualizes the coreference resolution problem as inducing coreference partitions on unlabeled documents, rather than classifying whether mention pairs are coreferent. For this they modify the Expectation-Maximization (EM) algorithm, so that the number of clusters does not have to be predetermined. Instead of initializing the model with a uniform distribution over clusters, the model is initialized with a small amount of labeled data for the first iteration of EM. The E-step is approximated by computing only the conditional probabilities that correspond to the $N$ most likely **clusterings**[2], where $N$ is a parameter

[2]In this context, a clustering is a different way of partitioning of the mentions.

to the algorithm. More precisely, given a document $D$, a clustering $C$, let $\theta$ be the model parameters. The following EM algorithm is used:

**E-step**: Compute the posterior probabilities of the clusterings, $P(C|D,\theta)$ based on the current $\theta$.

**M-step**: Using the value of $P(C|D,\theta)$ computed in the E-step, find the $\theta'$ that maximizes the expected log-likelihood: $\sum_C P(C|D,\theta) log P(D,C|\theta')$.

### 2.4 Joint Unsupervised Coreference Resolution with Markov Logic

Poon and Domingos (2008) present an unsupervised model using **Markov Logic Network** (MLN). MLN is a first-order knowledge base with a weight attached to each formula; if the weight is infinite, then the MLN behaves exactly as first-order logic does. With finite weights when a world violates a formula in a MLN, the world becomes less probable, but not impossible (Richardson and Domingos, 2006). The basic idea in a MLN is to soften the constraints imposed by a set of first-order logic formulas.

Under the hood, MLNs use first-order logic as a language to define a template that will be extended as a Markov network. The Markov network is created with one node per ground atom and one feature per ground clause. This combines first-order logic and probabilistic graphical models into a single representation. For example, table 2 shows a sample MLN (the weights are not shown). Its corresponding graphical model representation is shown in Figure 2.

Table 2: (Source: Richardson and Domingos (2006) ) Example of a MLN

| | |
|---|---|
| Smoking causes cancer: | $\neg Sm(x) \vee Ca(x)$ |
| If two people are friends, either both smoke or neither does: | $\neg Fr(x,y) \vee Sm(x) \vee \neg Sm(y), \neg Fr(x,y) \vee \neg Sm(x) \vee Sm(y)$ |

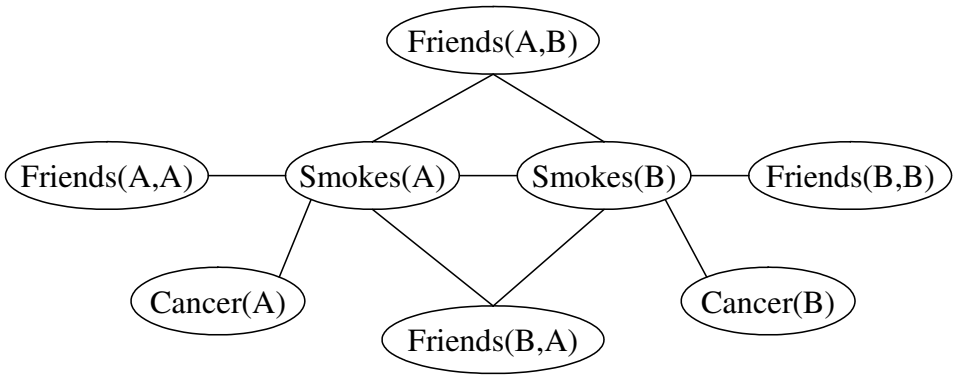


Figure 2: (Source: Richardson and Domingos (2006) )Ground Markov network obtained by applying the formulas in Table 2 to the constants Anna(A) and Bob(B).

As Ng (2008)'s formulation, the MLN approach works unsupervised, and performs comparatively better than Haghighi and Klein's model. It is unfortunate that there is no direct comparison between the Ng (2008) and Poon and Domingos (2008) models. Their improved

performance is achieved because their models allow them to represent more expressive features (such as apposition). see Section 3 for a more thorough discussion on features. Culotta et al claim that their supervised system achieves the representational power of MLN's by defining the features in their log-linear model to be scoped over sets of mentions (Culotta et al., 2007).

### 2.5 Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming

**Integer Linear Programming** (ILP), though typically regarded as an optimization method, has been used as a means of joining together multiple classifiers in coreference (e.g. both a coreference identifier and an anaphoricity identifier) (Denis and Baldridge, 2007). However, ILP can be used to produce a global assignment that maximally agrees with the decisions made a classifier (Denis and Baldridge, 2007):

$$\min \sum_{\langle i,j\rangle \in P} c^C_{\langle i,j\rangle} \cdot x_{\langle i,j\rangle} + \bar{c}^C_{\langle i,j\rangle} \cdot (1 - x_{\langle i,j\rangle}) x_{\langle i,j\rangle} \in \{0,1\}, \forall \langle i,j\rangle \in P \tag{4}$$

where the cost for committing to each coreference link $c^C_{\langle i,j\rangle} = -\log(P_C(\texttt{COREF}|i,j))$ and $\bar{c}^C_{\langle i,j\rangle} = -\log(1 - P_C(\texttt{COREF}|i,j))$ and the $x_{\langle i,j\rangle}$ indicate whether or not the link is selected (Denis and Baldridge, 2007). However, by itself, this simply results in choosing exactly the links with a probability greater than 0.5, making it clear why this method is desirable only when we wish to combine multiple classifiers jointly. It should be noted that this procedure used by Denis et al. becomes intractable unless we use a pairwise independence assumption (Section 3), rendering this technique unable to interface with modern cluster-based classifiers. More attention is given to ILP as a method for joint anaphoricity-coreference resolution in Section 5.1.2.

In this section, we have provided a brief overview of the papers that will be our focus for the remainder of this paper. With these details in mind, we will now discuss the more general themes that run through these papers.

## 3 Features

Many of the recent advances in state-of-the-art coreference resolution systems have come from improvements in the underlying models, that allow to represent linguistically more robust features. As we have described in Section 1, it is possible to categorize coreference models into two categories: (i) pairwise models, which are myopic in that they can only examine pairs of mentions at once, and (ii) cluster-based models, which have access to entire clusters of mentions at once, but present a difficult inference problem. This section will address each of these types of models in turn and will discuss common features used in each of these paradigms.

For a comprehensive study on features commonly used in many coreference resolution systems see Ng and Cardie (2002). We present a summary of their features in Appendix A.

### 3.1 Pairwise Features

In the **pairwise** formulation of coreference, mentions are represented by the vertices in a graph while the edges are weighted by the probability that the two nouns are coreferent

(Figure 3). While this figure indicates entity clusters with dotted circles, these clusters were formed by a greedy graph partitioning algorithm, which has access only to pairs of mentions. That is, even though the result is a partition in which each cluster represents an entity, the features used to create this partition are myopic at the level of mention pairs.

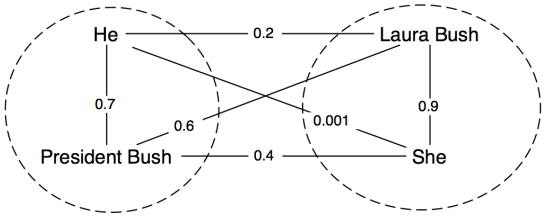


Figure 3: An example noun coreference graph from Culotta et al. (2007) in which vertices are noun phrases and edge weights are proportional to the probability that the two nouns are coreferent. Partitioning such a graph into disjoint clusters corresponds to performing pairwise coreference resolution.

Though these models are attractive due to their low computational cost, they fail to account for several phenomena important to coreference resolution. Primarily, pairwise models cannot enforce the **transitive closure** of a cluster, the property that if $a$ is coreferent with $b$ and $b$ is coreferent with $c$, then $a$ is also coreferent with $c$. Further, pairwise models also have an issue with creating entity clusters consisting solely of pronouns since there is very little evidence to show that they would disagree with each other yet the model provides no way of expressing "entities should not consist of only pronouns."

Another feature that has been shown to be important in coreference is **apposition**. This feature accounts for the observation that mentions in appositives typically refer to the same entity as the mention that the appositive modifies. For example, in "Bill Gates, generous donor to CMU," both "Bill Gates" and "generous donor to CMU" refer to the same entity. According to Poon and Domingos (2008), this feature made a difference of 3.2 MUC F1 on the MUC-6 data set. Ng (2008) also used this feature and reported good results in his overall system.

**Salience** has also gained popularity as a feature in coreference systems. Salience is the standing out of some mentions relative to other neighboring mentions. Haghighi and Klein (2007) model this by incrementing an activity score by 1 each time a mention is found. This activity score decays by a factor of 0.5 each time a new mention is generated. Poon and Domingos (2008) models salience only only pronoun salience, since the saliencies of proper nouns and nominals have only marginal influence. They do this by imposing a prior over distances between mentions rather than the more complicated notion of activation used by Haghighi. Notably, modeling salience is difficult in some formulations of learning algorithms, as is it is the case in Dirichlet Processes.

### 3.2 Cluster-based Features

Given the limitations of pairwise models, recent research has made significant progress by adopting first-order logic features, that allow to enforce the transitive closure for each cluster. By moving out of the pairwise paradigm, we can imagine designing cluster-based features

that allow modeling the properties of several or all of the mentions within an entity cluster at once. For example, Table 3 shows some of the cluster-based features that Culotta et al. implemented.

Cluster-based models typically usually include most of the same features as would be seen in a pairwise model while simply adding features that leverage the stronger representational power of first-order logic. Cluster-based features allow to leverage information about all the mentions in an entity cluster (Poon and Domingos, 2008).

Table 3: (Source: Culotta et al. (2007) ) Example of cluster-based features

| | |
|---|---|
| All-X | True iff all pairs share a feature X |
| Most-True-X | True iff the majority of pairs share a feature X |
| All-True | True all mention pairs are predicted to be coreferent |
| Most-True | True iff most mention pairs are predicted to be coreferent |
| Phrase count | A count of how many phrases in the cluster are of each mention type. By using this in combination with All-True and Most-True, this feature can capture the soft constraint that no cluster consists of only pronouns |

It is important to note that not only the choice of features that leads to improvements in system quality, the way in which those features are implemented can also make a big difference. For instance, Poon and Domingos (2008) showed that by simply using the head percolation rules from the Stanford parser instead of crudely choosing the right-most token in a constituent as in Haghighi and Klein (2007), a performance boost of 3.5 MUC F1 on the MUC-6 data set.

### 3.3 Generative models and their effect on features

In a statistical context, generative models define a joint distribution over features by using a series of conditional distributions that tell a "generative story" of how observables came into being. Many of the papers in this review cast their features in a generative model that conditions on clusters. These papers include Haghighi and Klein (2007), Poon and Domingos (2008), and Ng (2008). Such generative models can alter the way in which distributions over these features would normally behave if they were in a discriminative framework since a distribution over these features is created for each cluster and each of these distributions may become peaked around the gender, type, or number that bets fits in each cluster.

In Figure 4 we present an analysis of a few important features of our focus papers. We did not include features from Denis and Baldridge (2007) since they limit the description of their features as "similar to that used by Ng and Cardie (2002)". **C** indicates that the feature has access to a whole entity cluster at once whereas **P** indicates that the feature has access to only a single pair of mentions at a time. **G** indicates that the feature is inherently pairwise while its inclusion in a generative model allows it to capture some degree of dependence with the clusters that generate the feature.

In this section, we have given an overview of the standard pairwise coreference features that have become the standard for the last several years and discussed cluster-based features that are present in more recent models. While the choice of features and the care with which they are implemented continues to play an important role in the quality of coreference

| | Culotta et al. (2007) | Haghighi and Klein (2007) | Ng (2008) | Poon and Domingos (2008) |
|---|---|---|---|---|
| Salience | | G | | P |
| Apposition | P | | G | P |
| Entity Type Agr. | P, C | G | | C |
| Number Agr. | P, C | G | G | C |
| Gender Agr. | P, C | G | G | C |
| Head word | P, C | G | | C |

Figure 4: Selected features used in our focus papers

systems, recent advances in the state-of-the-art have come primarily from advancements in the underlying models. However, these more complex models present difficulties both in training and inference.

# 4 Parameter Estimation

In any supervised method (and some "unsupervised" methods), one usually obtain estimates of the free parameters of a model before performing inference. The mechanics of parameter optimization follow fairly standard practice. The procedure is left rather vague or mentioned uneventfully simply stating that the maximum entropy principle was employed via some method such as the limited memory variable metric algorithm (L-BGFS) (Denis and Baldridge, 2007). However, as we will see in this section, other training-time considerations such as the way in which training examples are constructed can have a strong impact on the quality of the resulting model.

## 4.1 Supervised Parameter Estimation

Often, the estimation of parameters for supervised log-linear models in coreference papers is left rather vague or mentioned uneventfully simply stating that the maximum entropy principle was employed via some method such as the limited memory variable metric algorithm (L-BGFS) (Denis and Baldridge, 2007). Here, we focus on those parameter estimation methods which deviate from this pattern.

Culotta casts **optimization as a ranking problem** (as opposed to a classification problem) and proposed the use of the **Margin Infused Relaxed Algorithm** (MIRA) due to the move to a cluster-based framework (Culotta et al., 2007). First, consider that a classification-based trainer might unjustly "penalize" all features associated with each incorrect cluster, even though there may be subsets of the cluster that are coreferent. This is then combined with an error-based sampling method (Section 4.2) that calculates the difference between the current weight vector and an improved weight vector via a "nearby" positive example, rather than a random or optimal positive example. This update is then accomplished via MIRA; the update then has two constraints: (i) the positive example must have a higher score by a given margin and (ii) the change to $\Lambda$ should be minimal.

### 4.2 Sampling the corpus

In pairwise models, sampling is needed to reduce the number of pairs artificially constructed from the annotated corpus. For instance, in training these pairwise models, one could imagine generating training instances by enumerating all pairs $\langle x_{ij}, y_{ij} \rangle$ where $y_{ij}$ is true iff $x_i$ and $x_j$ are coreferent. However, this would result in a very unbalanced training set having many more negative than positive examples (Culotta et al., 2007). This imbalance is an inherent flaw of pairwise models of coreference.

A canonical way of resolving this issue is Soon's method (Soon and Ng, 2006): Scan the document from left to right for a noun phrase $x_i$. Upon finding an $x_i$, comparing each noun phrase $x_i$ to each noun phrase $x_j$ where $j < i$. For each pair scanned, create a training instance $\langle x_{ij}, y_{ij} \rangle$ where $y_{ij}$ is true iff $x_i$ and $x_j$ are coreferent. The search for a matching $x_j$ terminates when a positive example is found, or the beginning of the document is reached. The result is that distant NP's are pruned from the training sample. This method is still commonly used (Culotta et al., 2007; Denis and Baldridge, 2007).

It is important to note that this sampling method should complement the clustering method being used. Therefore, the pairwise sampling strategy described above is typically used in combination with greedy clustering in a partitioning scheme that is guided by the word order of the document in the same fashion as the sampling strategy (Soon and Ng, 2006). This also holds true for modern cluster-based methods, as we will discuss below.

For cluster-based models, let us first consider the very simple method of **uniform sampling** that Culotta et al. call First-Order Uniform. This method generates training examples by sampling positive and negative examples uniformly at random from the training set. Positive examples are generated by first sampling a gold entity cluster, then sampling a subset of mentions from that cluster. Negative examples are generated by sampling two correct entities and merging them into one entity cluster (Culotta et al., 2007).

Culotta then goes a step further to propose an error-driven sampling method for online training that produces training examples on-the-fly, based on the mistakes that the model makes. It works by performing greedy agglomerative clustering on a training document $i$ given initial parameters $\Lambda$ until an incorrect cluster is formed. The parameter vector according to this mistake, then repeat for some fixed number of iterations (Culotta et al., 2007).

In this section, we have discussed the ways in which training examples are constructed for coreference systems. As for training methods, we have seen as early as Section 2 that recent cluster-based models use their own custom training procedures, tailored to meet the needs of the expanded model. Being that these models are so novel, no standard train This has a strong impact on both the quality and tractability of the system. Having discussed the training of models, we will next examine how inference can be performed with them.

## 5 Inference

Until recently, inference in coreference resolution has involved a relatively straightforward application of some standard inference mechanism for a well-known model such as a log-linear model. However, the advent of cluster-based and unsupervised models has added a degree of difficulty to this task. In this section, we describe some issues and possible solutions that have come about with these new models.

### 5.1 Supervised methods

#### 5.1.1 Cluster-based inference

When creating a partition over the set of mentions in a document, one typical method that has been used in pairwise models is graph partitioning (McCallum and Wellner, 2005). However, when cluster-based features are applied, it becomes desirable to form clusters at an earlier stage in the inference process. Culotta et al. (2007) performs greedy agglomerative clustering, where the decision of merging is proportional to the probability of the new clustering according to the log-linear model of equation 3. Clustering terminates when there is no additional merge that improves the likelihood of the clustering. Certainly, one could consider other methods of performing clustering, however, if such methods are explored in the future, they will have to address the issue of finding a training procedure that complements the desired method of clustering.

This approach makes exact locally-optimal decisions. However, in the case of the the unsupervised methods that we will discuss make, we will also see that it is possible to make globally-optimal decisions if an approximation is acceptable (though this need not correlate with the method being supervised or unsupervised in general).

#### 5.1.2 Joint inference of anaphoricity and coreference vs pipelining

Many systems view the task of coreference as a **pipeline** in which noun phrases are identified and then determined to be anaphoric mentions as a separate pre-processing step. However, as discussed in Denis and Baldridge (2007), this requires a classification threshold to be carefully set and any error introduced in this anaphora resolution step is irrecoverably propagated. For this reason, many modern systems including Denis and Baldridge (2007) and Poon and Domingos (2008) have abandoned this pipeline model in favor of joint determination of anaphoricity and coreference in a single step.

Let us consider three strategies of considering anaphoricity in a coreference resolution model:

1. Ignore anaphoricity. This design decision is the simplest, and doesn't consider the anaphoricity of pronouns. It is possible that the model will resolve antecedents to non-referential pronouns.

2. Sequential determination of anaphoricity and coreference. The system works with two classifiers in cascade. It is not clear how the threshold of the anaphoricity classifier should be tuned so it doesn't prune too much of the pronouns.

3. Joint determination of anaphoricity and coreference. The decision of whether a mention is coreferent is done simultaneously as resolving its (possible) antecedent.

Denis and Baldridge (2007) proposes the user of joint inference with two classifiers using Integer Linear Programming. The four reported advantages of this method, is that (i) ILP allows to perform global inference based on two classifiers, instead of having to come up with a new procedure, (ii) ILP allows inference over multiple classifiers without having to set a threshold, (iii) it is more efficient than than a global Conditional Random Field (CRF) and finally, (iv), it is straightforward to create global constraints on the parameters. Denis and Baldridge (2007)'s ILP objective function incorporates the probabilities produced by an

anaphoricity classifier and a coreference (pairwise) classifier as weights. Poon and Domingos (2008) also manage to perform joint inference by means of first-order logic clauses that serve as predictors for anaphoricity. By incorporating this decision directly into coreference systems, the error propagation that would have been incurred in a pipeline approach can be avoided.

### 5.2 Unsupervised methods

There has been increasing attention in the coreference community for methods that work with little or no labeled training data (Ng, 2008). Corpora annotated with coreferences are very expensive to create. Even in English language, where there are the most number of resources available, most of them belong to the news domain. It is not straightforward to adapt the supervised algorithms described above to work on different domains or in different languages when there is not enough labeled data. In this subsection we describe three approaches to do inference when little or no training data is available.

A cluster-based approach allows leveraging "easy" decisions to help deciding "hard" using joint inference (Poon and Domingos, 2008). As usual, the main challenge in applying joint inference methods is managing its computational complexity. In this section we explore different models and approximate inference algorithms to resolve coreference.

Our discussion of unsupervised inference is arranged into three parts. First, we present two approaches for automatically selecting the number of clusters in sections 5.2.1 and 5.2.2. Later, in section 5.2.3, we present some issues on approximate inference algorithms that deal with the complexity of unsupervised learning.

#### 5.2.1 $N$-best clustering

One of the challenges of clustering algorithms is to determine the number of clusters in the final output. This is a hard problem because the number of clusterings is exponential in the number of mentions in the documents (Ng, 2008). To learn the number of clusters, two strategies are usually used: (i)use $N$-best clustering and (ii) impose a prior on the number of clusters.

For consistency,in the context of this paper, we define a clustering as in Ng (2008): a clustering of $n$ mentions is a $n$ x $n$ boolean matrix $C$, where an entry $(i, j)$ is 1 if and only if mentions $m_i$ and $m_j$ are coreferent. In this way a clustering is a different way of partitioning the mentions.

Ng (2008) uses $N$-best clustering to estimate the number of clusters using the Expectation-Maximization (EM) algorithm. In this way, the algorithm receives as an argument the number $N$ of clusterings to evaluate, instead of the number of clusters. In this context, a **clustering** is a partition into entity clusters over the set of all mentions in a document.

Although typically the EM algorithm is used in a parametric setting in which the number of clusters is fixed, it is modified so it can handle an indeterminate (infinite) number of mixtures. This is achieved by redefining the E-step to calculate the $N$ most probable coreference partitions using a Bell tree. By doing this, it is possible to define linguistically more robust features. This is in contract to the Haghighi model in which features such as salience and apposition can force the use of additional sampling or even prevent the use sampling.

Algorithm 1 shows Ng (2008)'s implementation of the Bell tree beam-search algorithm. The algorithm takes as input a set of $n$ probabilities of mentions in the text, and returns the

$N$ most probable partitions of the mentions, where the constant $\delta$ is the penalty of starting a new cluster, $S$ is a data structure that stores intermediate scores, $H_i$ stores the most probable $i$th-order partial partitions. Note that $H_i$ has a maximum size of $2N$, defining the size of the beam, that is used to store for the most likely partitions given the value of the parameters in the current iteration of EM.

**Algorithm 1** Ng's implementation of Bell Trees

**Require:** $M = \{m_1, ..., m_n\}$: mentions, $N$: no. of best partitions
**Ensure:** Output: N-best partitions
  *//initialize the data structures that store partial partitions*
  $H_1 := \{PP := \{[m1]\}\}, S(PP) = 1$
  $H_2, ..., H_n = \emptyset$
  **for** i = 2 to n **do**
    *//process each partial partition*
    **for all** $PP \in H_{i-1}$ **do**
      *//process each cluster in* $PP$
      **for all** $C \in PP$ **do**
        Extend $PP$ to $PP'$ by linking $m_i$ to $C$
        Compute $S(PP')$
        $H_i := H_i \cup \{PP'\}$
      **end for**
    **end for**
    $H_i := H_i \cup \{PP'\}$
    Extend $PP$ to $PP^\delta$ by putting $m_i$ into a new cluster
    Compute $S(PP^\delta)$
    $Hi := H_i \cup \{PP^\delta\}$
  **end for**
  **return** $N$ most probable partitions in $H_n$

#### 5.2.2 Imposing a prior on the number of clusters

An alternate method to select the number of clusters is by imposing a prior on the number of clusters. This is the approach taken in a Dirichlet process(Haghighi and Klein, 2007) or a Markov-logic based model (Poon and Domingos, 2008).

A Dirichlet Process is often explained using a Chinese Restaurant Process: Suppose a Chinese Restaurant has an infinite number of tables, each of which can seat an infinite number of people. Each customer $n + 1$ is seated at one of $m + 1$ places ($m \leq n$): either at one of the $m$ already-occupied tables or at a new unoccupied table. Such a process is of clear benefit when we must define a clustering in which we do not know the number of desired clusters. These models also have the distinct advantage that their parameters can be analytically integrated out *if* all of our features conform to the structure of this model (see below for caveats). By moving to a Hierarchical Dirichlet Process (Teh et al., 2006) (said to be more like a Chinese Restaurant Franchise), Haghighi and Klein are able to perform cross-document coreference resolution (Haghighi and Klein, 2007).

Figure 1 shows a graphical representation of the generative model used in (Haghighi and Klein, 2007) without the salience lists used for mention type distributions. The performance of this approach is arguably comparable with supervised methods. However, the introduction of the saliency feature requires expensive Gibbs sampling over both assignments and parameters.

Another deficiency of Haghighi's approach is that it is not clear how to use Gibbs sampling when using linguistic features that have deterministic dependencies (Poon and Domingos, 2008). For instance, an example described in (Poon and Domingos, 2008) is that a sentence with two mentions "Bill Gates, the chairman of Microsoft" would break the Gibbs sampling if the apposition feature is used, because at a particular iteration, only one mention can be moved from one cluster to another.

These issues brings up the concern of **extensibility**; often, complicated generative models such as Dirichlet processes offer no clear way to add new features (a frequent task for researchers) as evidenced by the breakdown of analytic integration when adding the salience feature and the inability to add an apposition feature via any straightforward method to Haghighi's base model.

An alternative approach to the Haghighi and Klein (2007) model, is to use Markov Logic Networks (MLN) (Poon and Domingos, 2008) to extend the model to allow apposition and predicate nominal features. The advantage of using a MLN is that inference can be performed efficiently using the MC-SAT sampling technique (discussed below). Instead of modeling pronoun saliency as described by Haghighi, Poon and Domingos (2008) impose an exponential prior on the distance (number of mentions) between a pronoun and its antecedent. Poon and Domingos (2008) claim that the unsupervised Markov Logic Network's performance is better than supervised methods.

#### 5.2.3 Sampling prediction instances

To overcome the problem of the large event space in unsupervised cluster-based models, Markov Logic Networks and Dirichlet Processes employ sampling of prediction instances. Usually the sampling methods employed for prediction instances are variations of Markov Chain Monte Carlo (MCMC) methods in which random samples are taken from a pool to produce an aggregate result from these individual samples.

Poon and Domingos are very thorough in addressing this issue for their unsupervised model. They use **Lazy-MC-SAT**, a "slice sampling" MCMC algorithm that exploits the fact that features are represented in logical form, allowing them to use a satisfiability solver to improve on efficiency. Though they used Dirichlet Processes, Haghighi and Klein also have to deal with the sampling of prediction instances. They employ the costly procedure of **Gibbs sampling** (a type of MCMC) (Haghighi and Klein, 2007; Poon and Domingos, 2008) .

### 5.3 Rule-based methods

Poon and Domingos (2008) proposed a rule-based system as a baseline. The system has four rules coded as a Markov Logic Network:

- Three rules to enforce agreement in terms of (i) type (person, organization, location, miscellaneous), (ii) number (singular, plural) and (iii) gender (male, female).

- A rule to enforce that non-pronouns are clustered by their head. This is given by the first order logic clause:

$$\begin{aligned} &\neg IsPrn(m_1) \wedge \neg IsPrn(m_2) \\ &\wedge Head(m_1, h_1) \wedge Head(m_2, h_2) \\ &InClust(m_1, c_1) \wedge InClust(m_2, c_2) \\ &\Rightarrow (c_1 = c_2) \Leftrightarrow h_1 = h_2). \end{aligned}$$

The agreement rules are given infinite weights, so the formulation of the MLN is equivalent to first order logic. However, Poon and Domingos (2008) explains that by using a large, but not infinite weight (for example 100) for the head rule, the MLN will cluster non-pronouns by their heads, except when it violates the agreement. It is interesting to note that this rule-based MLN can be extended to encode apposition. As in the unsupervised case, an exponential prior on the number of non-empty clusters and on the distance between a pronoun and its antecedent is imposed. Poon and Domingos (2008) reports that these simple MLN formulations achieve a F1 of 70.3, in comparison to the F1 of 63.9 of the Haghighi and Klein (2007) model trained on 60 documents.

## 6 Conclusion

We have presented a set of papers representative of modern trends in coreference resolution. Table 5 provides a summary of various aspects of the focus papers discussed throughout this review. First, we note that one's choice of model can be a limiting factor in what features can be used. By choosing models that are easily extensible, the field is likely to see quicker progress. Second, we recommend that future work in this area uses a more standard set of the standard corpora already available for coreference; in this way, results such as those of Poon will be comparable with those of Culotta. Because of this, there is no clear winner for the state-of-the-art.

Although unsupervised methods for coreference resolution are improving rapidly, we still cannot expect to gain domain adaption for free. Since most work has focused on coreference resolution in news corpora, it is unclear how the performance of these systems would behave in other real-world domains. For instance features such as entity type, which describe whether a mention is a person, a location or an organization, would not be appropriate in a domain such as medical journals. Likewise, it would be interesting to see how these modern methods perform on different languages, though a lack of annotated corpora currently hinder research in this area.

The use of first-order cluster-based features has given the field of coreference resolution a much-needed push, and will likely remain a staple of future state-of-the-art coreference systems. Also, the surprising result that unsupervised methods are rivaling supervised approaches, will likely generate much more research in this area. With the advances that have been made in the recent past, the field of coreference resolution has made great progress since the disappointing results of just a few years ago.

| | Culotta et al. (2007) | Haghighi and Klein (2007) | Ng (2008) | Poon and Domingos (2008) | Denis and Baldridge (2007) |
|---|---|---|---|---|---|
| Joint anaphora resolution | | | | X | X |
| Supervised / Unsupervised | S | U | U | U | S |
| Discriminative / Generative | D | G | G | G | D |
| Evaluation sets | ACE 2004 test | ACE 2004 train, MUC-6 test | ACE 2003 test | MUC-6, ACE 2004 train, ACE-2 | ACE-2 |
| Evaluation metrics | $B^3$ | MUC | MUC, CEAF | MUC, $B^3$ | MUC |
| Mention types | true | true | true, system | true | true |
| Feature domain | cluster | pair | pair | cluster | pair |
| Entity-assignment scope | greedy | global | greedy | global | global |
| Cross-Document | | X | | | |

Figure 5: A summary of the properties of the systems reviewed in this paper.

## References


Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *Proceedings of the Conference on Human Language Technology*.

Pascal Denis and Jason Baldridge. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of the North American Association for Computational Linguistics and the Conference on Human Language Technology*.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the Association for Computational Linguistics*.

Linguistic Data Consortium, 2008. *ACE (Automatic Content Extraction) English Annotation Guidelines for Entities*, version 6.6 2008.06.13 edition.

Xiaoqiang Luo. 2007. Coreference or not: A twin model for coreference resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, April.

Andrew McCallum and Ben Wellner. 2005. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the Association for Computational Linguistics*.

Vincent Ng. 2005. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the Association for Computational Linguistics*.

Vincent Ng. 2008. Unsupervised models for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Ngan L.T. Nguyen and Jin-Dong Kim. 2008. Exploring domain differences for the design of a pronoun resolution system for biomedical text. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 625–632, Manchester, UK, August. Coling 2008 Organizing Committee.

Hoifung Poon and Pedro Domingos. 2008. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. In *Machine Learning*.

Wee Meng Soon and Daniel Chung Yong Lim Hwee Tou Ng. 2006. A machine learning approach to coreference resolution of noun phrases. In *Computational Linguistics*.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical dirichlet processes. In *Journal of the American Statistical Association*.

# Appendix A: Features used by Ng and Cardie (2002)

| | | | |
|---|---|---|---|
| Lexical | | PRO_STR* | C if both NPs are pronominal and are the same string; else I. |
| | | PN_STR* | C if both NPs are proper names and are the same string; else I. |
| | | WORDS_STR | C if both NPs are non-pronominal and are the same string; else I. |
| | | SOON_STR_NONPRO* | C if both NPs are non-pronominal and the string of $NP_i$ matches that of $NP_j$; else I. |
| | | WORD_OVERLAP | C if the intersection between the content words in $NP_i$ and $NP_j$ is not empty; else I. |
| | | MODIFIER | C if the prenominal modifiers of one NP are a subset of the prenominal modifiers of the other; else I. |
| | | PN_SUBSTR | C if both NPs are proper names and one NP is a proper substring (w.r.t. content words only) of the other; else I. |
| | | WORDS_SUBSTR | C if both NPs are non-pronominal and one NP is a proper substring (w.r.t. content words only) of the other; else I. |
| Grammatical | NP type | BOTH_DEFINITES | C if both NPs start with "the;" I if neither start with "the;" else NA. |
| | | BOTH_EMBEDDED | C if both NPs are prenominal modifiers ; I if neither are prenominal modifiers; else NA. |
| | | BOTH_IN_QUOTES | C if both NPs are part of a quoted string; I if neither are part of a quoted string; else NA. |
| | | BOTH_PRONOUNS* | C if both NPs are pronouns; I if neither are pronouns, else NA. |
| | role | BOTH_SUBJECTS | C if both NPs are grammatical subjects; I if neither are subjects; else NA. |
| | | SUBJECT_1* | Y if $NP_i$ is a subject; else N. |
| | | SUBJECT_2 | Y if $NP_j$ is a subject; else N. |
| | linguistic constraints | AGREEMENT* | C if the NPs agree in both gender and number; I if they disagree in both gender and number; else NA. |
| | | ANIMACY* | C if the NPs match in animacy; else I. |
| | | MAXIMALNP* | I if both NPs have the same maximal NP projection; else C. |
| | | PREDNOM* | C if the NPs form a predicate nominal construction; else I. |
| | | SPAN* | I if one NP spans the other; else C. |
| | | BINDING* | I if the NPs violate conditions B or C of the Binding Theory; else C. |
| | | CONTRAINDICES* | I if the NPs cannot be co-indexed based on simple heuristics; else C. For instance, two non-pronominal NPs separated by a preposition cannot be co-indexed. |
| | | SYNTAX* | I if the NPs have incompatible values for the BINDING, CONTRAINDICES, SPAN or MAXIMALNP constraints; else C. |
| | ling. prefs | INDEFINITE* | I if $NP_j$ is an indefinite and not appositive; else C. |
| | | PRONOUN | I if $NP_i$ is a pronoun and $NP_j$ is not; else C. |
| | heuristics | CONSTRAINTS* | C if the NPs agree in GENDER and NUMBER and do not have incompatible values for CONTRAINDICES, SPAN, ANIMACY, PRONOUN, and CONTAINS_PN; I if the NPs have incompatible values for any of the above features; else NA. |
| | | CONTAINS_PN | I if both NPs are not proper names but contain proper names that mismatch on every word; else C. |
| | | DEFINITE_1 | Y if $NP_i$ starts with "the;" else N. |
| | | EMBEDDED_1* | Y if $NP_i$ is an embedded noun; else N. |
| | | EMBEDDED_2 | Y if $NP_j$ is an embedded noun; else N. |
| | | IN_QUOTE_1 | Y if $NP_i$ is part of a quoted string; else N. |
| | | IN_QUOTE_2 | Y if $NP_j$ is part of a quoted string; else N. |
| | | PROPER_NOUN | I if both NPs are proper names, but mismatch on every word; else C. |
| | | TITLE* | I if one or both of the NPs is a title; else C. |
| Semantic | | CLOSEST_COMP | C if $NP_i$ is the closest NP preceding $NP_j$ that has the same semantic class as $NP_j$ and the two NPs do not violate any of the linguistic constraints; else I. |
| | | SUBCLASS | C if the NPs have different head nouns but have an ancestor-descendent relationship in WordNet; else I. |
| | | WNDIST | Distance between $NP_i$ and $NP_j$ in WordNet (using the first sense only) when they have an ancestor-descendent relationship but have different heads; else infinity. |
| | | WNSENSE | Sense number in WordNet for which there exists an ancestor-descendent relationship between the two NPs when they have different heads; else infinity. |
| Pos | | PARANUM | Distance between the NPs in terms of the number of paragraphs. |
| Other | | PRO_RESOLVE* | C if $NP_j$ is a pronoun and $NP_i$ is its antecedent according to a naive pronoun resolution algorithm; else I. |
| | | RULE_RESOLVE | C if the NPs are coreferent according to a rule-based coreference resolution algorithm; else I. |

Figure 6: The features used by Ng and Cardie (2002) in their influential 2002 paper, grouped by type.