Proceedings of the 2009 Conference on Empirical Methods in
Natural Language Processing (EMNLP), Suntec, Singapore, August 2009, pp. 968--977.

# Supervised Models for Coreference Resolution

**Altaf Rahman** and **Vincent Ng**
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
{altaf,vince}@hlt.utdallas.edu

## Abstract

Traditional learning-based coreference resolvers operate by training a *mention-pair* classifier for determining whether two mentions are coreferent or not. Two independent lines of recent research have attempted to improve these mention-pair classifiers, one by learning a *mention-ranking* model to rank preceding mentions for a given anaphor, and the other by training an *entity-mention* classifier to determine whether a preceding cluster is coreferent with a given mention. We propose a cluster-ranking approach to coreference resolution that combines the strengths of mention rankers and entity-mention models. We additionally show how our cluster-ranking framework naturally allows anaphoricity determination to be learned jointly with coreference resolution. Experimental results on the ACE data sets demonstrate its superior performance to competing approaches.

## 1 Introduction

Noun phrase (NP) coreference resolution is the task of identifying which NPs (or *mentions*) refer to the same real-world entity or concept. Traditional learning-based coreference resolvers operate by training a model for classifying whether two mentions are co-referring or not (e.g., Soon et al. (2001), Ng and Cardie (2002b), Kehler et al. (2004), Ponzetto and Strube (2006)). Despite their initial successes, these *mention-pair* models have at least two major weaknesses. First, since each candidate antecedent for a mention to be resolved (henceforth an *active mention*) is considered independently of the others, these models only determine how good a candidate antecedent is relative to the active mention, but not how good a candidate antecedent is relative to other candidates. In other words, they fail to answer the critical question of which candidate antecedent is most probable. Second, they have limitations in their expressiveness: the information extracted from the two mentions alone may not be sufficient for making an informed coreference decision, especially if the candidate antecedent is a pronoun (which is semantically empty) or a mention that lacks descriptive information such as gender (e.g., *Clinton*).

To address the first weakness, researchers have attempted to train a *mention-ranking* model for determining which candidate antecedent is most probable given an active mention (e.g., Denis and Baldridge (2008)). Ranking is arguably a more natural reformulation of coreference resolution than classification, as a ranker allows all candidate antecedents to be considered *simultaneously* and therefore directly captures the competition among them. Another desirable consequence is that there exists a natural resolution strategy for a ranking approach: a mention is resolved to the candidate antecedent that has the highest rank. This contrasts with classification-based approaches, where many clustering algorithms have been employed to co-ordinate the pairwise coreference decisions (because it is unclear which one is the best).

To address the second weakness, researchers have investigated the acquisition of *entity-mention* coreference models (e.g., Luo et al. (2004), Yang et al. (2004)). Unlike mention-pair models, these entity-mention models are trained to determine whether an active mention belongs to a preceding, possibly partially-formed, coreference cluster. Hence, they can employ *cluster-level* features (i.e., features that are defined over any subset of mentions in a preceding cluster), which makes them more expressive than mention-pair models.

Motivated in part by these recently developed models, we propose in this paper a *cluster-ranking* approach to coreference resolution that combines the strengths of mention-ranking models and entity-mention models. Specifically, we recast coreference as the problem of determining which of a set of preceding coreference *clusters* is the best to link to an active mention using a learned *cluster ranker*. In addition, we show how anaphoricity determination (i.e., the task of determining whether a mention has an antecedent) can be learned *jointly* with coreference resolution in our cluster-ranking framework. It is worth noting that researchers typically adopt a *pipeline* coreference architecture, performing anaphoricity determination prior to coreference resolution and using the resulting information to prevent a coreference system from resolving mentions that are determined to be non-anaphoric (see Poesio et al. (2004) for an overview). As a result, errors in anaphoricity determination could be propagated to the resolver, possibly leading to a deterioration of coreference performance (see Ng and Cardie (2002a)). Jointly learning anaphoricity determination and coreference resolution can potentially address this error-propagation problem.

In sum, we believe our work makes three main contributions to coreference resolution:

**Proposing a simple, yet effective coreference model.** Our work advances the state-of-the-art in coreference resolution by bringing learning-based coreference systems to the next level of performance. When evaluated on the ACE 2005 coreference data sets, cluster rankers outperform three competing models — mention-pair, entity-mention, and mention-ranking models — by a large margin. Also, our joint-learning approach to anaphoricity determination and coreference resolution consistently yields cluster rankers that outperform those adopting the pipeline architecture. Equally importantly, cluster rankers are conceptually simple and easy to implement and do not rely on sophisticated training and inference procedures to make coreference decisions in dependent relation to each other, unlike relational coreference models (see McCallum and Wellner (2004)).

**Bridging the gap between machine-learning approaches and linguistically-motivated approaches to coreference resolution.** While machine learning approaches to coreference resolution have received a lot of attention since the mid-90s, popular learning-based coreference frameworks such as the mention-pair model are arguably rather unsatisfactory from a linguistic point of view. In particular, they have not leveraged advances in discourse-based anaphora resolution research in the 70s and 80s. Our work bridges this gap by realizing in a new machine learning framework ideas rooted in Lappin and Leass's (1994) heuristic-based pronoun resolver, which in turn was motivated by classic salience-based approaches to anaphora resolution.

**Revealing the importance of adopting the right model.** While entity-mention models have previously been shown to be worse or at best marginally better than their mention-pair counterparts (Luo et al., 2004; Yang et al., 2008), our cluster-ranking models, which are a natural extension of entity-mention models, significantly outperformed all competing approaches. This suggests that the use of an appropriate learning framework can bring us a long way towards high-performance coreference resolution.

The rest of the paper is structured as follows. Section 2 discusses related work. Section 3 describes our baseline coreference models: mention-pair, entity-mention, and mention-ranking. We discuss our cluster-ranking approach in Section 4, evaluate it in Section 5, and conclude in Section 6.

## 2 Related Work

**Heuristic-based cluster ranking.** As mentioned previously, the work most related to ours is Lappin and Leass (1994), whose goal is to perform pronoun resolution by assigning an anaphoric pronoun to the highest-scored preceding cluster. Nevertheless, Lappin and Leass's work differs from ours in several respects. First, they only tackle pronoun resolution rather than the full coreference task. Second, their algorithm is heuristic-based; in particular, the score assigned to a preceding cluster is computed by summing over the weights associated with the factors applicable to the cluster, where the weights are determined heuristically, rather than learned, unlike ours.

Like many heuristic-based pronoun resolvers (e.g., Mitkov (1998)), they first apply a set of constraints to filter grammatically incompatible candidate antecedents and then rank the remaining ones using salience factors. As a result, their cluster-ranking model employs only factors that capture the salience of a cluster, and can therefore be viewed as a simple model of attentional state (see Grosz and Sidner (1986)) realized by coreference clusters. By contrast, our resolution strategy is learned without applying hand-coded constraints in a separate filtering step. In particular, we attempt to determine the compatibility between a cluster and an active mention, using factors that determine not only salience (e.g., the distance between the cluster and the mention) but also lexical and grammatical compatibility, for instance.

**Entity-mention coreference models.** Luo et al. (2004) represent one of the earliest attempts to investigate learning-based entity-mention models. They use the ANY predicate to generate cluster-level features as follows: given a binary-valued feature X defined over a pair of mentions, they introduce an ANY-X cluster-level feature, which has the value TRUE if X is true between the active mention and *any* mention in the preceding cluster under consideration. Contrary to common wisdom, this entity-mention model underperforms its mention-pair counterpart in spite of the generalization from mention-pair to cluster-level features.

In Yang et al.'s (2004) entity-mention model, a training instance is composed of an active mention $m_k$, a preceding cluster $C$, and a mention $m_j$ in $C$ that is closest in distance to $m_k$ in the associated text. The feature set used to represent the instance is primarily composed of features that describe the relationship between $m_j$ and $m_k$, as well as a few cluster-level features. In other words, the model still relies heavily on features used in a mention-pair model. In particular, the inclusion of $m_j$ in the feature vector representation to some extent reflects the authors' lack of confidence that a strong entity-mention model can be trained without mention-pair-based features. Our ranking model, on the other hand, is trained without such features. More recently, Yang et al. (2008) have proposed another entity-mention model trained by inductive logic programming. Like their previous work, the scarcity of cluster-level predicates (only two are used) under-exploits the expressiveness of entity-mention models.

**Mention ranking.** The notion of ranking candidate antecedents can be traced back to centering algorithms, many of which use grammatical roles to rank forward-looking centers (see Grosz et al. (1995), Walker et al. (1998), and Mitkov (2002)). However, mention ranking has been employed in learning-based coreference resolvers only recently. As mentioned before, Denis and Baldridge (2008) train a mention-ranking model. Their work can be viewed as an extension of Yang et al.'s (2003) twin-candidate coreference model, which ranks only two candidate antecedents at a time. Unlike ours, however, their model ranks mentions rather than clusters, and relies on an independently-trained anaphoricity model.

**Anaphoricity determination.** Anaphoricity determination is often tackled independently of coreference. Pleonastic *it*s have been detected using heuristics (e.g., Kennedy and Boguraev (1996)) and learning-based techniques such as rule learning (e.g., Müller (2006)), kernels (e.g., Versley et al. (2008)), and distributional methods (e.g., Bergsma et al. (2008)). Non-anaphoric definite descriptions have been detected using heuristics (e.g., Vieira and Poesio (2000)) and unsupervised methods (e.g., Bean and Riloff (1999)). General anaphoricity models that are applicable to different types of NPs have been built using heuristics (e.g., Byron and Gegg-Harrison (2004)) and modeled generatively (e.g., Elsner and Charniak (2007)) and discriminatively (e.g., Uryupina (2003)). There have also been attempts to perform joint *inference* for anaphoricity determination and coreference resolution using integer linear programming (ILP), where a anaphoricity classifier and a coreference classifier are trained *independently* of each other, and then ILP is applied as a post-processing step to jointly infer anaphoricity and coreference decisions so that they are consistent with each other (e.g., Denis and Baldridge (2007)). Joint inference is different from our joint-learning approach, which allows the two tasks to be learned jointly and not independently.

## 3 Baseline Coreference Models

In this section, we describe three coreference models that will serve as our baselines: the mention-pair model, the entity-mention model, and the mention-ranking model. For illustrative purposes, we will use the text segment shown in Figure 1. Each mention $m$ in the segment is annotated as $[m]^{cid}_{mid}$, where $mid$ is the mention id and $cid$ is the id of the cluster to which $m$ belongs. As we can see, the mentions are partitioned into four sets, with *Barack Obama*, *his*, and *he* in one cluster, and each of the remaining mentions in its own cluster.

### 3.1 Mention-Pair Model

As noted before, a mention-pair model is a classifier that decides whether or not an active mention $m_k$ is coreferent with a candidate antecedent $m_j$. Each instance $\boldsymbol{i}(m_j, m_k)$ represents $m_j$ and

[*Barack Obama*]$^1_1$ *nominated* [*Hillary Rodham Clinton*]$^2_2$ *as* [[*his*]$^1_3$ *secretary of state*]$^3_4$ *on* [*Monday*]$^4_5$. [*He*]$^1_6$ ...

Figure 1: An illustrative example

$m_k$ and consists of the 39 features shown in Table 1. These features have largely been employed by state-of-the-art learning-based coreference systems (e.g., Soon et al. (2001), Ng and Cardie (2002b), Bengtson and Roth (2008)), and are computed automatically. As can be seen, the features are divided into four blocks. The first two blocks consist of features that describe the properties of $m_j$ and $m_k$, respectively, and the last two blocks of features describe the relationship between $m_j$ and $m_k$. The classification associated with a training instance is either positive or negative, depending on whether $m_j$ and $m_k$ are coreferent.

If one training instance were created from each pair of mentions, the negative instances would significantly outnumber the positives, yielding a skewed class distribution that will typically have an adverse effect on model training. As a result, only a subset of mention pairs will be generated for training. Following Soon et al. (2001), we create (1) a positive instance for each anaphoric mention $m_k$ and its closest antecedent $m_j$; and (2) a negative instance for $m_k$ paired with each of the intervening mentions, $m_{j+1}, m_{j+2}, \ldots, m_{k-1}$. In our running example shown in Figure 1, three training instances will be generated for *He*: ***i***(*Monday*, *He*), ***i***(*secretary of state*, *He*), and ***i***(*his*, *He*). The first two of these instances will be labeled as negative, and the last one will be labeled as positive. To train a mention-pair classifier, we use the SVM learning algorithm from the SVM$^{light}$ package (Joachims, 2002), converting all multi-valued features into an equivalent set of binary-valued features.

After training, the resulting SVM classifier is used to identify an antecedent for a mention in a test text. Specifically, an active mention $m_k$ selects as its antecedent the closest preceding mention that is classified as coreferent with $m_k$. If $m_k$ is not classified as coreferent with any preceding mention, it will be considered non-anaphoric (i.e., no antecedent will be selected for $m_k$).

### 3.2 Entity-Mention Model

Unlike a mention-pair model, an entity-mention model is a classifier that decides whether or not an active mention $m_k$ is coreferent with a *partial cluster* $c_j$ that precedes $m_k$. Each training instance, $\boldsymbol{i}(c_j, m_k)$, represents $c_j$ and $m_k$. The features for an instance can be divided into two types: (1) features that describe $m_k$ (i.e, those shown in the second block of Table 1), and (2) cluster-level features, which describe the relationship between $c_j$ and $m_k$. Motivated by previous work (Luo et al., 2004; Culotta et al., 2007; Yang et al., 2008), we create cluster-level features from mention-pair features using four predicates: NONE, MOST-FALSE, MOST-TRUE, and ALL. Specifically, for each feature X shown in the last two blocks in Table 1, we first convert X into an equivalent set of binary-valued features if it is multi-valued. Then, for each resulting binary-valued feature $X_b$, we create four binary-valued cluster-level features: (1) NONE-$X_b$ is true when $X_b$ is false between $m_k$ and each mention in $c_j$; (2) MOST-FALSE-$X_b$ is true when $X_b$ is true between $m_k$ and less than half (but at least one) of the mentions in $c_j$; (3) MOST-TRUE-$X_b$ is true when $X_b$ is true between $m_k$ and at least half (but not all) of the mentions in $c_j$; and (4) ALL-$X_b$ is true when $X_b$ is true between $m_k$ and each mention in $c_j$. Hence, for each $X_b$, exactly one of these four cluster-level features evaluates to true.

Following Yang et al. (2008), we create (1) a positive instance for each anaphoric mention $m_k$ and the preceding cluster $c_j$ to which it belongs; and (2) a negative instance for $m_k$ paired with each partial cluster whose last mention appears between $m_k$ and its closest antecedent (i.e., the last mention of $c_j$). Consider again our running example. Three training instances will be generated for *He*: ***i***($\{$*Monday*$\}$, *He*), ***i***($\{$*secretary of state*$\}$, *He*), and ***i***($\{$*Barack Obama*, *his*$\}$, *He*). The first two of these instances will be labeled as negative, and the last one will be labeled as positive. As in the mention-pair model, we train an entity-mention classifier using the SVM learner.

After training, the resulting classifier is used to identify a preceding cluster for a mention in a test text. Specifically, the mentions are processed in a left-to-right manner. For each active mention $m_k$, a test instance is created between $m_k$ and each of the preceding clusters formed so far. All the test instances are then presented to the classifier. Finally, $m_k$ will be linked to the closest preceding cluster that is classified as coreferent with $m_k$. If $m_k$ is not classified as coreferent with

| | | |
|---|---|---|
| **Features describing $m_j$, a candidate antecedent** | | |
| 1 | PRONOUN_1 | Y if $m_j$ is a pronoun; else N |
| 2 | SUBJECT_1 | Y if $m_j$ is a subject; else N |
| 3 | NESTED_1 | Y if $m_j$ is a nested NP; else N |
| **Features describing $m_k$, the mention to be resolved** | | |
| 4 | NUMBER_2 | SINGULAR or PLURAL, determined using a lexicon |
| 5 | GENDER_2 | MALE, FEMALE, NEUTER, or UNKNOWN, determined using a list of common first names |
| 6 | PRONOUN_2 | Y if $m_k$ is a pronoun; else N |
| 7 | NESTED_2 | Y if $m_k$ is a nested NP; else N |
| 8 | SEMCLASS_2 | the semantic class of $m_k$; can be one of PERSON, LOCATION, ORGANIZATION, DATE, TIME, MONEY, PERCENT, OBJECT, OTHERS, determined using WordNet and an NE recognizer |
| 9 | ANIMACY_2 | Y if $m_k$ is determined as HUMAN or ANIMAL by WordNet and an NE recognizer; else N |
| 10 | PRO_TYPE_2 | the nominative case of $m_k$ if it is a pronoun; else NA. E.g., the feature value for *him* is HE |
| **Features describing the relationship between $m_j$, a candidate antecedent and $m_k$, the mention to be resolved** | | |
| 11 | HEAD_MATCH | C if the mentions have the same head noun; else I |
| 12 | STR_MATCH | C if the mentions are the same string; else I |
| 13 | SUBSTR_MATCH | C if one mention is a substring of the other; else I |
| 14 | PRO_STR_MATCH | C if both mentions are pronominal and are the same string; else I |
| 15 | PN_STR_MATCH | C if both mentions are proper names and are the same string; else I |
| 16 | NONPRO_STR_MATCH | C if the two mentions are both non-pronominal and are the same string; else I |
| 17 | MODIFIER_MATCH | C if the mentions have the same modifiers; NA if one of both of them don't have a modifier; else I |
| 18 | PRO_TYPE_MATCH | C if both mentions are pronominal and are either the same pronoun or different only w.r.t. case; NA if at least one of them is not pronominal; else I |
| 19 | NUMBER | C if the mentions agree in number; I if they disagree; NA if the number for one or both mentions cannot be determined |
| 20 | GENDER | C if the mentions agree in gender; I if they disagree; NA if the gender for one or both mentions cannot be determined |
| 21 | AGREEMENT | C if the mentions agree in both gender and number; I if they disagree in both number and gender; else NA |
| 22 | ANIMACY | C if the mentions match in animacy; I if they don't; NA if the animacy for one or both mentions cannot be determined |
| 23 | BOTH_PRONOUNS | C if both mentions are pronouns; I if neither are pronouns; else NA |
| 24 | BOTH_PROPER_NOUNS | C if both mentions are proper nouns; I if neither are proper nouns; else NA |
| 25 | MAXIMALNP | C if the two mentions does not have the same maximial NP projection; else I |
| 26 | SPAN | C if neither mention spans the other; else I |
| 27 | INDEFINITE | C if $m_k$ is an indefinite NP and is not in an appositive relationship; else I |
| 28 | APPOSITIVE | C if the mentions are in an appositive relationship; else I |
| 29 | COPULAR | C if the mentions are in a copular construction; else I |
| 30 | SEMCLASS | C if the mentions have the same semantic class; I if they don't; NA if the semantic class information for one or both mentions cannot be determined |
| 31 | ALIAS | C if one mention is an abbreviation or an acronym of the other; else I |
| 32 | DISTANCE | binned values for sentence distance between the mentions |
| **Additional features describing the relationship between $m_j$, a candidate antecedent and $m_k$, the mention to be resolved** | | |
| 33 | NUMBER' | the concatenation of the NUMBER_2 feature values of $m_j$ and $m_k$. E.g., if $m_j$ is *Clinton* and $m_k$ is *they*, the feature value is SINGULAR-PLURAL, since $m_j$ is singular and $m_k$ is plural |
| 34 | GENDER' | the concatenation of the GENDER_2 feature values of $m_j$ and $m_k$ |
| 35 | PRONOUN' | the concatenation of the PRONOUN_2 feature values of $m_j$ and $m_k$ |
| 36 | NESTED' | the concatenation of the NESTED_2 feature values of $m_j$ and $m_k$ |
| 37 | SEMCLASS' | the concatenation of the SEMCLASS_2 feature values of $m_j$ and $m_k$ |
| 38 | ANIMACY' | the concatenation of the ANIMACY_2 feature values of $m_j$ and $m_k$ |
| 39 | PRO_TYPE' | the concatenation of the PRO_TYPE_2 feature values of $m_j$ and $m_k$ |

Table 1: The feature set for coreference resolution. Non-relational features describe a mention and in most cases take on a value of **Y**ES or **N**O. Relational features describe the relationship between the two mentions and indicate whether they are **C**OMPATIBLE, **I**NCOMPATIBLE or **N**OT **A**PPLICABLE.

any preceding cluster, it will be considered non-anaphoric. Note that all partial clusters preceding $m_k$ are formed incrementally based on the predictions of the classifier for the first $k-1$ mentions.

### 3.3 Mention-Ranking Model

As noted before, a ranking model imposes a ranking on all the candidate antecedents of an active mention $m_k$. To train a ranker, we use the SVM ranker-learning algorithm from the SVM$^{light}$ package. Like the mention-pair model, each training instance $\boldsymbol{i}(m_j, m_k)$ represents $m_k$ and a preceding mention $m_j$. In fact, the features that represent the instance as well as the method for creating training instances are identical to those employed by the mention-pair model.

The only difference lies in the assignment of class values to training instances. Assuming that $S_k$ is the set of training instances created for anaphoric mention $m_k$, the class value for an instance $\boldsymbol{i}(m_j, m_k)$ in $S_k$ is the rank of $m_j$ among competing candidate antecedents, which is 2 if $m_j$ is the closest antecedent of $m_k$, and 1 otherwise.[1] To exemplify, consider our running example. As in the mention-pair model, three training instances will be generated for *He*: ***i***(*Monday*, *He*), ***i***(*secretary of state*, *He*), ***i***(*his*, *He*). The third instance will have a class value of 2, and the remaining two will have a class value of 1.

After training, the mention-ranking model is applied to rank the candidate antecedents for an active mention in a test text as follows. Given an active mention $m_k$, we follow Denis and Baldridge (2008) and use an independently-trained classifier to determine whether $m_k$ is non-anaphoric. If so, $m_k$ will not be resolved. Otherwise, we create test instances for $m_k$ by pairing it with each of its preceding mentions. The test instances are then presented to the ranker, and the preceding mention that is assigned the largest value by the ranker is selected as the antecedent of $m_k$.

The anaphoricity classifier used in the resolution step is trained with 26 of the 37 features[2] described in Ng and Cardie (2002a) that are deemed useful for distinguishing between anaphoric and non-anaphoric mentions. These features can be broadly divided into two types: (1) features that encode the form of the mention (e.g., NP type, number, definiteness), and (2) features that compare the mention to one of its preceding mentions.

## 4 Coreference as Cluster Ranking

In this section, we describe our cluster-ranking approach to NP coreference. As noted before, our approach aims to combine the strengths of entity-mention models and mention-ranking models.

### 4.1 Training and Applying a Cluster Ranker

For ease of exposition, we will describe in this subsection how to train and apply a cluster ranker when it is used in a pipeline architecture, where anaphoricity determination is performed prior to coreference resolution. In the next subsection, we will show how the two tasks can be learned jointly.

[1]A larger class value implies a better rank in SVM$^{light}$.

[2]The 11 features that we did not employ are CONJ, POSSESSIVE, MODIFIER, POSTMODIFIED, SPECIAL_NOUNS, POST, SUBCLASS, TITLE, and the positional features.

Recall that a cluster ranker ranks a set of preceding clusters for an active mention $m_k$. Since a cluster ranker is a hybrid of a mention-ranking model and an entity-mention model, the way it is trained and applied is also a hybrid of the two. In particular, the instance representation employed by a cluster ranker is identical to that used by an entity-mention model, where each training instance $\boldsymbol{i}(c_j,\ m_k)$ represents a preceding cluster $c_j$ and an anaphoric mention $m_k$ and consists of cluster-level features formed from predicates. Unlike in an entity-mention model, however, in a cluster ranker, (1) a training instance is created between each anaphoric mention $m_k$ and *each* of its preceding clusters; and (2) since we are training a model for ranking clusters, the assignment of class values to training instances is similar to that of a mention ranker. Specifically, the class value of a training instance $\boldsymbol{i}(c_j,\ m_k)$ created for $m_k$ is the rank of $c_j$ among the competing clusters, which is 2 if $m_k$ belongs to $c_j$, and 1 otherwise.

Applying the learned cluster ranker to a test text is similar to applying a mention ranker. Specifically, the mentions are processed in a left-to-right manner. For each active mention $m_k$, we first apply an independently-trained classifier to determine if $m_k$ is non-anaphoric. If so, $m_k$ will not be resolved. Otherwise, we create test instances for $m_k$ by pairing it with each of its preceding clusters. The test instances are then presented to the ranker, and $m_k$ is linked to the cluster that is assigned the highest value by the ranker. Note that these partial clusters preceding $m_k$ are formed incrementally based on the predictions of the ranker for the first $k-1$ mentions; no gold-standard coreference information is used in their formation.

### 4.2 Joint Anaphoricity Determination and Coreference Resolution

The cluster ranker described above can be used to determine which preceding cluster an anaphoric mention should be linked to, but it cannot be used to determine whether a mention is anaphoric or not. The reason is simple: all the training instances are generated from anaphoric mentions. Hence, to jointly learn discourse-new detection and coreference resolution, we must train the ranker using instances generated from *both* anaphoric and non-anaphoric mentions.

Specifically, when training the ranker, we provide each active mention with the option to start a new cluster by creating an additional instance that (1) contains features that solely describe the active mention (i.e., the features shown in the second block of Table 1), and (2) has the highest rank value among competing clusters (i.e., 2) if it is non-anaphoric and the lowest rank value (i.e., 1) otherwise. The main advantage of jointly learning the two tasks is that it allows the ranking model to evaluate *all* possible options for an active mention (i.e., whether to resolve it, and if so, which preceding cluster is the best) *simultaneously*.

After training, the resulting cluster ranker processes the mentions in a test text in a left-to-right manner. For each active mention $m_k$, we create test instances for it by pairing it with each of its preceding clusters. To allow for the possibility that $m_k$ is non-anaphoric, we create an additional test instance that contains features that solely describe the active mention (similar to what we did in the training step above). All these test instances are then presented to the ranker. If the additional test instance is assigned the highest rank value by the ranker, then $m_k$ is classified as non-anaphoric and will not be resolved. Otherwise, $m_k$ is linked to the cluster that has the highest rank. As before, all partial clusters preceding $m_k$ are formed incrementally based on the predictions of the ranker for the first $k - 1$ mentions.

## 5 Evaluation

### 5.1 Experimental Setup

**Corpus.** We use the ACE 2005 coreference corpus as released by the LDC, which consists of the 599 training documents used in the official ACE evaluation.[3] To ensure diversity, the corpus was created by selecting documents from six different sources: Broadcast News (bn), Broadcast Conversations (bc), Newswire (nw), Webblog (wb), Usenet (un), and conversational telephone speech (cts). The number of documents belonging to each source is shown in Table 2. For evaluation, we partition the 599 documents into a training set and a test set following a 80/20 ratio, ensuring that the two sets have the same proportion of documents from the six sources.

**Mention extraction.** We evaluate each coreference model using both *true mentions* (i.e., gold standard mentions[4]) and *system mentions* (i.e., automatically identified mentions). To extract system mentions from a test text, we trained a mention extractor on the training texts. Following Florian et al. (2004), we recast mention extraction as a sequence labeling task, where we assign to each token in a test text a label that indicates whether it **b**egins a mention, is **i**nside a mention, or is **o**utside a mention. Hence, to learn the extractor, we create one training instance for each token in a training text and derive its class value (one of **b**, **i**, and **o**) from the annotated data. Each instance represents $w_i$, the token under consideration, and consists of 29 linguistic features, many of which are modeled after the systems of Bikel et al. (1999) and Florian et al. (2004), as described below.

| Dataset | bn | bc | nw | wl | un | cts |
|---|---|---|---|---|---|---|
| # of documents | 60 | 226 | 106 | 119 | 49 | 39 |

Table 2: Statistics for the ACE 2005 corpus

**Lexical (7):** Tokens in a window of 7: $\{w_{i-3}, \ldots, w_{i+3}\}$.

**Capitalization (4):** Determine whether $w_i$ `IsAllCap`, `IsInitCap`, `IsCapPeriod`, and `IsAllLower` (see Bikel et al. (1999)).

**Morphological (8):** $w_i$'s prefixes and suffixes of length one, two, three, and four.

**Grammatical (1):** The part-of-speech (POS) tag of $w_i$ obtained using the Stanford log-linear POS tagger (Toutanova et al., 2003).

**Semantic (1):** The named entity (NE) tag of $w_i$ obtained using the Stanford CRF-based NE recognizer (Finkel et al., 2005).

**Gazetteers (8):** Eight dictionaries containing pronouns (77 entries), common words and words that are not names (399.6k), person names (83.6k), person titles and honorifics (761), vehicle words (226), location names (1.8k), company names (77.6k), and nouns extracted from WordNet that are hyponyms of PERSON (6.3k).

We employ CRF++[5], a C++ implementation of conditional random fields, for training the mention detector, which achieves an F-score of 86.7 (86.1 recall, 87.2 precision) on the test set. These extracted mentions are to be used as system mentions in our coreference experiments.

**Scoring programs.** To score the output of a coreference model, we employ three scoring programs: MUC (Vilain et al., 1995), $B^3$ (Bagga and Baldwin, 1998), and $\phi_3$-CEAF (Luo, 2005).

[3]Since we did not participate in ACE 2005, we do not have access to the official test set.

[4]Note that only mention *boundaries* are used.

[5]Available from http://crfpp.sourceforge.net

There is a complication, however. When scoring a *response* (i.e., system-generated) partition against a *key* (i.e., gold-standard) partition, a scoring program needs to construct a mapping between the mentions in the response and those in the key. If the response is generated using true mentions, then every mention in the response is mapped to some mention in the key and vice versa; in other words, there are no *twinless* (i.e., unmapped) mentions (Stoyanov et al., 2009). However, this is not the case when system mentions are used. The aforementioned complication does not arise from the construction of the mapping, but from the fact that Bagga and Baldwin (1998) and Luo (2005) do not specify how to apply $B^3$ and CEAF to score partitions generated from system mentions.

We propose a simple solution to this problem: we remove all and only those twinless system mentions that are singletons before applying $B^3$ and CEAF. The reason is simple: since the coreference resolver has successfully identified these mentions as singletons, it should not be penalized, and removing them allows us to avoid such penalty. Note that we only remove twinless (as opposed to all) system mentions that are singletons: this allows us to reward a resolver for successful identification of singleton mentions that have twins, thus overcoming a major weakness of and common criticism against the MUC scorer. Also, we retain twinless system mentions that are non-singletons, as the resolver should be penalized for identifying spurious coreference relations. On the other hand, we do not remove twinless mentions in the key partition, as we want to ensure that the resolver makes the correct (non-)coreference decisions for them. We believe that our proposal addresses Stoyanov et al.'s (2009) problem of having very low precision when applying the CEAF scorer to score partitions of system mentions.

### 5.2 Results and Discussions

**The mention-pair baseline.** We train our first baseline, the mention-pair coreference classifier, using the SVM learning algorithm as implemented in the $\text{SVM}^{light}$ package (Joachims, 2002).[6] Results of this baseline using true mentions and system mentions, shown in row 1 of Tables 3 and 4, are reported in terms of recall (R), precision (P), and F-score (F) provided by the three scoring programs. As we can see, this baseline achieves F-scores of 54.3–70.0 and 53.4–62.5 for true mentions and system mentions, respectively.

**The entity-mention baseline.** Next, we train our second baseline, the entity-mention coreference classifier, using the SVM learner. Results of this baseline are shown in row 2 of Tables 3 and 4. For true mentions, this baseline achieves an F-score of 54.8–70.7. In comparison to the mention-pair baseline, F-score rises insignificantly according to all three scorers.[7] Similar trends can be observed for system mentions, where the F-scores between the two models are statistically indistinguishable across the board. While the insignificant performance difference is somewhat surprising given the improved expressiveness of entity-mention models over mention-pair models, similar trends have been reported by Luo et al. (2004).

**The mention-ranking baseline.** Our third baseline is the mention-ranking coreference model, trained using the ranker-learning algorithm in $\text{SVM}^{light}$. To identify non-anaphoric mentions, we employ two methods. In the first method, we adopt a pipeline architecture, where we train an SVM classifier for anaphoricity determination independently of the mention ranker on the training set using the 26 features described in Section 3.3. We then apply the resulting classifier to each test text to filter non-anaphoric mentions prior to coreference resolution. Results of the mention ranker are shown in row 3 of Tables 3 and 4. As we can see, the ranker achieves F-scores of 57.8–71.2 and 54.1–65.4 for true mentions and system mentions, respectively, yielding a significant improvement over the entity-mention baseline in all but one case (MUC/true mentions).

In the second method, we perform anaphoricity determination jointly with coreference resolution using the method described in Section 4.2. While we discussed this joint learning method in the context of cluster ranking, it should be easy to see that the method is equally applicable to a mention ranker. Results of the mention ranker using this joint architecture are shown in row 4 of Tables 3 and 4. As we can see, the ranker achieves F-scores of 61.6–73.4 and 55.6–67.1 for true mentions and system mentions, respectively. For both types of mentions, the improvements over the corresponding results for the entity-mention baseline

[6] For this and subsequent uses of the SVM learner in our experiments, we set all parameters to their default values.

[7] We use Approximate Randomization (Noreen, 1989) for testing statistical significance, with $p$ set to 0.05.

| | Coreference Model | MUC R | MUC P | MUC F | CEAF R | CEAF P | CEAF F | $B^3$ R | $B^3$ P | $B^3$ F |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Mention-pair model | 71.7 | 69.2 | 70.4 | 54.3 | 54.3 | 54.3 | 53.3 | 63.6 | 58.0 |
| 2 | Entity-mention model | 71.7 | 69.7 | 70.7 | 54.8 | 54.8 | 54.8 | 53.2 | 65.1 | 58.5 |
| 3 | Mention-ranking model (Pipeline) | 68.7 | 73.9 | 71.2 | 57.8 | 57.8 | 57.8 | 55.8 | 63.9 | 59.6 |
| 4 | Mention-ranking model (Joint) | 69.4 | 77.8 | 73.4 | 61.6 | 61.6 | 61.6 | 57.0 | 70.1 | 62.9 |
| 5 | Cluster-ranking model (Pipeline) | 71.7 | 78.2 | 74.8 | 61.8 | 61.8 | 61.8 | 58.2 | 69.1 | 63.2 |
| 6 | Cluster-ranking model (Joint) | 69.9 | 83.3 | **76.0** | 63.3 | 63.3 | **63.3** | 56.0 | 74.6 | **64.0** |

Table 3: MUC, CEAF, and $B^3$ coreference results using true mentions.

| | Coreference Model | MUC R | MUC P | MUC F | CEAF R | CEAF P | CEAF F | $B^3$ R | $B^3$ P | $B^3$ F |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Mention-pair model | 70.0 | 56.4 | 62.5 | 56.1 | 51.0 | 53.4 | 50.8 | 57.9 | 54.1 |
| 2 | Entity-mention model | 68.5 | 57.2 | 62.3 | 56.3 | 50.2 | 53.1 | 51.2 | 57.8 | 54.3 |
| 3 | Mention-ranking model (Pipeline) | 62.2 | 68.9 | 65.4 | 51.6 | 56.7 | 54.1 | 52.3 | 61.8 | 56.6 |
| 4 | Mention-ranking model (Joint) | 62.1 | 73.0 | 67.1 | 53.0 | 58.5 | 55.6 | 50.4 | 65.5 | 56.9 |
| 5 | Cluster-ranking model (Pipeline) | 65.3 | 72.3 | 68.7 | 54.1 | 59.3 | 56.6 | 55.3 | 63.7 | 59.2 |
| 6 | Cluster-ranking model (Joint) | 64.1 | 75.4 | **69.3** | 56.7 | 62.6 | **59.5** | 54.4 | 70.5 | **61.4** |

Table 4: MUC, CEAF, and $B^3$ coreference results using system mentions.

are significant, and suggest that mention ranking is a precision-enhancing device. Moreover, in comparison to the pipeline architecture in row 3, we see that F-score rises significantly by 2.2–3.8% for true mentions, and improves by a smaller margin of 0.3–1.7% for system mentions. These results demonstrate the benefits of joint modeling.

**Our cluster-ranking model.** Finally, we evaluate our cluster-ranking model. As in the mention-ranking baseline, we employ both the pipeline architecture and the joint architecture for anaphoricity determination. Results are shown in rows 5 and 6 of Tables 3 and 4, respectively, for the two architectures. When true mentions are used, the pipeline architecture yields an F-score of 61.8–74.8, which represents a significant improvement over the mention ranker adopting the pipeline architecture. With the joint architecture, the cluster ranker achieves an F-score of 63.3–76.0. This also represents a significant improvement over the mention ranker adopting the joint architecture, the best of the baselines, and suggests that cluster ranking is a better precision-enhancing model than mention ranking. Moreover, comparing the results in these two rows reveals the superiority of the joint architecture over the pipeline architecture, particularly in terms of its ability to enhance system precision. Similar performance trends can be observed when system mentions are used.

## 6 Conclusions

We have presented a cluster-ranking approach that recasts the mention resolution process as the problem of finding the best preceding cluster to link an active mention to. Crucially, our approach combines the strengths of entity-mention models and mention-ranking models. Experimental results on the ACE 2005 corpus show that (1) jointly learning coreference resolution and anaphoricity determination allows the cluster ranker to achieve better performance than adopting a pipeline coreference architecture; and (2) our cluster ranker significantly outperforms the mention ranker, the best of the three baseline coreference models, under both the pipeline architecture and the joint architecture. Overall, we believe that our cluster-ranking approach advances the state-of-the-art in coreference resolution both theoretically and empirically.

## Acknowledgments

We thank the three anonymous reviewers for their invaluable comments on the paper. This work was supported in part by NSF Grant IIS-0812261.

## References

A. Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proc. of COLING-ACL*, pages 79–85.

D. Bean and E. Riloff. 1999. Corpus-based identification of non-anaphoric noun phrases. In *Proc. of the ACL*, pages 373–380.

E. Bengtson and D. Roth. 2008. Understanding the values of features for coreference resolution. In *Proc. of EMNLP*, pages 294–303.

S. Bergsma, D. Lin, and R. Goebel. 2008. Distributional identification of non-referential pronouns. In *Proc. of ACL-08:HLT*, pages 10–18.

D. Bikel, R. Schwartz, and R. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1–3):211–231.

D. Byron and W. Gegg-Harrison. 2004. Eliminating non-referring noun phrases from coreference resolution. In *Proc. of DAARC*, pages 21–26.

A. Culotta, M. Wick, and A. McCallum. 2007. First-order probabilistic models for coreference resolution. In *Proc. of NAACL-HLT*, pages 81–88.

P. Denis and J. Baldridge. 2007. Global, joint determination of anaphoricity and coreference resolution using integer programming. In *Proc. of NAACL-HLT*, pages 236–243.

P. Denis and J. Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proc. of EMNLP*, pages 660–669.

M. Elsner and E. Charniak. 2007. A generative discourse-new model for text coherence. Technical Report CS-07-04, Brown University.

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proc. of the ACL*, pages 363–370.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, and I. Zitouni. 2004. A statistical model for multilingual entity detection and tracking. In *Proc. of HLT/NAACL*.

B. J. Grosz, A. K. Joshi, and S. Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

B. J. Grosz and C. L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

T. Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. of KDD*, pages 133–142.

A. Kehler, D. Appelt, L. Taylor, and A. Simma. 2004. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proc. of HLT/NAACL*.

C. Kennedy and B. Boguraev. 1996. Anaphor for everyone: Pronominal anaphora resolution without a parser. In *Proc. of COLING*, pages 113–118.

S. Lappin and H. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–562.

X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proc. of the ACL*, pages 135–142.

X. Luo. 2005. On coreference resolution performance metrics. In *Proc. of HLT/EMNLP*, pages 25–32.

A. McCallum and B. Wellner. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Advances in NIPS*.

R. Mitkov. 2002. *Anaphora Resolution*. Longman.

R. Mitkov. 1998. Robust pronoun resolution with limited knowledge. In *Proc. of COLING/ACL*, pages 869–875.

C. Müller. 2006. Automatic detection of nonreferential it in spoken multi-party dialog. In *Proc. of EACL*, pages 49–56.

V. Ng and C. Cardie. 2002a. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proc. of COLING*, pages 730–736.

V. Ng and C. Cardie. 2002b. Improving machine learning approaches to coreference resolution. In *Proc. of the ACL*, pages 104–111.

E. W. Noreen. 1989. *Computer Intensive Methods for Testing Hypothesis: An Introduction*. John Wiley & Sons.

M. Poesio, O. Uryupina, R. Vieira, M. Alexandrov-Kabadjov, and R. Goulart. 2004. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In *Proc. of the ACL Workshop on Reference Resolution*.

S. P. Ponzetto and M. Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. of HLT/NAACL*, pages 192–199.

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

V. Stoyanov, N. Gilbert, C. Cardie, and E. Riloff. 2009. Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proc. of the ACL*.

K. Toutanova, D. Klein, C. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proc. of HLT-NAACL*, pages 252–259.

O. Uryupina. 2003. High-precision identification of discourse new and unique noun phrases. In *Proc. of the ACL Student Research Workshop*.

Y. Versley, A. Moschitti, M. Poesio, and X. Yang. 2008. Coreference systems based on kernel methods. In *Proc. of COLING*, pages 961–968.

R. Vieira and M. Poesio. 2000. Processing definite descriptions in corpora. In *Corpus-based and Computational Approaches to Discourse Anaphora*, pages 189–212. UCL Press.

M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proc. of MUC-6*, pages 45–52.

M. Walker, A. Joshi, and E. Prince, editors. 1998. *Centering Theory in Discourse*. Oxford University Press.

X. Yang, G. Zhou, J. Su, and C. L. Tan. 2003. Coreference resolution using competitive learning approach. In *Proc. of the ACL*, pages 176–183.

X. Yang, J. Su, G. Zhou, and C. L. Tan. 2004. An NP-cluster based approach to coreference resolution. In *Proc. of COLING*, pages 226–232.

X. Yang, J. Su, J. Lang, C. L. Tan, and S. Li. 2008. An entity-mention model for coreference resolution with inductive logic programming. In *Proc. of the ACL*, pages 843–851.