Proceedings of the 2008 Conference on Empirical Methods in
Natural Language Processing (EMNLP), Honolulu, HI, October 2008, pp. 640-649.

Unsupervised Models for Coreference Resolution

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Richardson, TX 75083-0688
vince@hlt.utdallas.edu

Abstract

We present a generative model for unsupervised coreference resolution that views coreference as an EM clustering process. For comparison purposes, we revisit Haghighi and Klein's (2007) fully-generative Bayesian model for unsupervised coreference resolution, discuss its potential weaknesses and consequently propose three modifications to their model. Experimental results on the ACE data sets show that our model outperforms their original model by a large margin and compares favorably to the modified model.

1 Introduction

Coreference resolution is the problem of identifying which *mentions* (i.e., noun phrases) refer to which real-world *entities*. The availability of annotated coreference corpora produced as a result of the MUC conferences and the ACE evaluations has prompted the development of a variety of supervised machine learning approaches to coreference resolution in recent years. The focus of learning-based coreference research has also shifted from the acquisition of a *pairwise* model that determines whether two mentions are co-referring (e.g., Soon et al. (2001), Ng and Cardie (2002), Yang et al. (2003)) to the development of rich linguistic features (e.g., Ji et al. (2005), Ponzetto and Strube (2006)) and the exploitation of advanced techniques that involve joint learning (e.g., Daumé III and Marcu (2005)) and joint inference (e.g., Denis and Baldridge (2007)) for coreference resolution and a related extraction task. The rich features, coupled with the increased complexity of coreference models, have made these supervised approaches more dependent on labeled data and less applicable to languages for which little or no annotated data exists. Given the growing importance of multi-lingual processing in the NLP community, however, the development of unsupervised and weakly supervised approaches for the automatic processing of resource-scarce languages has become more important than ever.

In fact, several popular weakly supervised learning algorithms such as self-training, co-training (Blum and Mitchell, 1998), and EM (Dempster et al., 1977) have been applied to coreference resolution (Ng and Cardie, 2003) and the related task of pronoun resolution (Müller et al., 2002; Kehler et al., 2004; Cherry and Bergsma, 2005). Given a small number of coreference-annotated documents and a large number of unlabeled documents, these weakly supervised learners aim to incrementally augment the labeled data by iteratively training a classifier[1] on the labeled data and using it to label mention pairs randomly drawn from the unlabeled documents as COREFERENT or NOT COREFERENT. However, classifying mention pairs using such iterative approaches is undesirable for coreference resolution: since the non-coreferent mention pairs significantly outnumber their coreferent counterparts, the resulting classifiers generally have an increasing tendency to (mis)label a pair as non-coreferent as bootstrapping progresses (see Ng and Cardie (2003)).

Motivated in part by these results, we present a generative, unsupervised model for probabilistically

[1]For co-training, a pair of *view* classifiers are trained; and for EM, a generative model is trained instead.

inducing coreference *partitions* on unlabeled *documents*, rather than classifying mention pairs, via EM clustering (Section 2). In fact, our model combines the best of two worlds: it operates at the document level, while exploiting essential linguistic constraints on coreferent mentions (e.g., gender and number agreement) provided by traditional pairwise classification models.

For comparison purposes, we revisit a fully-generative Bayesian model for unsupervised coreference resolution recently introduced by Haghighi and Klein (2007), discuss its potential weaknesses and consequently propose three modifications to their model (Section 3). Experimental results on the ACE data sets show that our model outperforms their original model by a large margin and compares favorably to the modified model (Section 4).

2 Coreference as EM Clustering

In this section, we will explain how we recast unsupervised coreference resolution as EM clustering. We begin by introducing some of the definitions and notations that we will use in this paper.

2.1 Definitions and Notations

A *mention* can be a pronoun, a name (i.e., a proper noun), or a nominal (i.e., a common noun). An *entity* is a set of coreferent mentions. Given a document D consisting of n mentions, $m_1, \ldots, m_n$, we use $Pairs(D)$ to denote the set of $\binom{n}{2}$ mention pairs, $\{m_{ij} \mid 1 \leq i < j \leq n\}$, where m_{ij} is formed from mentions m_i and m_j. The *pairwise probability* formed from m_i and m_j refers to the probability that the pair m_{ij} is coreferent and is denoted as $P_{coref}(m_{ij})$. A *clustering* of n mentions is an n x n Boolean matrix C, where C_{ij} (the (i,j)-th entry of C) is 1 if and only if mentions m_i and m_j are coreferent. An entry in C is *relevant* if it corresponds to a mention pair in $Pairs(D)$. A *valid clustering* is a clustering in which the relevant entries satisfy the transitivity constraint. In other words, C is valid if and only if (C_{ij} = 1 $\wedge$ C_{jk} = 1) $\Longrightarrow$ C_{ik} = 1 $\forall$ $1 \leq i < j < k \leq n$. Hence, a valid clustering corresponds to a *partition* of a given set of mentions, and the goal of coreference resolution is to produce a valid clustering in which each cluster corresponds to a distinct entity.

2.2 The Model

As mentioned previously, our generative model operates at the document level, inducing a valid clustering on a given document D. More specifically, our model consists of two steps. It first chooses a clustering C based on some clustering distribution $P(C)$, and then generates D given C:

$$P(D, C) = P(C)P(D \mid C).$$

To facilitate the incorporation of linguistic constraints defined on a pair of mentions, we represent D by its mention pairs, $Pairs(D)$. Now, assuming that these mention pairs are generated conditionally independently of each other given C_{ij},

$$P(D \mid C) = \prod_{m_{ij} \in Pairs(D)} P(m_{ij} \mid C_{ij}).$$

Next, we represent m_{ij} as a set of seven features that is potentially useful for determining whether m_i and m_j are coreferent (see Table 1).[2] Hence, we can rewrite $P(D \mid C)$ as

$$\prod_{m_{ij} \in Pairs(D)} P(m^1_{ij}, \ldots, m^7_{ij} \mid C_{ij}),$$

where m^k_{ij} is the value of the kth feature of m_{ij}.

To reduce data sparseness and improve the estimation of the above probabilities, we make conditional independence assumptions about the generation of these feature values. Specifically, as shown in the first column of Table 1, we divide the seven features into three *groups* (namely, strong coreference indicators, linguistic constraints, and mention types), assuming that two feature values are conditionally independent if and only if the corresponding features belong to different groups. With this assumption, we can decompose $P(m^1_{ij}, \ldots m^7_{ij} \mid C_{ij})$ into a product of three probabilities: $P(m^1_{ij}, m^2_{ij}, m^3_{ij} \mid C_{ij})$, $P(m^4_{ij}, m^5_{ij}, m^6_{ij} \mid C_{ij})$, and $P(m^7_{ij} \mid C_{ij})$. Each of these distributions represents a pair of multinomial distributions, one for the coreferent mention pairs (C_{ij} = 1) and the other for the non-coreferent mention pairs (C_{ij} = 0). Hence, the set of parameters of our model, Θ, consists of $P(m^1, m^2, m^3 \mid c)$, $P(m^4, m^5, m^6 \mid c)$, and $P(m^7 \mid c)$.

[2] See Soon et al. (2001) for details on feature value computations. Note that all feature values are computed automatically.

Feature Type	Feature ID	Feature	Description
Strong Coreference Indicators	1	STR_MATCH	T if neither of the two mentions is a pronoun and after discarding determiners, the string denoting mention m_i is identical to that of mention m_j; else F.
	2	ALIAS	T if one mention is an acronym, an abbreviation, or a name variant of the other; else F. For instance, *Bill Clinton* and *President Clinton* are aliases, so are *MIT* and *Massachusetts Institute of Technology*.
	3	APPOSITIVE	T if the mentions are in an appositive relationship; else F.
Linguistic Constraints	4	GENDER	T if the mentions agree in gender; F if they disagree; NA if gender information for one or both mentions cannot be determined.
	5	NUMBER	T if the mentions agree in number; F if they disagree; NA if number information for one or both mentions cannot be determined.
	6	SEM_CLASS	T if the mentions have the same semantic class; F if they don't; NA if the semantic class information for one or both mentions cannot be determined.
Mention Types	7	NPTYPE	the feature value is the concatenation of the mention type of the two mentions, $t_i t_j$, where $t_i, t_j \in \{$ PRONOUN, NAME, NOMINAL $\}$.

Table 1: Feature set for representing a mention pair. The first six features are relational features that test whether some property P holds for the mention pair under consideration and indicate whether the mention pair is **T**RUE or **F**ALSE w.r.t. P; a value of **N**OT **A**PPLICABLE is used when property P does not apply.

2.3 The Induction Algorithm

To induce a clustering C on a document D, we run EM on our model, treating D as observed data and C as hidden data. Specifically, we use EM to iteratively estimate the model parameters, Θ, from documents that are probabilistically labeled (with clusterings) and apply the resulting model to probabilistically re-label a document (with clusterings). More formally, we employ the following EM algorithm:

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

M-step: Using $P(C|D,\Theta)$ computed in the E-step, find the Θ' that maximizes the expected complete log likelihood, $\sum_C P(C|D,\Theta) \log P(D,C|\Theta')$.

We begin the induction process at the M-step.[3] To find the Θ that maximizes the expected complete log likelihood, we use maximum likelihood estimation with add-one smoothing. Since $P(C|D,\Theta)$ is not available in the first EM iteration, we instead use an initial distribution over clusterings, $P(C)$. The question, then, is: which $P(C)$ should we use? One possibility is the uniform distribution over all (possibly invalid) clusterings. Another, presumably better, choice is a distribution that assigns non-zero probability mass to only the valid clusterings. Yet another possibility is to set $P(C)$ based on a document labeled with coreference information. In our experiments, we employ this last method, assigning a probability of one to the correct clustering of the labeled document (see Section 4.1 for details).

After (re-)estimating Θ in the M-step, we proceed to the E-step, where the goal is to find the conditional clustering probabilities. Given a document D, the number of coreference clusterings is exponential in the number of mentions in D, even if we limit our attention to those that are valid. To cope with this computational complexity, we approximate the E-step by computing only the conditional probabilities that correspond to the N most probable coreference clusterings given the current Θ. We identify the N most probable clusterings and compute their probabilities as follows. First, using the current Θ, we reverse the generative model and compute $P_{coref}(m_{ij})$ for each mention pair m_{ij} in $Pairs(D)$. Next, using these pairwise probabilities, we apply Luo et al.'s (2004) Bell tree approach to coreference resolution to compute the N-best clusterings and their probabilities (see Section 2.4 for details). Finally, to obtain the required conditional clustering probabilities for the E-step, we normalize the probabilities assigned to the N-best clusterings so that they sum to one.

2.4 Computing the N-Best Partitions

As described above, given the pairwise probabilities, we use Luo et al.'s (2004) algorithm to heuristically compute the N-best clusterings (or, more precisely, N-best partitions[4]) and their probabilities based on

[3]Another possibility, of course, is to begin at the E-step by making an initial guess at Θ.

[4]Note that Luo et al.'s search algorithm only produces valid clusterings, implying that the resulting N-best clusterings are

Input: $M = \{m_1, ..., m_n\}$: mentions, N: no. of best partitions
Output: N-best partitions
1: *// initialize the data structures that store partial partitions*
2: $H_1 := \{PP := \{[m_1]\}\}$, $S(PP) = 1$
3: $H_2, ..., H_n = \emptyset$
4: **for** $i = 2$ to n
5:   *// process each partial partition*
6:   **foreach** $PP \in H_{i-1}$
7:     *// process each cluster in PP*
8:     **foreach** $C \in PP$
9:       Extend PP to PP' by linking m_i to C
10:       Compute $S(PP')$
11:       $H_i := H_i \cup \{PP'\}$
12:     Extend PP to PP^{δ} by putting m_i into a new cluster
13:     Compute $S(PP^{\delta})$
14:     $H_i := H_i \cup \{PP^{\delta}\}$
15: **return** N most probable partitions in H_n

Figure 1: Our implementation of Luo et al.'s algorithm

the Bell tree. Informally, each node in a Bell tree corresponds to an ith-order *partial* partition (i.e., a partition of the first i mentions of the given document), and the ith level of the tree contains *all* possible ith-order partial partitions. Hence, the set of leaf nodes constitutes all possible partitions of all of the mentions. The search for the N most probable partitions starts at the root, and a partitioning of the mentions is incrementally constructed as we move down the tree. Since an exhaustive search is computationally infeasible, Luo et al. employ a beam search procedure to explore only the most probable paths at each step of the search process. Figure 1 shows our implementation of this heuristic search algorithm.

The algorithm takes as input a set of n mentions (and their pairwise probabilities), and returns the N most probable partitionings of the mentions. It uses data structures S and the H_i's to store intermediate results. Specifically, $S(PP)$ stores the score of the partial partition PP. H_i is associated with the ith level of the Bell tree, and is used to store the most probable ith-order partial partitions. Each H_i has a maximum size of $2N$: if more than $2N$ partitions are inserted into a given H_i, then only the $2N$ most probable ones will be stored. This amounts to pruning the search space by employing a beam size of $2N$ (i.e., expanding only the $2N$ most probable partial partitions) at each step of the search.

The algorithm begins by initializing H_1 with the only partial partition of order one, $\{[m_1]\}$, which has a score of one (line 2). Then it processes the mentions sequentially, starting with m_2 (line 4). When processing m_i, it takes each partial partition PP in H_{i-1} and creates a set of ith-order partitions by extending PP with m_i in all possible ways. Specifically, for each cluster C (formed by a subset of the first i–1 mentions) in PP, the algorithm generates a new ith-order partition, PP', by linking m_i to C (line 9), and stores PP' in H_i (line 11). The score of PP', $S(PP')$, is computed by using the pairwise coreference probabilities as follows:

$$S(PP') = S(PP) \cdot \max_{m_k \in C} P_{coref}(m_{ki}).$$

Of course, PP can also be extended by putting m_i into a new cluster (line 12). This yields PP^{δ}, another partition to be inserted into H_i (line 14), and

$$S(PP^{\delta}) = \delta \cdot S(PP) \cdot (1 - \max_{k \in \{1,\ldots,i-1\}} P_{coref}(m_{ki})),$$

where δ (the *start penalty*) is a positive constant ($<$ 1) used to penalize partitions that start a new cluster. After processing each of the n mentions using the above steps, the algorithm returns the N most probable partitions in H_n (line 15).

Our implementation of Luo et al.'s search algorithm differs from their original algorithm only in terms of the number of pruning strategies adopted. Specifically, Luo et al. introduce a number of heuristics to prune the search space in order to speed up the search. We employ only the beam search heuristic, with a beam size that is five times larger than theirs. Our larger beam size, together with the fact that we do not use other pruning strategies, implies that we are searching through a larger part of the space than them, thus potentially yielding better partitions.

3 Haghighi and Klein's Coreference Model

To gauge the performance of our model, we compare it with a Bayesian model for unsupervised coreference resolution that was recently proposed by Haghighi and Klein (2007). In this section, we will give an overview of their model, discuss its weaknesses and propose three modifications to the model.

3.1 Notations

For consistency, we follow Haghighi and Klein's (H&K) notations. **Z** is the set of random variables

indeed partitions. This is desirable, as there is no reason for us to put non-zero probability mass on invalid clusterings.

that refer to (indices of) entities. ϕ_z is the set of parameters associated with entity z. ϕ is the entire set of model parameters, which includes all the ϕ_z's. Finally, $\mathbf{X}$ is the set of observed variables (e.g., the head of a mention). Given a document, the goal is to find the most probable assignment of entity indices to its mentions given the observed values. In other words, we want to maximize $P(\mathbf{Z}|\mathbf{X})$. In a Bayesian approach, we compute this probability by integrating out all the parameters. Specifically,

$$P(\mathbf{Z}|\mathbf{X}) = \int P(\mathbf{Z}|\mathbf{X}, \phi)P(\phi|\mathbf{X})d\phi.$$

3.2 The Original H&K Model

The original H&K model is composed of a set of models: the *basic* model and two other models (namely, the *pronoun head* model and the *salience* model) that aim to improve the basic model.[5]

3.2.1 Basic Model

The basic model generates a mention in a two-step process. First, an entity index is chosen according to an *entity distribution*, and then the head of the mention is generated given the entity index based on an entity-specific *head distribution*. Here, we assume that (1) all heads H are observed and (2) a mention is represented solely by its head noun, so nothing other than the head is generated. Furthermore, we assume that the head distribution is drawn from a symmetric Dirichlet with concentration λ_H. Hence,

$$P(H_{i,j} = h|\mathbf{Z}, \mathbf{H}^{-i,j}) \propto n_{h,z} + \lambda_H$$

where $H_{i,j}$ is the head of mention j in document i, and $n_{h,z}$ is the number of times head h is emitted by entity index z in $(\mathbf{Z}, \mathbf{H}^{-i,j})$.[6] On the other hand, since the number of entities in a document is *not* known a priori, we draw the entity distribution from a Dirichlet *process* with concentration α, effectively yielding a model with an infinite number of mixture components. Using the Chinese restaurant process representation (see Teh et al. (2006)),

$$P(Z_{ij} = z|\mathbf{Z}^{-i,j}) \propto \begin{cases} \alpha & , \quad \text{if } z = z_{new} \\ n_z & , \quad \text{otherwise} \end{cases}$$

[5]H&K also present a cross-document coreference model, but since it focuses primarily on cross-document coreference and improves within-document coreference performance by only 1.5% in F-score, we will not consider this model here.

[6]$\mathbf{H}^{-i,j}$ is used as a shorthand for $\mathbf{H} - \{H_{i,j}\}$.

where n_z is the number of mentions in $\mathbf{Z}^{-i,j}$ labeled with entity index z, and z_{new} is a new entity index not already in $\mathbf{Z}^{-i,j}$. To perform inference, we use Gibbs sampling (Geman and Geman, 1984) to generate samples from this conditional distribution:

$$P(Z_{i,j}|\mathbf{Z}^{-i,j}, \mathbf{H}) \propto P(Z_{i,j}|\mathbf{Z}^{-i,j})P(H_{i,j}|\mathbf{Z}, \mathbf{H}^{-i,j})$$

where the two distributions on the right are defined as above. Starting with a random assignment of entity indices to mentions, the Gibbs sampler iteratively re-samples an entity index according to this posterior distribution given the current assignment.

3.2.2 Pronoun Head Model

Not surprisingly, the basic model is too simplistic: it has a strong tendency to assign the same entity index to mentions having the same head. This is particularly inappropriate for pronouns. Hence, we need a different model for generating pronouns.

Before introducing this pronoun head model, we need to augment the set of entity-specific parameters, which currently contains only a distribution over heads (ϕ_Z^h). Specifically, we add distributions ϕ_Z^t, ϕ_Z^g, and ϕ_Z^n over entity *properties*: ϕ_Z^t is a distribution over semantic types (PER, ORG, LOC, MISC), ϕ_Z^g over gender (MALE, FEMALE, EITHER, NEUTER), and ϕ_Z^n over number (SG, PL). We assume that each of these distributions is drawn from a symmetric Dirichlet. A small concentration parameter is used, since each entity should have a dominating value for each of these properties.

Now, to estimate ϕ_Z^t, ϕ_Z^g, and ϕ_Z^n, we need to know the gender, number, and semantic type of each mention. For some mentions (e.g., "he"), these properties are easy to compute; for others (e.g., "it"), they are not. Whenever a mention has unobserved properties, we need to fill in the missing values. We could resort to sampling, but sampling these properties is fairly inefficient. So, following H&K, we keep soft counts for each of these properties and use them rather than perform hard sampling.

When an entity z generates a pronoun h using the pronoun head model,[7] it first generates a gender g, a number n, and a semantic type t independently from the distributions ϕ_z^g, ϕ_z^n, and ϕ_z^t; and then generates h using the distribution $P(H = h|G = g, N =$

[7]While pronouns are generated by this pronoun head model, names and nominals continue to be handled by the basic model.

$n, T = t, \theta)$. Note that this last distribution is a global distribution that is independent of the chosen entity index. θ is a parameter drawn from a symmetric Dirichlet (with concentration λ_P) that encodes our prior knowledge of the relationship between a semantic type and a pronoun. For instance, given the type PERSON, there is a higher probability of generating "he" than "it". As a result, we maintain a list of compatible semantic types for each pronoun, and give a pronoun a count of $(1 + \lambda_P)$ if it is compatible with the drawn semantic type; otherwise, we give it a count of λ_P. In essence, we use this prior to prefer the generation of pronouns that are compatible with the chosen semantic type.

3.2.3 Salience Model

Pronouns typically refer to salient entities, so the basic model could be improved by incorporating salience. We start by assuming that each entity has an activity score that is initially set to zero. Given a set of mentions and an assignment of entity indices to mentions, $\mathbf{Z}$, we process the mentions in a left-to-right manner. When a mention, m, is encountered, we multiply the activity score of each entity by 0.5 and add one to the activity score of the entity to which m belongs. This captures the intuitive notion that frequency and recency both play a role in determining salience. Next, we rank the entities based on their activity scores and discretize the ranks into five "salience" buckets S: TOP (1), HIGH (2–3), MID (4–6), LOW (7+), and NONE. Finally, this salience information is used to modify the entity distribution:[8]

$$P(Z_{ij} = z|\mathbf{Z}^{-i,j}) \propto n_z \cdot P(M_{i,j}|S_{i,j}, \mathbf{Z})$$

where $S_{i,j}$ is the salience value of the jth mention in document i, and $M_{i,j}$ is its mention type, which can take on one of three values: pronoun, name, and nominal. $P(M_{i,j}|S_{i,j}, \mathbf{Z})$, the distribution of mention type given salience, was computed from H&K's development corpus (see Table 2). According to the table, pronouns are preferred for salient entities, whereas names and nominals are preferred for entities that are less active.

[8]Rather than having just one probability term on the right hand side of the sampling equation, H&K actually have a product of probability terms, one for each mention that appears later than mention j in the given document. However, they acknowledge that having the product makes sampling inefficient, and decided to simplify the equation to this form in their evaluation.

Salience Feature	Pronoun	Name	Nominal
TOP	0.75	0.17	0.08
HIGH	0.55	0.28	0.17
MID	0.39	0.40	0.21
LOW	0.20	0.45	0.35
NONE	0.00	0.88	0.12

Table 2: Posterior distribution of mention type given salience (taken from Haghighi and Klein (2007))

3.3 Modifications to the H&K Model

Next, we discuss the potential weaknesses of H&K's model and propose three modifications to it.

Relaxed head generation. The basic model focuses on head matching, and is therefore likely to (incorrectly) posit *the large airport* and *the small airport* as coreferent, for instance. In fact, head matching is a relatively inaccurate indicator of coreference, in comparison to the "strong coreference indicators" shown in the first three rows of Table 1. To improve H&K's model, we replace head matching with these three strong indicators as follows. Given a document, we assign each of its mentions a *head index*, such that two mentions have the same head index if and only if at least one of the three strong indicators returns a value of True. Now, instead of generating a head, the head model generates a head index, thus increasing the likelihood that aliases are assigned the same entity index, for instance. Note that this modification is applied only to the basic model. In particular, pronoun generation continues to be handled by the pronoun head model and will not be affected. We hypothesize that this modification would improve precision, as the strong indicators are presumably more precise than head match.

Agreement constraints. While the pronoun head model naturally prefers that a pronoun be generated by an entity whose gender and number are compatible with those of the pronoun, the entity (index) that is re-sampled for a pronoun according to the sampling equation for $P(Z_{i,j}|\mathbf{Z}^{-i,j}, \mathbf{H})$ may still not be compatible with the pronoun with respect to gender and number. The reason is that an entity index is assigned based not only on the head distribution but also on the entity distribution. Since entities with many mentions are preferable to those with few mentions, it is possible for the model to favor the assignment of a grammatically incompatible entity (index) to a pronoun if the entity is sufficiently large. To eliminate this possibility, we enforce the agreement constraints at the global level. Specifically, we sample an entity index for a given mention with a non-zero probability if and only if the corresponding entity and the head of the mention agree in gender and number. We hypothesize that this modification would improve precision.

Pronoun-only salience. In Section 3.2.3, we motivate the need for salience using pronouns only, since proper names can to a large extent be resolved using string-matching facilities and are not particularly sensitive to salience. Nominals (especially definite descriptions), though more sensitive to salience than names, can also be resolved by simple string-matching heuristics in many cases (Vieira and Poesio, 2000; Strube et al., 2002). Hence, we hypothesize that the use of salience for names and nominals would adversely affect their resolution performance, as incorporating salience could diminish the role of string match in the resolution process, according to the sampling equations. Consequently, we modify H&K's model by limiting the application of salience to the resolution of pronouns only. We hypothesize that this change would improve precision.

4 Evaluation

4.1 Experimental Setup

To evaluate our EM-based model and H&K's model, we use the ACE 2003 coreference corpus, which is composed of three sections: Broadcast News (BNEWS), Newswire (NWIRE), and Newspaper (NPAPER). Each section is in turn composed of a training set and a test set. Due to space limitations, we will present evaluation results only for the test sets of BNEWS and NWIRE, but verified that the same performance trends can be observed on NPAPER as well. Unlike H&K, who report results using only true mentions (extracted from the answer keys), we show results for true mentions as well as system mentions that were extracted by an in-house noun phrase chunker. The relevant statistics of the BNEWS and NWIRE test sets are shown in Table 3.

	BNEWS	NWIRE
Number of documents	51	29
Number of true mentions	2608	2630
Number of system mentions	5424	5197

Table 3: Statistics of the BNEWS and NWIRE test sets

Scoring programs. To score the output of the coreference models, we employ the commonly-used MUC scoring program (Vilain et al., 1995) and the recently-developed CEAF scoring program (Luo, 2005). In the MUC scorer, recall is computed as the percentage of coreference *links* in the reference partition that appear in the system partition; precision is defined in a similar fashion as recall, except that the roles of the reference partition and the system partition are reversed. As a *link-based* scoring program, the MUC scorer (1) does not reward successful identification of singleton entities and (2) tends to underpenalize partitions that have too few entities. The *entity-based* CEAF scorer was proposed in response to these two weaknesses. Specifically, it operates by computing the optimal alignment between the set of reference entities and the set of system entities. CEAF precision and recall are both positively correlated with the score of this optimal alignment, which is computed by summing over each aligned entity pair the number of mentions that appear in both entities of that pair. As a consequence, a system that proposes too many entities or too few entities will have low precision and recall.

Parameter initialization. We use a small amount of labeled data for parameter initialization for the two models. Specifically, for evaluations on the BNEWS test data, we use as labeled data one randomly-chosen document from the BNEWS training set, which has 58 true mentions and 102 system mentions. Similarly for NWIRE, where the chosen document has 42 true mentions and 72 system mentions. For our model, we use the labeled document to initialize the parameters. Also, we set N (the number of most probable partitions) to 50 and δ (the start penalty used in the Bell tree) to 0.8, the latter being recommended by Luo et al. (2004).

For H&K's model, we use the labeled data to tune the concentration parameter α. While H&K set α to 0.4 without much explanation, a moment's thought reveals that the choice of α should reflect the fraction of mentions that appear in a singleton cluster. We therefore estimate this value from the labeled document, yielding 0.4 for true mentions (which is consistent with H&K's choice) and 0.7 for system mentions. The remaining parameters, the λ's, are all set to e^{-4}, following H&K. In addition, as is commonly done in Bayesian approaches, we do not sample entities directly from the conditional distribution $P(\mathbf{Z}|\mathbf{X})$; rather, we sample from this distribution raised to the power $\exp \frac{ci}{k-1}$, where c=1.5, i is the current iteration number that starts at 0, and k (the number of sampling iterations) is set to 20. Finally, due to sampling and the fact that the initial assignment of entity indices to mentions is random, all the reported results for H&K's model are averaged over five runs.

4.2 Results and Discussions

The Heuristic baseline. As our first baseline, we employ a simple rule-based system that posits two mentions as coreferent if and only if at least one of the three strong coreference indicators listed in Table 1 returns True. Results of this baseline, reported in terms of recall (R), precision (P), and F-score (F) using the MUC scorer and the CEAF scorer, are shown in row 1 of Tables 4 and 5, respectively. Each row in these tables shows performance using true mentions and system mentions for the BNEWS and NWIRE data sets. As we can see, (1) recall is generally low, since this simple heuristic can only identify a small fraction of the coreference relations; (2) CEAF recall for true mentions is by definition equal to CEAF precision; (3) CEAF recall is consistently higher than MUC recall, since CEAF also rewards successful identification of non-coreference relations; and (4) precision for true mentions is higher than that for system mentions, since the number of non-coreferent pairs that satisfy the heuristic is larger for system mentions.

The Degenerate EM baseline. Our second baseline is obtained by running only one iteration of our EM-based coreference model. Specifically, it starts with the M-step by initializing the model parameters using the labeled document, and ends with the E-step by applying the resulting model (in combination with the Bell tree search algorithm) to obtain the most probable coreference partition for each test document. Since there is no parameter re-estimation, this baseline is effectively a purely supervised system trained on one (labeled) document.

Results are shown in row 2 of Tables 4 and 5. As we can see, recall is consistently much higher than precision, suggesting that the model has produced fewer entities than it should. Perhaps more interestingly, in comparison to the Heuristic baseline, Degenerate EM performs consistently worse according to CEAF but generally better according to MUC. This discrepancy stems from the aforementioned properties that MUC under-penalizes partitions with too few entities, whereas CEAF lowers both recall and precision when given such partitions.

Our EM-based coreference model. Our model operates in the same way as the Degenerate EM baseline, except that EM is run until convergence, with the test set being used as unlabeled data for parameter re-estimation. Any performance difference between our model and Degenerate EM can thus be attributed to EM's exploitation of the unlabeled data.

Results of our model are shown in row 3 of Tables 4 and 5. In comparison to Degenerate EM, MUC F-score increases by 4-5% for BNEWS and 4-21% for NWIRE; CEAF F-score increases even more dramatically, by 12-16% for BNEWS and 27-32% for NWIRE. Improvements stem primarily from large gains in precision and comparatively smaller loss in recall. Such improvements suggest that our model has effectively exploited the unlabeled data.

In comparison to the Heuristic baseline, we see fairly large increases in both recall and precision when system mentions are used, and as a result, F-score improves substantially by 5-15%. When true mentions are used, we again see large increases in CEAF recall and precision; MUC recall also increases considerably, but such gains are accompanied by a small loss in MUC precision. Overall, F-score for true mentions increases by 3-22%.

The Original H&K model. We use as our third baseline the Original H&K model (see Section 3.2). Results of this model are shown in row 4 of Tables 4 and 5.[9] Overall, it underperforms our model by 6-16% in MUC F-score and 8-11% in CEAF F-score, due primarily to considerable drop in both recall and precision in all cases.

The Modified H&K model. Next, we incorporate our three modifications into the Original H&K baseline one after the other. Results are shown in rows 5-7 of Tables 4 and 5. Several points deserve men-

[9]The H&K results shown here are not directly comparable with those reported in Haghighi and Klein (2007), since H&K evaluated their system on the ACE 2004 coreference corpus.

		Broadcast News (BNEWS)						Newswire (NWIRE)					
		True Mentions			System Mentions			True Mentions			System Mentions		
	Experiments	R	P	F	R	P	F	R	P	F	R	P	F
1	Heuristic Baseline	27.8	72.0	40.1	30.9	44.3	36.4	31.2	70.3	43.3	36.3	53.4	43.2
2	Degenerate EM Baseline	63.6	53.1	57.9	70.8	36.3	48.0	64.5	42.6	51.3	69.0	25.1	36.8
3	Our EM-based Model	56.1	71.4	**62.8**	42.4	66.0	51.6	47.0	68.3	**55.7**	55.2	60.6	**57.8**
4	Haghighi and Klein Baseline	49.4	60.2	54.3	50.8	40.7	45.2	44.7	55.5	49.5	43.0	40.9	41.9
5	+ Relaxed Head Generation	53.0	65.4	58.6	48.3	45.7	47.0	45.1	62.5	52.4	40.9	50.0	45.0
6	+ Agreement Constraints	53.6	68.7	60.2	50.4	47.5	48.9	44.6	63.7	52.5	41.7	51.2	46.0
7	+ Pronoun-only Salience	56.8	68.3	62.0	52.2	53.0	**52.6**	46.8	66.2	54.8	44.3	57.3	50.0
8	Fully Supervised Model	53.7	70.8	61.1	53.0	70.3	60.4	52.0	69.6	59.6	53.1	70.5	60.6

Table 4: Results obtained using the MUC scoring program for the Broadcast News and Newswire data sets

		Broadcast News (BNEWS)						Newswire (NWIRE)					
		True Mentions			System Mentions			True Mentions			System Mentions		
	Experiments	R	P	F	R	P	F	R	P	F	R	P	F
1	Heuristic Baseline	53.7	53.7	53.7	54.3	43.7	48.4	58.0	58.0	58.0	58.9	50.2	54.2
2	Degenerate EM Baseline	48.6	48.6	48.6	49.5	32.7	39.4	34.1	34.1	34.1	38.2	22.0	27.9
3	Our EM-based Model	60.9	60.9	**60.9**	57.0	54.6	**55.7**	61.2	61.2	**61.2**	62.9	56.5	**59.6**
4	Haghighi and Klein Baseline	50.3	50.3	50.3	53.2	39.3	45.2	53.5	53.5	53.5	54.5	44.2	48.8
5	+ Relaxed Head Generation	53.4	53.4	53.4	53.4	42.8	47.5	58.1	58.1	58.1	55.9	49.8	52.6
6	+ Agreement Constraints	59.2	59.2	59.2	57.8	46.3	51.4	59.9	59.9	59.9	57.9	51.5	54.5
7	+ Pronoun-only Salience	60.7	60.7	60.7	59.2	50.8	54.7	60.9	60.9	60.9	59.4	55.6	57.4
8	Fully Supervised Model	61.3	61.3	61.3	63.4	60.3	61.8	64.2	64.2	64.2	65.8	63.2	64.5

Table 5: Results obtained using the CEAF scoring program for the Broadcast News and Newswire data sets

tioning. First, the addition of each modification improves the F-score for both true and system mentions in both data sets using both scorers. These results provide suggestive evidence that our modifications are highly beneficial. The three modifications, when applied in combination, improve Original H&K substantially by 5-8% in MUC F-score and 7-10% in CEAF F-score, yielding results that compare favorably to those of our model in almost all cases.

Second, the use of agreement constraints yields larger improvements with CEAF than with MUC. This discrepancy can be attributed to the fact that CEAF rewards the correct identification of non-coreference relations, whereas MUC does not. Since agreement constraints are intended primarily for disallowing coreference, they contribute to the successful identification of non-coreference relations and as a result yield gains in CEAF recall and precision.

Third, the results are largely consistent with our hypothesis that these modifications enhance precision. Together, they improve the precision of the Original H&K baseline by 8-16% (MUC) and 7-12% (CEAF), yielding a coreference model that compares favorably with our EM-based approach.

Comparison with a supervised model. Finally, we compare our EM-based model with a fully supervised coreference resolver. Inspired by state-of-the-art resolvers, we create our supervised classification model by training a discriminative learner (the C4.5 decision tree induction system (Quinlan, 1993)) with a diverse set of features (the 34 features described in Ng (2007)) on a large training set (the entire ACE 2003 coreference training corpus), and cluster using the Bell tree search algorithm. The fully supervised results shown in row 8 of Tables 4 and 5 suggest that our EM-based model has room for improvements, especially when system mentions are used.

5 Conclusions

We have presented a generative model for unsupervised coreference resolution that views coreference as an EM clustering process. Experimental results indicate that our model outperforms Haghighi and Klein's (2007) coreference model by a large margin on the ACE data sets and compares favorably to a modified version of their model. Despite these improvements, its performance is still not comparable to that of a fully supervised coreference resolver.

A natural way to extend these unsupervised coreference models is to incorporate additional linguistic knowledge sources, such as those employed by our fully supervised resolver. However, feature engineering is in general more difficult for generative models than for discriminative models, as the former typically require non-overlapping features. We plan to explore this possibility in future work.

Acknowledgments

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

References

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*, pages 92–100.

Colin Cherry and Shane Bergsma. 2005. An expectation maximization approach to pronoun resolution. In *Proceedings of CoNLL*, pages 88–95.

Hal Daumé III and Daniel Marcu. 2005. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of HLT/EMNLP*, pages 97–104.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Pascal Denis and Jason Baldridge. 2007. Global, joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of NAACL/HLT*, pages 236–243.

Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Aria Haghighi and Dan Klein. 2007. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the ACL*, pages 848–855.

Heng Ji, David Westbrook, and Ralph Grishman. 2005. Using semantic relations to refine coreference decisions. In *Proceedings of HLT/EMNLP*, pages 17–24.

Andrew Kehler, Douglas Appelt, Lara Taylor, and Aleksandr Simma. 2004. Competitive self-trained pronoun interpretation In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 33–36.

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proceedings of the ACL*, pages 135–142.

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

Christoph Müller, Stefan Rapp, and Michael Strube. 2002. Applying co-training to reference resolution. In *Proceedings of the ACL*, pages 352–359.

Vincent Ng. 2007. Shallow semantics for coreference resolution. In *Proceedings of IJCAI*, pages 1689–1694.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the ACL*, pages 104–111.

Vincent Ng and Claire Cardie. 2003. Weakly supervised natural language learning without redundant views. In *HLT-NAACL: Main Proceedings*, pages 173–180.

Simone Paolo Ponzetto and Michael Strube. 2006. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of HLT/NAACL*, pages 192–199.

J. Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Michael Strube, Stefan Rapp, and Christoph Müller. 2002. The influence of minimum edit distance on reference resolution. In *Proceedings of EMNLP*, pages 312–319.

Yee Whye Teh, Michael Jordan, Matthew Beal, and David Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1527–1554.

Renata Vieira and Massimo Poesio. 2000. An empirically-based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of MUC-6*, pages 45–52.

Xiaofeng Yang, GuoDong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competitive learning approach. In *Proceedings of the ACL*, pages 176–183.