Joint Unsupervised Coreference Resolution with Markov Logic

Hoifung Poon **Pedro Domingos**
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{hoifung,pedrod}@cs.washington.edu

Abstract

Machine learning approaches to coreference resolution are typically supervised, and require expensive labeled data. Some unsupervised approaches have been proposed (e.g., Haghighi and Klein (2007)), but they are less accurate. In this paper, we present the first unsupervised approach that is competitive with supervised ones. This is made possible by performing joint inference across mentions, in contrast to the pairwise classification typically used in supervised methods, and by using Markov logic as a representation language, which enables us to easily express relations like apposition and predicate nominals. On MUC and ACE datasets, our model outperforms Haghigi and Klein's one using only a fraction of the training data, and often matches or exceeds the accuracy of state-of-the-art supervised models.

1 Introduction

The goal of coreference resolution is to identify *mentions* (typically noun phrases) that refer to the same *entities*. This is a key subtask in many NLP applications, including information extraction, question answering, machine translation, and others. Supervised learning approaches treat the problem as one of classification: for each pair of mentions, predict whether they corefer or not (e.g., McCallum & Wellner (2005)). While successful, these approaches require labeled training data, consisting of mention pairs and the correct decisions for them. This limits their applicability.

Unsupervised approaches are attractive due to the availability of large quantities of unlabeled text. However, unsupervised coreference resolution is much more difficult. Haghighi and Klein's (2007) model, the most sophisticated to date, still lags supervised ones by a substantial margin. Extending it appears difficult, due to the limitations of its Dirichlet process-based representation.

The lack of label information in unsupervised coreference resolution can potentially be overcome by performing joint inference, which leverages the "easy" decisions to help make related "hard" ones. Relations that have been exploited in supervised coreference resolution include transitivity (McCallum & Wellner, 2005) and anaphoricity (Denis & Baldridge, 2007). However, there is little work to date on joint inference for unsupervised resolution.

We address this problem using Markov logic, a powerful and flexible language that combines probabilistic graphical models and first-order logic (Richardson & Domingos, 2006). Markov logic allows us to easily build models involving relations among mentions, like apposition and predicate nominals. By extending the state-of-the-art algorithms for inference and learning, we developed the first general-purpose unsupervised learning algorithm for Markov logic, and applied it to unsupervised coreference resolution.

We test our approach on standard MUC and ACE datasets. Our basic model, trained on a minimum of data, suffices to outperform Haghighi and Klein's (2007) one. Our full model, using apposition and other relations for joint inference, is often as accurate as the best supervised models, or more.

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 650–659,
Honolulu, October 2008.

We begin by reviewing the necessary background on Markov logic. We then describe our Markov logic network for joint unsupervised coreference resolution, and the learning and inference algorithms we used. Finally, we present our experiments and results.

2 Related Work

Most existing supervised learning approaches for coreference resolution are suboptimal since they resolve each mention pair independently, only imposing transitivity in postprocessing (Ng, 2005). Moreover, many of them break up the resolution step into subtasks (e.g., first determine whether a mention is anaphoric, then classify whether it is coreferent with an antecedent), which further forsakes opportunities for joint inference that have been shown to be helpful (Poon & Domingos, 2007). Using graph partitioning, McCallum & Wellner (2005) incorporated transitivity into pairwise classification and achieved the state-of-the-art result on the MUC-6 dataset, but their approach can only leverage one binary relation at a time, not arbitrary relations among mentions. Denis & Baldridge (2007) determined anaphoricity and pairwise classification jointly using integer programming, but they did not incorporate transitivity or other relations.

While potentially more appealing, unsupervised learning is very challenging, and unsupervised coreference resolution systems are still rare to this date. Prior to our work, the best performance in unsupervised coreference resolution was achieved by Haghighi & Klein (2007), using a nonparametric Bayesian model based on hierarchical Dirichlet processes. At the heart of their system is a mixture model with a few linguistically motivated features such as head words, entity properties and salience. Their approach is a major step forward in unsupervised coreference resolution, but extending it is challenging. The main advantage of Dirichlet processes is that they are exchangeable, allowing parameters to be integrated out, but Haghighi and Klein forgo this when they introduce salience. Their model thus requires Gibbs sampling over both assignments and parameters, which can be very expensive. Haghighi and Klein circumvent this by making approximations that potentially hurt accuracy. At the same time, the Dirichlet process prior favors skewed cluster sizes and a number of clusters that grows logarithmically with the number of data points, neither of which seems generally appropriate for coreference resolution.

Further, deterministic or strong non-deterministic dependencies cause Gibbs sampling to break down (Poon & Domingos, 2006), making it difficult to leverage many linguistic regularities. For example, apposition (as in "Bill Gates, the chairman of Microsoft") suggests coreference, and thus the two mentions it relates should always be placed in the same cluster. However, Gibbs sampling can only move one mention at a time from one cluster to another, and this is unlikely to happen, because it would require breaking the apposition rule. Blocked sampling can alleviate this problem by sampling multiple mentions together, but it requires that the block size be predetermined to a small fixed number. When we incorporate apposition and other regularities the blocks can become arbitrarily large, making this infeasible. For example, suppose we also want to leverage predicate nominals (i.e., the subject and the predicating noun of a copular verb are likely coreferent). Then a sentence like "He is Bill Gates, the chairman of Microsoft" requires a block of four mentions: "He", "Bill Gates", "the chairman of Microsoft", and "Bill Gates, the chairman of Microsoft". Similar difficulties occur with other inference methods. Thus, extending Haghighi and Klein's model to include richer linguistic features is a challenging problem.

Our approach is instead based on Markov logic, a powerful representation for joint inference with uncertainty (Richardson & Domingos, 2006). Like Haghighi and Klein's, our model is cluster-based rather than pairwise, and implicitly imposes transitivity. We do not predetermine anaphoricity of a mention, but rather fuse it into the integrated resolution process. As a result, our model is inherently joint among mentions and subtasks. It shares several features with Haghighi & Klein's model, but removes or refines features where we believe it is appropriate to. Most importantly, our model leverages apposition and predicate nominals, which Haghighi & Klein did not use. We show that this can be done very easily in our framework, and yet results in very substantial accuracy gains.

It is worth noticing that Markov logic is also well suited for joint inference in supervised systems (e.g., transitivity, which took McCallum & Wellner (2005) nontrivial effort to incorporate, can be handled in Markov logic with the addition of a single formula (Poon & Domingos, 2008)).

3 Markov Logic

In many NLP applications, there exist rich relations among objects, and recent work in statistical relational learning (Getoor & Taskar, 2007) and structured prediction (Bakir *et al.*, 2007) has shown that leveraging these can greatly improve accuracy. One of the most powerful representations for joint inference is Markov logic, a probabilistic extension of first-order logic (Richardson & Domingos, 2006). A *Markov logic network (MLN)* is a set of weighted first-order clauses. Together with a set of constants, it defines a Markov network with one node per ground atom and one feature per ground clause. The weight of a feature is the weight of the first-order clause that originated it. The probability of a state x in such a network is given by $P(x) = (1/Z) \exp\left(\sum_i w_i f_i(x)\right)$, where Z is a normalization constant, w_i is the weight of the ith clause, $f_i = 1$ if the ith clause is true, and $f_i = 0$ otherwise.

Markov logic makes it possible to compactly specify probability distributions over complex relational domains. Efficient inference can be performed using MC-SAT (Poon & Domingos, 2006). MC-SAT is a "slice sampling" Markov chain Monte Carlo algorithm. Slice sampling introduces auxiliary variables u that decouple the original ones x, and alternately samples u conditioned on x and vice-versa. To sample from the slice (the set of states x consistent with the current u), MC-SAT calls SampleSAT (Wei *et al.*, 2004), which uses a combination of satisfiability testing and simulated annealing. The advantage of using a satisfiability solver (WalkSAT) is that it efficiently finds isolated modes in the distribution, and as a result the Markov chain mixes very rapidly. The slice sampling scheme ensures that detailed balance is (approximately) preserved. MC-SAT is orders of magnitude faster than previous MCMC algorithms like Gibbs sampling, making efficient sampling possible on a scale that was previously out of reach.

Algorithm 1 MC-SAT(*clauses, weights, num_samples*)

$x^{(0)} \leftarrow$ Satisfy(hard *clauses*)
for $i \leftarrow 1$ to *num_samples* **do**
  $M \leftarrow \emptyset$
  for all $c_k \in$ *clauses* satisfied by $x^{(i-1)}$ **do**
    With probability $1 - e^{-w_k}$ add c_k to M
  end for
  Sample $x^{(i)} \sim \mathcal{U}_{SAT(M)}$
end for

Algorithm 1 gives pseudo-code for MC-SAT. At iteration $i - 1$, the factor ϕ_k for clause c_k is either e^{w_k} if c_k is satisfied in $x^{(i-1)}$, or 1 otherwise. MC-SAT first samples the auxiliary variable u_k uniformly from $(0, \phi_k)$, then samples a new state uniformly from the set of states that satisfy $\phi'_k \geq u_k$ for all k (the slice). Equivalently, for each k, with probability $1 - e^{-w_k}$ the next state must satisfy c_k. In general, we can factorize the probability distribution in any way that facilitates inference, sample the u_k's, and make sure that the next state is drawn uniformly from solutions that satisfy $\phi'_k \geq u_k$ for all factors.

MC-SAT, like most existing relational inference algorithms, grounds all predicates and clauses, thus requiring memory and time exponential in the predicate and clause arities. We developed a general method for producing a "lazy" version of relational inference algorithms (Poon & Domingos, 2008), which carries exactly the same inference steps as the original algorithm, but only maintains a small subset of "active" predicates/clauses, grounding more as needed. We showed that Lazy-MC-SAT, the lazy version of MC-SAT, reduced memory and time by orders of magnitude in several domains. We use Lazy-MC-SAT in this paper.

Supervised learning for Markov logic maximizes the conditional log-likelihood $L(x, y) = \log P(Y = y | X = x)$, where Y represents the non-evidence predicates, X the evidence predicates, and x, y their values in the training data. For simplicity, from now on we omit X, whose values are fixed and always conditioned on. The optimization problem is convex and a global optimum can be found using gradient

descent, with the gradient being

$$\begin{aligned}\frac{\partial}{\partial w_i}L(y) &= n_i(y) - \sum_{y'} P(Y = y')n_i(y') \\ &= n_i(y) - E_Y[n_i].\end{aligned}$$

where n_i is the number of true groundings of clause i. The expected count can be approximated as

$$E_Y[n_i] \approx \frac{1}{N}\sum_{k=1}^{N} n_i(y_k)$$

where y_k are samples generated by MC-SAT. To combat overfitting, a Gaussian prior is imposed on all weights.

In practice, it is difficult to tune the learning rate for gradient descent, especially when the number of groundings varies widely among clauses. Lowd & Domingos (2007) used a preconditioned scaled conjugate gradient algorithm (PSCG) to address this problem. This estimates the optimal step size in each step as

$$\alpha = \frac{-d^T g}{d^T H d + \lambda d^T d}.$$

where g is the gradient, d the conjugate update direction, and λ a parameter that is automatically tuned to trade off second-order information with gradient descent. H is the Hessian matrix, with the (i, j)th entry being

$$\begin{aligned}\frac{\partial^2}{\partial w_i \partial w_j}L(y) &= E_Y[n_i] \cdot E_Y[n_j] - E_Y[n_i \cdot n_j] \\ &= -Cov_Y[n_i, n_j].\end{aligned}$$

The Hessian can be approximated with the same samples used for the gradient. Its negative inverse diagonal is used as the preconditioner.[1]

The open-source Alchemy package (Kok *et al.*, 2007) provides implementations of existing algorithms for Markov logic. In Section 5, we develop the first general-purpose unsupervised learning algorithm for Markov logic by extending the existing algorithms to handle hidden predicates.[2]

[1] Lowd & Domingos showed that α can be computed more efficiently, without explicitly approximating or storing the Hessian. Readers are referred to their paper for details.

[2] Alchemy includes a discriminative EM algorithm, but it assumes that only a few values are missing, and cannot handle completely hidden predicates. Kok & Domingos (2007) applied Markov logic to relational clustering, but they used hard EM.

4 An MLN for Joint Unsupervised Coreference Resolution

In this section, we present our MLN for joint unsupervised coreference resolution. Our model deviates from Haghighi & Klein's (2007) in several important ways. First, our MLN does not model saliences for proper nouns or nominals, as their influence is marginal compared to other features; for pronoun salience, it uses a more intuitive and simpler definition based on distance, and incorporated it as a prior. Another difference is in identifying heads. For the ACE datasets, Haghighi and Klein used the gold heads; for the MUC-6 dataset, where labels are not available, they crudely picked the rightmost token in a mention. We show that a better way is to determine the heads using head rules in a parser. This improves resolution accuracy and is always applicable. Crucially, our MLN leverages syntactic relations such as apposition and predicate nominals, which are not used by Haghighi and Klein. In our approach, what it takes is just adding two formulas to the MLN.

As common in previous work, we assume that true mention boundaries are given. We do not assume any other labeled information. In particular, we do not assume gold name entity recognition (NER) labels, and unlike Haghighi & Klein (2007), we do not assume gold mention types (for ACE datasets, they also used gold head words). We determined the head of a mention either by taking its rightmost token, or by using the head rules in a parser. We detected pronouns using a list.

4.1 Base MLN

The main query predicate is $\texttt{InClust}(\texttt{m}, \texttt{c!})$, which is true iff mention `m` is in cluster `c`. The "`t!`" notation signifies that for each `m`, this predicate is true for a unique value of `c`. The main evidence predicate is $\texttt{Head}(\texttt{m}, \texttt{t!})$, where `m` is a mention and `t` a token, and which is true iff `t` is the head of `m`. A key component in our MLN is a simple head mixture model, where the mixture component priors are represented by the unit clause

$$\texttt{InClust}(+\texttt{m}, +\texttt{c})$$

and the head distribution is represented by the *head prediction rule*

$$\texttt{InClust}(\texttt{m}, +\texttt{c}) \wedge \texttt{Head}(\texttt{m}, +\texttt{t}).$$

All free variables are implicitly universally quantified. The "+" notation signifies that the MLN contains an instance of the rule, with a separate weight, for each value combination of the variables with a plus sign.

By convention, at each inference step we name each non-empty cluster after the earliest mention it contains. This helps break the symmetry among mentions, which otherwise produces multiple optima and makes learning unnecessarily harder. To encourage clustering, we impose an exponential prior on the number of non-empty clusters with weight -1.

The above model only clusters mentions with the same head, and does not work well for pronouns. To address this, we introduce the predicate $\texttt{IsPrn}(\texttt{m})$, which is true iff the mention m is a pronoun, and adapt the head prediction rule as follows:

$$\neg\texttt{IsPrn}(\texttt{m}) \wedge \texttt{InClust}(\texttt{m}, +\texttt{c}) \wedge \texttt{Head}(\texttt{m}, +\texttt{t})$$

This is always false when m is a pronoun, and thus applies only to non-pronouns.

Pronouns tend to resolve with mentions that are semantically compatible with them. Thus we introduce predicates that represent entity type, number, and gender: $\texttt{Type}(\texttt{x}, \texttt{e!})$, $\texttt{Number}(\texttt{x}, \texttt{n!})$, $\texttt{Gender}(\texttt{x}, \texttt{g!})$, where x can be either a cluster or mention, $\texttt{e} \in \{\texttt{Person}, \texttt{Organization}, \texttt{Location}, \texttt{Other}\}$, $\texttt{n} \in \{\texttt{Singular}, \texttt{Plural}\}$ and $\texttt{g} \in \{\texttt{Male}, \texttt{Female}, \texttt{Neuter}\}$. Many of these are known for pronouns, and some can be inferred from simple linguistic cues (e.g., "Ms. Galen" is a singular female person, while "XYZ Corp." is an organization).[3] Entity type assignment is represented by the unit clause

$$\texttt{Type}(+\texttt{x}, +\texttt{e})$$

and similarly for number and gender. A mention should agree with its cluster in entity type. This is ensured by the hard rule (which has infinite weight and must be satisfied)

$$\texttt{InClust}(\texttt{m}, \texttt{c}) \Rightarrow (\texttt{Type}(\texttt{m}, \texttt{e}) \Leftrightarrow \texttt{Type}(\texttt{c}, \texttt{e}))$$

[3]We used the following cues: Mr., Ms., Jr., Inc., Corp., corporation, company. The proportions of known properties range from 14% to 26%.

There are similar hard rules for number and gender.

Different pronouns prefer different entity types, as represented by

$$\texttt{IsPrn}(\texttt{m}) \wedge \texttt{InClust}(\texttt{m}, \texttt{c}) \\ \wedge\texttt{Head}(\texttt{m}, +\texttt{t}) \wedge \texttt{Type}(\texttt{c}, +\texttt{e})$$

which only applies to pronouns, and whose weight is positive if pronoun t is likely to assume entity type e and negative otherwise. There are similar rules for number and gender.

Aside from semantic compatibility, pronouns tend to resolve with nearby mentions. To model this, we impose an exponential prior on the distance (number of mentions) between a pronoun and its antecedent, with weight -1.[4] This is similar to Haghighi and Klein's treatment of salience, but simpler.

4.2 Full MLN

Syntactic relations among mentions often suggest coreference. Incorporating such relations into our MLN is straightforward. We illustrate this with two examples: apposition and predicate nominals. We introduce a predicate for apposition, $\texttt{Appo}(\texttt{x}, \texttt{y})$, where x, y are mentions, and which is true iff y is an appositive of x. We then add the rule

$$\texttt{Appo}(\texttt{x}, \texttt{y}) \Rightarrow (\texttt{InClust}(\texttt{x}, \texttt{c}) \Leftrightarrow \texttt{InClust}(\texttt{y}, \texttt{c}))$$

which ensures that x, y are in the same cluster if y is an appositive of x. Similarly, we introduce a predicate for predicate nominals, $\texttt{PredNom}(\texttt{x}, \texttt{y})$, and the corresponding rule.[5] The weights of both rules can be learned from data with a positive prior mean. For simplicity, in this paper we treat them as hard constraints.

4.3 Rule-Based MLN

We also consider a rule-based system that clusters non-pronouns by their heads, and attaches a pronoun to the cluster which has no known conflicting

[4]For simplicity, if a pronoun has no antecedent, we define the distance to be ∞. So a pronoun must have an antecedent in our model, unless it is the first mention in the document or it can not resolve with previous mentions without violating hard constraints. It is straightforward to soften this with a finite penalty.

[5]We detected apposition and predicate nominatives using simple heuristics based on parses, e.g., if (NP, comma, NP) are the first three children of an NP, then any two of the three noun phrases are apposition.

type, number, or gender, and contains the closest antecedent for the pronoun. This system can be encoded in an MLN with just four rules. Three of them are the ones for enforcing agreement in type, number, and gender between a cluster and its members, as defined in the base MLN. The fourth rule is

$$\begin{aligned}&\neg\texttt{IsPrn}(\texttt{m1}) \wedge \neg\texttt{IsPrn}(\texttt{m2})\\ \wedge&\texttt{Head}(\texttt{m1},\texttt{h1}) \wedge \texttt{Head}(\texttt{m2},\texttt{h2})\\ \wedge&\texttt{InClust}(\texttt{m1},\texttt{c1}) \wedge \texttt{InClust}(\texttt{m2},\texttt{c2})\\ \Rightarrow&\ (\texttt{c1}=\texttt{c2} \Leftrightarrow \texttt{h1}=\texttt{h2}).\end{aligned}$$

With a large but not infinite weight (e.g., 100), this rule has the effect of clustering non-pronouns by their heads, except when it violates the hard rules. The MLN can also include the apposition and predicate-nominal rules. As in the base MLN, we impose the same exponential prior on the number of non-empty clusters and that on the distance between a pronoun and its antecedent. This simple MLN is remarkably competitive, as we will see in the experiment section.

5 Learning and Inference

Unsupervised learning in Markov logic maximizes the conditional log-likelihood

$$\begin{aligned}L(x,y) &= \log P(Y=y|X=x)\\ &= \log \textstyle\sum_z P(Y=y,Z=z|X=x)\end{aligned}$$

where Z are unknown predicates. In our coreference resolution MLN, Y includes `Head` and known groundings of `Type`, `Number` and `Gender`, Z includes `InClust` and unknown groundings of `Type`, `Number`, `Gender`, and X includes `IsPrn`, `Appo` and `PredNom`. (For simplicity, from now on we drop X from the formula.) With Z, the optimization problem is no longer convex. However, we can still find a local optimum using gradient descent, with the gradient being

$$\frac{\partial}{\partial w_i}L(y) = E_{Z|y}[n_i] - E_{Y,Z}[n_i]$$

where n_i is the number of true groundings of the ith clause. We extended PSCG for unsupervised learning. The gradient is the difference of two expectations, each of which can be approximated using samples generated by MC-SAT. The (i,j)th entry of the Hessian is now

$$\frac{\partial^2}{\partial w_i \partial w_j}L(y) = Cov_{Z|y}[n_i,n_j] - Cov_{Y,Z}[n_i,n_j]$$

and the step size can be computed accordingly. Since our problem is no longer convex, the negative diagonal Hessian may contain zero or negative entries, so we first took the absolute values of the diagonal and added 1, then used the inverse as the preconditioner. We also adjusted λ more conservatively than Lowd & Domingos (2007).

Notice that when the objects form independent subsets (in our cases, mentions in each document), we can process them in parallel and then gather sufficient statistics for learning. We developed an efficient parallelized implementation of our unsupervised learning algorithm using the message-passing interface (MPI). Learning in MUC-6 took only one hour, and in ACE-2004 two and a half.

To reduce burn-in time, we initialized MC-SAT with the state returned by MaxWalkSAT (Kautz *et al.*, 1997), rather than a random solution to the hard clauses. In the existing implementation in Alchemy (Kok *et al.*, 2007), SampleSAT flips only one atom in each step, which is inefficient for predicates with unique-value constraints (e.g., $\texttt{Head}(\texttt{m},\texttt{c}!)$). Such predicates can be viewed as multi-valued predicates (e.g., $\texttt{Head}(\texttt{m})$ with value ranging over all `c`'s) and are prevalent in NLP applications. We adapted SampleSAT to flip two or more atoms in each step so that the unique-value constraints are automatically satisfied. By default, MC-SAT treats each ground clause as a separate factor while determining the slice. This can be very inefficient for highly correlated clauses. For example, given a non-pronoun mention `m` currently in cluster `c` and with head `t`, among the mixture prior rules involving `m` $\texttt{InClust}(\texttt{m},\texttt{c})$ is the only one that is satisfied, and among those head-prediction rules involving `m`, $\neg\texttt{IsPrn}(\texttt{m}) \wedge \texttt{InClust}(\texttt{m},\texttt{c}) \wedge \texttt{Head}(\texttt{m},\texttt{t})$ is the only one that is satisfied; the factors for these rules multiply to $\phi = \exp(w_{\texttt{m,c}} + w_{\texttt{m,c,t}})$, where $w_{\texttt{m,c}}$ is the weight for $\texttt{InClust}(\texttt{m},\texttt{c})$, and $w_{\texttt{m,c,t}}$ is the weight for $\neg\texttt{IsPrn}(\texttt{m}) \wedge \texttt{InClust}(\texttt{m},\texttt{c}) \wedge \texttt{Head}(\texttt{m},\texttt{t})$, since an unsatisfied rule contributes a factor of $e^0 = 1$. We extended MC-SAT to treat each set of mutually exclusive and exhaustive rules as a single factor. E.g., for the above `m`, MC-SAT now samples u uniformly

from $(0, \phi)$, and requires that in the next state ϕ' be no less than u. Equivalently, the new cluster and head for $\mathtt{m}$ should satisfy $w_{\mathtt{m},\mathtt{c}'} + w_{\mathtt{m},\mathtt{c}',\mathtt{t}'} \geq \log(u)$. We extended SampleSAT so that when it considers flipping any variable involved in such constraints (e.g., $\mathtt{c}$ or $\mathtt{t}$ above), it ensures that their new values still satisfy these constraints.

The final clustering is found using the MaxWalkSAT weighted satisfiability solver (Kautz *et al.*, 1997), with the appropriate extensions. We first ran a MaxWalkSAT pass with only finite-weight formulas, then ran another pass with all formulas. We found that this significantly improved the quality of the results that MaxWalkSAT returned.

6 Experiments

6.1 System

We implemented our method as an extension to the Alchemy system (Kok *et al.*, 2007). Since our learning uses sampling, all results are the average of five runs using different random seeds. Our optimization problem is not convex, so initialization is important. The core of our model (head mixture) tends to cluster non-pronouns with the same head. Therefore, we initialized by setting all weights to zero, and running the same learning algorithm on the base MLN, while assuming that in the ground truth, non-pronouns are clustered by their heads. (Effectively, the corresponding `InClust` atoms are assigned to appropriate values and are included in Y rather than Z during learning.) We used 30 iterations of PSCG for learning. (In preliminary experiments, additional iterations had little effect on coreference accuracy.) We generated 100 samples using MC-SAT for each expectation approximation.[6]

6.2 Methodology

We conducted experiments on MUC-6, ACE-2004, and ACE Phrase-2 (ACE-2). We evaluated our systems using two commonly-used scoring programs: MUC (Vilain *et al.*, 1995) and B^3 (Amit & Baldwin, 1998). To gain more insight, we also report pairwise resolution scores and mean absolute error in the number of clusters.

[6]Each sample actually contains a large number of groundings, so 100 samples yield sufficiently accurate statistics for learning.

The MUC-6 dataset consists of 30 documents for testing and 221 for training. To evaluate the contribution of the major components in our model, we conducted five experiments, each differing from the previous one in a single aspect. We emphasize that our approach is unsupervised, and thus the data only contains raw text plus true mention boundaries.

MLN-1 In this experiment, the base MLN was used, and the head was chosen crudely as the rightmost token in a mention. Our system was run on each test document separately, using a minimum of training data (the document itself).

MLN-30 Our system was trained on all 30 test documents together. This tests how much can be gained by pooling information.

MLN-H The heads were determined using the head rules in the Stanford parser (Klein & Manning, 2003), plus simple heuristics to handle suffixes such as "Corp." and "Inc."

MLN-HA The apposition rule was added.

MLN-HAN The predicate-nominal rule was added. This is our full model.

We also compared with two rule-based MLNs: **RULE** chose the head crudely as the rightmost token in a mention, and did not include the apposition rule and predicate-nominal rule; **RULE-HAN** chose the head using the head rules in the Stanford parser, and included the apposition rule and predicate-nominal rule.

Past results on ACE were obtained on different releases of the datasets, e.g., Haghighi and Klein (2007) used the ACE-2004 training corpus, Ng (2005) and Denis and Baldridge (2007) used ACE Phrase-2, and Culotta *et al.* (2007) used the ACE-2004 formal test set. In this paper, we used the ACE-2004 training corpus and ACE Phrase-2 (ACE-2) to enable direct comparisons with Haghighi & Klein (2007), Ng (2005), and Denis and Baldridge (2007). Due to license restrictions, we were not able to obtain the ACE-2004 formal test set and so cannot compare directly to Culotta *et al.* (2007). The English version of the ACE-2004 training corpus contains two sections, BNEWS and NWIRE, with 220 and 128 documents, respectively. ACE-2 contains a

Table 1: Comparison of coreference results in MUC scores on the MUC-6 dataset.

	# Doc.	Prec.	Rec.	F1
H&K	60	80.8	52.8	63.9
H&K	381	80.4	62.4	70.3
M&W	221	-	-	73.4
RULE	-	76.0	65.9	70.5
RULE-HAN	-	81.3	72.7	76.7
MLN-1	1	76.5	66.4	71.1
MLN-30	30	77.5	67.3	72.0
MLN-H	30	81.8	70.1	75.5
MLN-HA	30	82.7	75.1	78.7
MLN-HAN	30	83.0	75.8	79.2

Table 2: Comparison of coreference results in MUC scores on the ACE-2004 (English) datasets.

EN-BNEWS	Prec.	Rec.	F1
H&K	63.2	61.3	62.3
MLN-HAN	66.8	67.8	67.3
EN-NWIRE	Prec.	Rec.	F1
H&K	66.7	62.3	64.2
MLN-HAN	71.3	70.5	70.9

training set and a test set. In our experiments, we only used the test set, which contains three sections, BNEWS, NWIRE, and NPAPER, with 51, 29, and 17 documents, respectively.

6.3 Results

Table 1 compares our system with previous approaches on the MUC-6 dataset, in MUC scores. Our approach greatly outperformed Haghighi & Klein (2007), the state-of-the-art unsupervised system. Our system, trained on individual documents, achieved an F1 score more than 7% higher than theirs trained on 60 documents, and still outperformed it trained on 381 documents. Training on the 30 test documents together resulted in a significant gain. (We also ran experiments using more documents, and the results were similar.) Better head identification (MLN-H) led to a large improvement in accuracy, which is expected since for mentions with a right modifier, the rightmost tokens confuse rather than help coreference (e.g., "the chairman of Microsoft"). Notice that with this improvement our system already outperforms a state-of-the-art supervised system (McCallum & Wellner, 2005). Leveraging apposition resulted in another large improvement, and predicate nominals also helped. Our full model scores about 9% higher than Haghighi & Klein (2007), and about 6% higher than McCallum & Wellner (2005). To our knowledge, this is the best coreference accuracy reported on MUC-6 to date.[7] The B^3 scores of MLN-HAN on the MUC-6 dataset are 77.4 (precision), 67.6 (recall) and 72.2 (F1). (The other systems did not report B^3.) Interestingly, the rule-based MLN (**RULE**) sufficed to outperform Haghighi & Klein (2007), and by using better heads and the apposition and predicate-nominal rules (**RULE-HAN**), it outperformed McCallum & Wellner (2005), the supervised system. The MLNs with learning (**MLN-30** and **MLN-HAN**), on the

Table 3: Comparison of coreference results in MUC scores on the ACE-2 datasets.

BNEWS	Prec.	Rec.	F1
Ng	67.9	62.2	64.9
D&B	78.0	62.1	69.2
MLN-HAN	68.3	66.6	67.4
NWIRE	Prec.	Rec.	F1
Ng	60.3	50.1	54.7
D&B	75.8	60.8	67.5
MLN-HAN	67.7	67.3	67.4
NPAPER	Prec.	Rec.	F1
Ng	71.4	67.4	69.3
D&B	77.6	68.0	72.5
MLN-HAN	69.2	71.7	70.4

Table 4: Comparison of coreference results in B^3 scores on the ACE-2 datasets.

BNEWS	Prec.	Rec.	F1
Ng	77.1	57.0	65.6
MLN-HAN	70.3	65.3	67.7
NWIRE	Prec.	Rec.	F1
Ng	75.4	59.3	66.4
MLN-HAN	74.7	68.8	71.6
NPAPER	Prec.	Rec.	F1
Ng	75.4	59.3	66.4
MLN-HAN	70.0	66.5	68.2

[7] As pointed out by Haghighi & Klein (2007), Luo *et al.* (2004) obtained a very high accuracy on MUC-6, but their system used gold NER features and is not directly comparable.

Table 5: Our coreference results in precision, recall, and F1 for pairwise resolution.

Pairwise	Prec.	Rec.	F1
MUC-6	63.0	57.0	59.9
EN-BNEWS	51.2	36.4	42.5
EN-NWIRE	62.6	38.9	48.0
BNEWS	44.6	32.3	37.5
NWIRE	59.7	42.1	49.4
NPAPER	64.3	43.6	52.0

Table 6: Average gold number of clusters per document vs. the mean absolute error of our system.

# Clusters	MUC-6	EN-BN	EN-NW
Gold	15.4	22.3	37.2
Mean Error	4.7	3.0	4.8
# Clusters	BNEWS	NWIRE	NPAPER
Gold	20.4	39.2	55.2
Mean Error	2.5	5.6	6.6

other hand, substantially outperformed the corresponding rule-based ones.

Table 2 compares our system to Haghighi & Klein (2007) on the ACE-2004 training set in MUC scores. Again, our system outperformed theirs by a large margin. The B^3 scores of MLN-HAN on the ACE-2004 dataset are 71.6 (precision), 68.4 (recall) and 70.0 (F1) for BNEWS, and 75.7 (precision), 69.2 (recall) and 72.3 (F1) for NWIRE. (Haghighi & Klein (2007) did not report B^3.) Due to license restrictions, we could not compare directly to Culotta *et al.* (2007), who reported overall B^3-F1 of 79.3 on the formal test set.

Tables 3 and 4 compare our system to two recent supervised systems, Ng (2005) and Denis & Baldridge (2007). Our approach significantly outperformed Ng (2005). It tied with Denis & Baldridge (2007) on NWIRE, and was somewhat less accurate on BNEWS and NPAPER.

Luo *et al.* (2004) pointed out that one can obtain a very high MUC score simply by lumping all mentions together. B^3 suffers less from this problem but is not perfect. Thus we also report pairwise resolution scores (Table 5), the gold number of clusters, and our mean absolute error in the number of clusters (Table 6). Systems that simply merge all mentions will have exceedingly low pairwise precision (far below 50%), and very large errors in the number of clusters. Our system has fairly good pairwise precisions and small mean error in the number of clusters, which verifies that our results are sound.

6.4 Error Analysis

Many of our system's remaining errors involve nominals. Additional features should be considered to distinguish mentions that have the same head but are different entities. For pronouns, many remaining errors can be corrected using linguistic knowledge like binding theory and salience hierarchy. Our heuristics for identifying appositives and predicate nominals also make many errors, which often can be fixed with additional name entity recognition capabilities (e.g., given "Mike Sullivan, VOA News", it helps to know that the former is a person and the latter an organization). The most challenging case involves phrases with different heads that are both proper nouns (e.g., "Mr. Bush" and "the White House"). Handling these cases requires domain knowledge and/or more powerful joint inference.

7 Conclusion

This paper introduces the first unsupervised coreference resolution system that is as accurate as supervised systems. It performs joint inference among mentions, using relations like apposition and predicate nominals. It uses Markov logic as a representation language, which allows it to be easily extended to incorporate additional linguistic and world knowledge. Future directions include incorporating additional knowledge, conducting joint entity detection and coreference resolution, and combining coreference resolution with other NLP tasks.

8 Acknowledgements

We thank the anonymous reviewers for their comments. This research was funded by DARPA contracts NBCH-D030010/02-000225, FA8750-07-D-0185, and HR0011-07-C-0060, DARPA grant FA8750-05-2-0283, NSF grant IIS-0534881, and ONR grant N-00014-05-1-0313 and N00014-08-1-0670. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, NSF, ONR, or the United States Government.

References

Amit, B. & Baldwin, B. 1998. Algorithms for scoring coreference chains. In *Proc. MUC-7*.

Bakir, G.; Hofmann, T.; Schölkopf, B.; Smola, A.; Taskar, B. and Vishwanathan, S. (eds.) 2007. *Predicting Structured Data*. MIT Press.

Culotta, A.; Wick, M.; Hall, R. and McCallum, A. 2007. First-order probabilistic models for coreference resolution. In *Proc. NAACL-07*.

Denis, P. & Baldridge, J. 2007. Joint determination of anaphoricity and coreference resolution using integer programming. In *Proc. NAACL-07*.

Getoor, L. & Taskar, B. (eds.) 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Haghighi, A. & Klein, D. 2007. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proc. ACL-07*.

Kautz, H.; Selman, B.; and Jiang, Y. 1997. A general stochastic approach to solving problems with hard and soft constraints. In *The Satisfiability Problem: Theory and Applications*. AMS.

Klein, D. & Manning, C. 2003. Accurate unlexicalized parsing. In *Proc. ACL-03*.

Kok, S.; Singla, P.; Richardson, M.; Domingos, P.; Sumner, M.; Poon, H. & Lowd, D. 2007. The Alchemy system for statistical relational AI. http://alchemy.cs.washington.edu/.

Lowd, D. & Domingos, D. 2007. Efficient weight learning for Markov logic networks. In *Proc. PKDD-07*.

Luo, X.; Ittycheriah, A.; Jing, H.; Kambhatla, N. and Roukos, S. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. ACL-04*.

McCallum, A. & Wellner, B. 2005. Conditional models of identity uncertainty with application to noun coreference. In *Proc. NIPS-04*.

Ng, V. 2005. Machine Learning for Coreference Resolution: From Local Classification to Global Ranking. In *Proc. ACL-05*.

Poon, H. & Domingos, P. 2006. Sound and efficient inference with probabilistic and deterministic dependencies. In *Proc. AAAI-06*.

Poon, H. & Domingos, P. 2007. Joint inference in information extraction. In *Proc. AAAI-07*.

Poon, H. & Domingos, P. 2008. A general method for reducing the complexity of relational inference and its application to MCMC. In *Proc. AAAI-08*.

Richardson, M. & Domingos, P. 2006. Markov logic networks. *Machine Learning* 62:107–136.

Vilain, M.; Burger, J.; Aberdeen, J.; Connolly, D. & Hirschman, L. 1995. A model-theoretic coreference scoring scheme. In *Proc. MUC-6*.

Wei, W.; Erenrich, J. and Selman, B. 2004. Towards efficient sampling: Exploiting random walk strategies. In *Proc. AAAI-04*.