

**Advances in Learning and Inference for Partition-wise
Models of Coreference Resolution**

Michael Wick and Andrew McCallum

CMPSCI Technical Report
UM-CS-2009-028

June 2, 2009

Department of Computer Science
University of Massachusetts
140 Governors Drive
Amherst, Massachusetts 01003

Abstract

Noun phrase coreference resolution is a difficult task that has driven research in both natural language processing and machine learning. It has been a subject of feature engineering as well as model development, including partition-wise conditional random fields with Markov-chain Monte Carlo inference. In this paper we combine the latest feature engineering with an exploration of machine learning advances in three areas: the proposal distribution for inference, the ground-truth evaluation signal for the objective function, and the online update rule for parameter estimation. In particular, we investigate learned, adaptive proposal distributions; we evaluate various methods of ranking possible worlds; and we adapt a recently proposed confidence-weighted classification method to structured prediction with SampleRank. We achieve new best results on ACE 2004, surpassing two previous state-of-the-art systems with 10% and 3% error reductions respectively.

1 Introduction

Noun phrase coreference resolution—the process of clustering noun-phrases into anaphoric sets—is a notoriously difficult task. In particular, the problem has not only inspired new machine learning (ML) techniques (McCallum and Wellner, 2003; Daumé III and Marcu, 2005; Daumé III et al., 2006; Culotta et al., 2007), but has also driven researchers to develop highly sophisticated features (Bengston and Roth, 2008), incorporate other prediction tasks (Poon and Domingos, 2007), as well as leverage topic models and encyclopedic knowledge (Poesio et al., 2007).

Recently, machine learning (ML) has become the dominant approach to coreference because it enables weight-learning for the myriad of features required for the task. Initial ML efforts learned to make binary pairwise decisions about whether a mention m_1 refers to a mention m_2 , but these approaches ignore the transitivity constraint (Soon et al., 2001). Later, joint models treated the problem as a structured prediction task, simultaneously predicting coreferent pairs, allowing them to take transitivity into consideration (McCallum and Wellner, 2003). More recently, models have been developed that leverage features defined by first-order logic over arbitrarily-sized partitions, enabling reasoning at the set-level rather than the pairwise level (Pasula et al., 2003; Milch et al., 2004; Culotta et al., 2007).

These sophisticated models allow increased expressiveness, but at the expense of exact inference. Therefore, approximate inference and parameter estimation techniques are crucial to obtaining high coreference performance and present a promising avenue of research. Approximations for parameter estimation have often involved piecewise variants of maximum likelihood (McCallum and Sutton,

2004; McCallum and Wellner, 2003; Denis and Baldridge, 2007a). That is, factors are learned independently, either as classifiers or rankers, and then stitched together at inference time to produce consistent transitivity-obeying coreference configurations.

In our work we apply a recently developed structured learning algorithm *SampleRank* (Culotta, 2008) that can *globally* train a coreference model. The premise behind SampleRank is that we can exploit the neighborhood structure of stochastic local search and make *global* gradient updates based solely on the *local* modifications. There are two key ingredients to SampleRank: (a) the *proposal distribution*, which defines the neighborhood over which updates are possible, and (b) the *ground truth signal*, which in the context of coreference is a global measurement of the quality of the configuration, (for example, B-cubed F1 or accuracy). In contrast, many other approaches to coreference traditionally learn a 0-1 or likelihood-based binary classification (Soon et al., 2001; McCallum and Wellner, 2003) or rank loss, (Denis and Baldridge, 2007b). We note, however, that research in other NLP tasks has shown that optimizing risk (according to some ground-truth-based evaluation signal) results in better performance—for example BLEU score in machine translation (Smith and Eisner, 2006).

In this paper we propose several advances in modeling and machine learning, and explore their impact through a series of ablations studies. First, we present a new method for learning a dynamic proposal distribution in Metropolis-Hastings—a particularly flexible form of Markov-chain Monte Carlo inference. Next, we adapt the confidence-weighted classification update method of Dredze et al. (2008) to satisfy constraints in the SampleRank objective. Finally, we explore alternative cost structures by investigating three ground-truth training metrics

Our contributions enable us to achieve new state-of-the-art results on the ACE 2004 dataset, reducing B-cubed error by 10% over Culotta et al. (2007) and 3% over Bengston and Roth (2008). More specifically, we find that our adaptive proposer yields superior results in comparison to a conventional split-merge or Gibbs-style jump function. In addition we find that confidence-weighted parameter updates outperform the earlier passive aggressive (MIRA) method in our structured prediction setting. Finally, we compare the results of different ground truth training signals based on F1, accuracy, and mutual information.

2 Model

In this section we overview a discriminative factor-graph model that enables features to be expressed with first-order logic over sets of arbitrary size. Let x be the set of all the mentions in document D and let $x^i \in \mathcal{P}(x)$ be a cluster of mentions (in our notation superscript indexes a set of objects while subscripts refers to

a single object). Let y_i be a hidden binary variable that is true if and only if all the mentions in x^i are coreferent. Furthermore, let $\psi : y_i \times x^i \mapsto \mathfrak{R}$ be a factor that maps clusters of mentions and their corresponding binary coreference label to a real value. Intuitively, factors are a measurement of the cluster’s coreferential compatibility. Each factor is a parameterized log-linear function of the form:

$$\psi(x^i, y_i) = \exp \left(\sum_k \theta_k \phi_k(x^i, y_i) \right)$$

where θ_k is a real-valued parameter corresponding to a real-valued feature function $\phi_k(x^i, y_i)$. For concrete examples of feature functions please see Section 5.1.

Furthermore, let $x^j \in \mathcal{P}(x)$ be some other cluster. We define a binary coreference variable between clusters, y_{ij} , and a corresponding compatibility factor $\psi(x^i, x^j, y_{ij})$, which decomposes into pairwise computations as follows:

$$\psi(x^i, x^j, y_i) = \prod_{x_i \in x^i} \prod_{x_j \in x^j} \exp \left(\sum_k \theta_k \phi_k(x_i, x_j, y_i) \right)$$

intuitively, the score of factors between clusters should be low, indicating that the two clusters are not coreferent.

In summary, we adopt a model that has first-order (setwise) factors over single clusters with the addition of pairwise factors that cross cluster boundaries.

The goal is to model the conditional probability of a coreference configuration y given a set of observed mentions x . Formally, a coreference configuration is described as a setting to each of the binary y variables such that a valid equivalence relation results. Rather than including deterministic factors (constraints) to enforce properties such as transitivity, we adopt a more efficient approach by defining a class of property preserving jump functions (see Section 3).

Since we have described a coreference configuration as a setting to all the hidden variables, we can now define the discriminative coreference model as:

$$P(Y = y | x) = \frac{1}{Z_x} \prod_{y_i \in y} \psi(x^i, y_i) \prod_{y_i, y_j \in y} \psi(x^i, x^j, y_i)$$

where Z_x is an input dependent normalizing constant ensuring that the distribution sums to one over all configurations.

3 Inference

The inference problem we are most concerned with is finding the most probable setting of the hidden coreference variables given the observed mentions. This is an instance of the *Maximum a Posteriori* (MAP) inference problem:

$$\operatorname{argmax}_{y \in \mathcal{F}} P(Y = y|x)$$

where \mathcal{F} is the feasible region defined by the deterministic transitivity-constraints. We can rewrite the above equation in terms of our unnormalized model:

$$\operatorname{argmax}_{y \in \mathcal{F}} \prod_{y_i \in y} \psi(x^i, y_i) \prod_{y_i, y_j \in y} \psi(x^i, x^j, y_i)$$

Because our model contains no latent variables, we do not require computing any expectations; therefore, we are justified in using a non-ergodic annealing variant of the Metropolis-Hastings algorithm as a local search procedure for finding the MAP configuration.

3.1 Metropolis-Hastings

The Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) requires a proposal distribution $q(y'|y)$, from which local modifications are drawn; and a model $p(y)$, which is a distribution over all possible configurations in \mathcal{F} . The distribution $q(y'|y)$ *proposes* a local modification to y which results in a new configuration y' . The new configuration y' is accepted probabilistically by a Bernoulli distribution with success parameter α defined below:

$$\alpha = \min \left(\frac{p(y')}{p(y)} \times \frac{p(y|y')}{p(y'|y)}, 1 \right)$$

where $p(y')/p(y)$ is the ratio of the proposed configuration and the current configuration according to the model, and $p(y|y')/p(y'|y)$ is the backwards/forwards ratio, which intuitively cancels the bias introduced by the proposal distribution. However, because we are performing MAP with no latent variables, we can forgo this term in computing the acceptance probability:

$$\alpha = \min \left(\frac{p(y')}{p(y)}, 1 \right) = \min \left(\frac{p(Y = y'|X, \theta)}{p(Y = y|X, \theta)}, 1 \right)$$

note that in the ratio above both the partition function and all factors not involved in the proposed change cancel, allowing efficient computations. More specifically, only factors neighboring variables that are changed by the proposal distribution need to be computed. In this paper we consider three proposal distributions which modify a constant $O(1)$ number of variables at each step. For our particular model this means only a constant number of factors need to be computed to compute acceptance (proof omitted for space, but can be obtained by plugging model in for acceptance distribution). MAP inference using Metropolis-Hastings is outlined in more detail in Algorithm 1

Algorithm 1 Metropolis-Hastings for MAP

```
1: Input:  
   Data  $x$ ,  
   model  $p$   
2: randomly initialize coreference config.  $y$   
3:  $result \leftarrow y$   
4:  $max \leftarrow -\infty$   
5: for  $i = 1, \dots$ , number of steps do  
6:    $y' \leftarrow q(y)$   
7:   if  $true \sim \alpha(y', y)$  then  
8:      $y \leftarrow y'$   
9:     if  $p(Y = y'|x) > max$  then  
10:       $max \leftarrow p(Y = y'|x)$   
11:       $result \leftarrow y'$   
12:     end if  
13:   end if  
14: return result  
15: end for
```

The Metropolis-Hastings algorithm also allows us to perform efficient inference in the presence of deterministic factors. Specifically, we can initialize the y variables to a configuration that obeys transitivity, and then use a proposal distribution which preserves the transitivity property; therefore, we can avoid hypothesizing gratuitous configurations. Three property-preserving proposal distributions are presented below.

3.2 Proposal Distributions

Perhaps the simplest proposal distribution for clustering is the *single-flip* jump function, which uniformly picks a mention and then moves it to a random cluster. This proposal distribution changes at most two clusters and therefore only a constant number of factors must be computed to evaluate the change. Furthermore, the distribution preserves transitivity trivially because it first removes the mention from its original cluster before placing it in a new one.

The next distribution we consider is a uniform *split-merge* proposer based on one used in (?). The distribution picks two mentions randomly. If the two mentions are in different clusters, then the clusters are merged, otherwise, they are in the same cluster and that cluster is split. This proposer also modifies at most two clusters and so the difference can be computed efficiently. Additionally, splitting and merging are closed under transitivity.

The two above proposal distributions are impoverished in the sense that they rely on blind-luck to stumble across a better configuration. In practice, it may require a large number of samples to before an uphill move is proposed. This directly motivates a new proposal distribution that can adaptively learn to propose moves

that are more likely to be accepted. Our adaptive proposal distribution observes the model’s belief in each proposed move and updates its parameters to correspond to the model. Details are provided in Section 3.3

3.3 Adaptive Proposal Distributions

The idea behind an adaptive proposal distribution is to add a set of parameters that can be learned to encourage the jump function to propose transitions that the model is likely to accept. The parameters in our method are an $n \times n$ compatibility matrix M , (where n is the number of mentions in the document we are clustering). Intuitively, M_{ij} can be interpreted as a measurement of affinity, or more formally as the unnormalized log probability of mention i being in the same cluster as mention j . The adaptive proposal distribution learns by proposing a move, and then observing the ratio $r = P(y'|x, \theta)/P(y|x, \theta)$. If the $r < 0$ then the model dislikes the proposed configuration and the proposer’s weights are updated so that it is less likely to propose that move in the future. The sufficient statistics required by the update are determined by the pairwise affinities involved in the move (described in more detail later).

We now discuss how to use the matrix M to improve the simple split-merge proposal distribution described in the previous section. Rather than picking two mentions randomly, we pick a cluster c_i randomly. Using the affinity matrix, we compute the probability ζ of all mentions in c_i being coreferent. Therefore, with probability ζ , the mentions in c_i are all coreferent and we probabilistic select a cluster c_j to merge it with (this procedure is described below); and with probability $1 - \zeta$ the cluster is not coreferent, so we probabilistic split the cluster (described below).

If c_i is selected for a merge, then we compute a probability distribution d_m over c_i being coreferent with each other cluster in the document. We then draw a cluster $c_j \sim d_m$ to merge with c_i . On the other hand, if c_i is selected for a split, then we use M to compute the pairwise repulsions (1 minus affinity) of each mention in the cluster. Two mentions $m_a \in c_i$ and $m_b \in c_i$ are drawn randomly from this distribution and placed in separate clusters c_a and c_b . Then, for each of the remaining mentions in c_i , we compute a Bernoulli distribution for each belonging to c_a or c_b and make a decision about where to move the mention based on this probability.

The parameter update rule for the adaptive split-merge depends on whether the proposed jump was a merge or a split. In the case of a merge error, clusters c_a and c_b were incorrectly combined. Therefore, the sufficient statistics that should be involved in the update are the mention pairs that cross the cluster boundaries. More specifically, the update is as follows:

$$M_{ij} \leftarrow M_{ij} - \eta \text{ s.t. } m_i \in c_a m_j \in c_b$$

where η is the learning rate.

Similarly, a split error incorrectly splits a cluster c into c_a and c_b , and so the split update increases the compatibility of c_a and c_b . The update is as follows:

$$M_{ij} \leftarrow M_{ij} + \eta \text{ s.t. } m_i \in c_a m_j \in c_b$$

If required, we can compute the backward/forward ratio for the the adaptive split-merge proposal distribution so that detailed balance is obeyed. Since the proposer picks moves in a hierarchical fashion, we are able to exploit independence assumptions to compute the probabilities. Let $p(c)$ be the probability of a cluster being coreferent and $p(c_i, c_j)$ be the probability that two clusters corefer to each other. The forward probability of a merge can be computed as:

$$\frac{1}{|D|} \times \zeta \times p(\text{coref}(c_i, c_j); M)$$

the forward probability of a split is:

$$\frac{1 - \zeta}{|D|} \times p(\text{coref}(c_a); M) \times p(\text{coref}(c_b); M)$$

where make use of the exchangeability property which in this context states that the order in which mentions are added to clusters c_a and c_b do not impact their respective probabilities of being coreferent. The backward probabilities are computed nearly identically.

4 Learning

We learn the parameters of our coreference factor graph with SampleRank (Culotta, 2008). SampleRank observes the series of neighbor configuration pairs produced by the proposal distribution in Metropolis-Hastings and updates the model parameters θ based on a ground-truth evaluation signal (such as F1 or accuracy). Let $n_{y,y'} = \langle y', y \rangle$ be a neighbor configuration pair, $\mathcal{S}(y)$ be a ground truth scoring metric (signal) with $\mathcal{S}(y', y)$ shorthand for $\mathcal{S}(y') - \mathcal{S}(y)$, and $\mathcal{M}(y', y)$ be the unnormalized log probability ratio according to the model. Note that \mathcal{M} can be computed efficiently since it is merely the log of the model ratio in the Metropolis-Hastings algorithm.

At each time-step SampleRank observes a pair $n_{y',y}$ and checks whether $\text{sign}(\mathcal{S}(y', y)) = \text{sign}(\mathcal{M}(y', y))$. There are two possible updates to consider, either $\mathcal{S}(y', y)$ is positive and $\mathcal{M}(y', y)$ is negative (or the other way around). Let

$\phi_{y',y}$ be the sufficient statistics involved in the gradient computation, then the update rule for SampleRank is as follows:

$$\theta = \theta + \begin{cases} -\eta \phi_{y',y} & \text{if } \mathcal{S}(y', y) < 0 \wedge \theta \cdot \phi_{y',y} > 0 \\ \eta \phi_{y',y} & \text{if } \mathcal{S}(y', y) > 0 \wedge \theta \cdot \phi_{y',y} \leq 0 \end{cases}$$

where $\mathcal{M}(y', y) = \phi_{y',y} \cdot \theta$ according to the log-linear model and η is the learning rate. For more details of SampleRank in the context of Metropolis-Hastings see Algorithm 2. The choice of η in this algorithm is particularly important and we discuss three possibilities in the sequel.

Algorithm 2 SampleRank for Coreference

```

1: Input: Data  $x$ 
   Ground truth function  $\mathcal{S}$ 
2: randomly initialize  $y$ 
3:  $\theta \leftarrow \mathbf{0}$ 
4: for  $i = 1, \dots$ , number of steps do
5:    $y' \leftarrow q(y)$ 
6:    $a \sim \alpha(y', y)$ 
7:   if  $\mathcal{S}(y', y) < 0 \wedge \theta \cdot \phi_{y',y} > 0$  then
8:      $\theta \leftarrow \theta - \eta \phi_{y',y}$ 
9:   end if
10:  if  $\mathcal{S}(y', y) > 0 \wedge \theta \cdot \phi_{y',y} \leq 0$  then
11:     $\theta \leftarrow \theta + \eta \phi_{y',y}$ 
12:  end if
13:  if  $a = \text{true}$  then
14:     $y \leftarrow y'$ 
15:  end if
16: end for
17: return  $\theta$ 

```

4.1 Parameter Update Rules

The parameter update rule can play an important role in the generalization and convergence of SampleRank. We explore three possibilities for η , including a recently developed confidence-weighted version. The first and most basic sets η to a constant scalar, resulting in a perceptron update. The second dynamically determines η to correctly satisfy the SampleRank constraint (that the signs match) using the margin infused relaxed algorithm (MIRA) (Crammer et al., 2006). The MIRA update correctly satisfies the constraint by a specified margin γ subject to the additional constraint that the change to the parameter vector is minimal. This involves solving the following quadratic program:

$$\theta^{t+1} = \underset{\theta}{\operatorname{argmin}} \|\theta^t - \theta\|^2 \quad \text{s.t.} \\ \theta \cdot \phi_{y',y} \geq \gamma$$

In practice we apply the l_1 box-constraint regularizer to bound the size of η in case the features are insufficient to satisfy SampleRank’s constraint.

Note that η does not have to be a scalar. Dredze et al. (2008) derived a version of MIRA that updates each feature with a different learning rate proportional to a feature-specific confidence estimate. Each update requires modifying both the model parameters and feature confidence values, which are estimated with a diagonal covariance matrix. The algebraic steps are quite involved but the details can be found in Dredze et al. (2008). This confidence-weighted update method is the third and final method we test in the context of SampleRank.

5 Results

In this section we present three sets of results on the ACE 2004 coreference corpus using the ground-truth mentions. The corpus contains 450 newswire documents from newspapers, newswire, and broadcast news (transcribed talk shows). We use 336 documents for training and 114 for evaluation. Unless otherwise stated, all experiments make ten random passes over the training set and perform 4000 Metropolis-Hastings walk-steps per document (for SampleRank learning). Because there is no agreed upon performance metric, we evaluate our experiments with several different methods: B-cubed F1 (Amit and Baldwin, 1998), MUC F1 (Vilain et al., 1995), pairwise F1, accuracy, and normalized mutual information. We hope that this will make comparisons with previous and future work easier.

We run a set of ablation experiments. In Section 5.2 we evaluate the effect of the proposal distribution. In Section 5.3 we compare gradient update strategies. In Section 5.4 we demonstrate the importance of choosing a proper ground-truth evaluation metric for learning. Finally in 5.5 we compare with recent work.

5.1 Features

Before we present the results, we will describe the features used in our model. First, we describe the data structure used to encode an ACE noun-phrase mention. A mention m consists of the following fields: *text*, the full mention string; *head*, the *extent*, *modifiers*, *sentence number*, *sentence position*, *gender*, *number*, *part of speech*.

We begin by describing a collection of pairwise feature extractions $\chi(m_i, m_j) \mapsto \mathbb{R}$, which take mentions m_i and m_j as arguments and return a real-valued result (many, but not all, are strictly boolean). These extractors include:

- **exact match:** returns 1 iff both m_i and m_j have the same gender, number, entity type, head text or full text, or acronym encoding of the full text (each condition is a different exact match extractor).
- **substring match:** returns 1 iff the full text of mention m_i is a substring of m_j .
- **cosine distance:** returns the token-wise TFIDF-weighted cosine distance of the full text of mention m_i and m_j . The TFIDF weights are computed over the entire ACE corpus.
- **number distance:** depending on the type, returns the distance between m_i and m_j measured by sentences (number of sentence between them), tokens, and mentions (number of mentions between them).
- **nearby sentence:** returns 1 if the mentions are in the same sentence or if they are one sentence apart (these are considered separate extractors).
- **nested mentions:** returns 1 if mention m_i is nested inside m_j (for example U.S. president Bill Clinton nests the mention U.S. inside of U.S. president . . .).
- **previous mention match:** returns 1 if m_i is the closest mention to m_j whose string overlaps with m_j .
- **c-commanding:** returns 1 if m_i is a pronoun in a c-commanding relationship with m_j
- **mention sequence bias:** returns 1 if mention m_i has mention type α and m_j has mention type β .
- **memorization:** this feature is the same as the one described in (Bengston and Roth, 2008)—returns 1 if mention m_i has token α and m_j has token β .
- **parse tree:** this feature is 1 iff mention m_i and m_j are in the same sentence and parse tree node n occurs between them.
- **intervening tokens:** this feature is 1 iff m_i and m_j are in the same sentence and token t occurs between them.
- **entity type:** entity types for pronouns and common nouns are not available, so we include a feature that is 1 iff m_i is a proper noun with a known entity type α and m_j has a head token β . This feature captures an entity type's distribution over head tokens.

The features described above are classical features used throughout the coreference literature, plus some additional features inspired by Bengston and Roth (2008). However, the above features are pairwise, but our model is set-wise. We take full advantage of our setwise model by quantifying (as in first order logic) and aggregating the above features over sets of mentions. For example, we can count the average number of mention pairs in cluster c that contain a head match. Let $\chi(m_i, m_j)$ be a pairwise extractor, then an exhaustive list of the aggregates used in our model include:

- **existential** an existential quantifier that returns 1 iff $\exists m_i, m_j \in c \chi(m_i, m_j) = 1$

- **universal** a universal quantifier that returns 1 iff $\forall m_i, m_j \in c \chi(m_i, m_j) = 1$
- **average** computes the mean value $\sum_{i,j \in c} \chi(m_i, m_j) / n$ of relevant pairs in the cluster, where n is the number of relevant pairs
- **min** computes minimum value of the extractor in the cluster $\min_{i,j \in c} \chi(m_i, m_j)$
- **max** computes maximum value of the extractor in the cluster $\max_{i,j \in c} \chi(m_i, m_j)$
- **bin(s)** bins the real-valued aggregates (min, max, avg) into binary features. We use $s \in \{2, 4, 20, \text{and } 100\}$ where s is the number of bins (determines granularity).

Finally, we also include pairwise feature filters that determine whether or not a pairwise extractor is relevant for a pair of mentions m_i, m_j . For example, we do not care about string matches between a pronoun and a proper noun, so we filter out pairs involving pronouns. This means, for example, that the universal quantifier will return true if all *relevant* extraction pairs of mentions are true. We use a *same sentence* filter for the intervening tokens and parse nodes features. We also use a *proper noun only* feature for head match, text match, and entity-type match (since we do not know entity types of pronouns and common nouns).

5.2 Comparing Proposal Distributions

In this section we evaluate the performance impact of the three proposal distributions described previously:

- **adaptive split-merge**: the modified split-merge proposal distribution described in Section 3.2 that updates its parameters to be in accordance with the full model.
- **uniform split-merge**: this is the simple split-merge distribution that uniformly splits or merges clusters
- **single-flip** a Gibbs-style proposal distribution that randomly moves a mention from one cluster and puts it in another.

We use accuracy as the ground truth loss signal and confidence-weighting as the update rule. The same proposal distribution is used for both training and testing. The results are displayed in Table 2. While simple split-merge performs worse than single-flip, the adaptive split-merge outperforms both proposal distributions across all evaluation metrics. In other words, the learning mechanism transforms a sub-standard proposal distribution a superb one.

	Proposer	Prec	Recall	F1
BCubed	adaptive	87.9	76.0	81.5
	split/merge	89.7	71.5	79.6
	single-flip	87.5	73.7	80.1
Pair F1	adaptive	60.5	44.1	51.0
	split/merge	60.9	31.6	41.6
	single-flip	52.0	39.3	44.7
MUC	adaptive	78.1	63.7	70.1
	split/merge	76.2	55.1	63.9
	single-flip	76.7	58.6	66.5
MutInf	adaptive			90.4
	split/merge	—	—	89.9
	single-flip			90.0
Acc	adaptive			96.2
	split/merge	—	—	96.0
	single-flip			95.6

Table 1: A comparison of various proposal distributions

5.3 Comparing Update Strategies

In this section we compare the following three update rules:

- **CW**: the confidence-weighted update rule (Dredze et al., 2008) adapted for SampleRank
- **MIRA**: the passive aggressive margin-based update rule (Crammer et al., 2006)
- **perceptron**: standard perceptron style update

The results are displayed in Table 2. We can see that the confidence-weighted update scheme reduces error across all evaluation metrics.

5.4 Ground Truth Signal Comparisons

In this section we compare various choices for the ground truth signal. The results reported here use the adaptive proposal distribution and SampleRank with confidence weighted updates. We compare three signals: accuracy, BCubed F1, and normalized mutual information. Accuracy performs the best across nearly all evaluation metrics, including BCubed F1. This may not be surprising since F1 is the harmonic mean of precision and recall, and is likely riddled with local optima. SampleRank may be learning a bumpy optimization surface making MAP inference more difficult. The results are displayed in Table 3.

	Update	Prec	Recall	F1
BCubed	CW	87.9	76.0	81.5
	MIRA	89.7	73.0	80.5
	perceptron	81.3	74.2	77.6
Pair F1	CW	60.5	44.1	51.0
	MIRA	60.7	36.4	45.5
	perceptron	52.9	38.0	44.3
MUC	CW	78.1	63.7	70.1
	MIRA	78.8	58.9	67.4
	perceptron	68.5	63.0	65.6
MutInf	CW			90.4
	MIRA	—	—	90.3
	perceptron			89.0
Acc	CW			96.2
	MIRA	—	—	96.1
	perceptron			95.7

Table 2: A comparison of three update rules: confidence weighted (CW), passive aggressive (MIRA), and conventional perceptron.

5.5 Comparison with Recent Work

In this section we compare with recent supervised methods for coreference. We reduce error by 10% over the system by Culotta et al. (2007), which trains the model in a piecewise fashion using rank-based passive aggressive updates. Additionally, we are competitive with a system by Bengston and Roth (2008)—which investigates and applies sophisticated features—achieving a 3% error reduction over their system. The comparison (using B-cubed) is presented in Table 4.

6 Related Work

We have provided some comparisons with related work in Section 1, but provide additional comparisons here.

There have been other studies evaluating the impact of modeling on coreference resolution. Culotta et al. (2007) develop a first-order probabilistic model that enables high-order features over sets of mentions and show that this improves over a pairwise baseline. They propose an online learning algorithm using MIRA in the context of greedy structured prediction to approximately train their model. In contrast, we globally train the parameters of the model with SampleRank and explore confidence-weighted alternatives to MIRA. Additionally, our inference pro-

	Signal	Prec	Recall	F1
BCubed	Acc	87.9	76.0	81.5
	BCu	81.8	80.0	80.9
	MI	80.3	79.9	80.1
Pair F1	Acc	60.5	44.1	51.0
	BCu	45.6	57.5	50.9
	MI	41.0	57.7	47.9
MUC	Acc	78.1	63.7	70.1
	BCu	74.5	69.1	71.7
	MI	72.9	69.1	70.9
MutInf	Acc			90.4
	BCu	—	—	90.0
	MI			89.6
Acc	Acc			96.2
	BCu	—	—	95.0
	MI			94.3

Table 3: A comparison of different ground truth signals accuracy (Acc), B-Cubed (BCu), and normalized mutual information (MI)

Work	Prec	Recall	B^3 F1
this work	87.9	76.0	81.5
BR	88.3	74.5	80.8
CWHM	86.7	73.2	79.3

Table 4: A comparison with recent state of the art systems: Bengston and Roth (BR), and Culotta et al. (CWHM).

cedure is stochastic rather than purely greedy.

More recently, Bengston and Roth (2008) explore the impact of sophisticated features in the context of a relatively simple pairwise model. They find that using *predicted* features dramatically improves performance and argue that the impact of feature development would likely dampen the need for more sophisticated coreference models. However, these models enable the expression of additional features that are not possible in simple pairwise models. Furthermore, we believe that these models can drive research in approximate learning and inference. The purpose of this work is to explore the impact of machine learning techniques; we conclude that learning is still an important avenue for improving coreference systems even in the presence of baseline based on sophisticated features.

7 Conclusions and Future Work

We have demonstrated the importance of machine learning algorithms in the context of large coreference models. In particular, we have evaluated the impact of update-rules, proposal distributions, and structured evaluation metrics in the context of SampleRank, a new training method for large factor graphs. We also demonstrated how research in machine learning has an impact on coreference by outperforming a previous state of the art system.

There are many possible directions for future work, including a more detailed investigation of adaptive proposal distributions, and other parameter update schemes using alternative measures of ‘confidence’. Future work should also investigate SampleRank’s ability to learn the cost-structure of more complicated evaluation metrics, as well as explore additional training methods that can exploit this structure. For example, training SampleRank with accuracy lead to better B-cubed scores than when SampleRank was trained with B-cubed. It is likely that the optimization surface learned from B-cubed is riddled with local optima indicating that delayed-reward ideas from reinforcement-learning could be promising direction for future work.

References

- B. Amit and B. Baldwin. 1998. Algorithms for scoring coreference chains. In *Proceedings of the Seventh Message Understanding Conference (MUC7)*.
- Eric Bengston and Dan Roth. 2008. Understanding the value of features for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.
- Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.
- Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *HLT/NAACL*. to appear.
- Aron Culotta. 2008. *Learning and inference in weighted logic with application to natural language processing*. Ph.D. thesis, University of Massachusetts, May.
- Hal Daumé III and Daniel Marcu. 2005. Learning as search optimization: Approximate large margin methods for structured prediction. In *International Conference on Machine Learning (ICML)*.
- Hal Daumé III, John Langford, and Daniel Marcu. 2006. Search-based structured prediction. Technical Note.
- Pascal Denis and Jason Baldridge. 2007a. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies*, pages 236–243, Rochester, New York, April. Association for Computational Linguistics.

- Pascal Denis and Jason Baldrige. 2007b. A ranking approach to pronoun resolution. In *IJCAI*.
- Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 264–271, New York, NY, USA. AC-M.
- W.K. Hastings. 1970. Monte carlo sampling methods using markov chains and their applications. In *Biometrika*.
- Andrew McCallum and Charles Sutton. 2004. Piecewise training with parameter independence diagrams: Comparing globally- and locally-trained linear-chain crfs. In *NIPS 2004 Workshop on Learning with Structured Outputs*.
- A. McCallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *IJCAI Workshop on Information Integration on the Web*.
- N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. 1953. Equations of state calculations by fast computing machines. In *Journal of Chemical Physics*.
- Brian Milch, Bhaskara Marthi, and Stuart Russell. 2004. Blog: Relational modeling with unknown objects. In *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields*.
- Hanna Pasula, Bhaskara Marthi, Brian Milch, Stuart Russell, and Ilya Shpitser. 2003. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems 15*. MIT Press.
- Massimo Poesio, David Day, Ron Arstein, Jason Duncan, Vladimir Eidelman, Claudio Giuliano, Rob Hall, Janet Hitzeman, Alan Jern, Mijail Kabadjov, Gideon Mann, Paul McNamee, Alessandro Moschitti, Simone Ponzetto, Jason Smith, Josef Steinberger, Michael Strubte, Jian Su, Yannick Versley, Xiaofeng Yang, and Michael Wick. 2007. Exploiting encyclopedic and lexical resources for entity disambiguation. Technical report, Johns Hopkins University.
- Hoifung Poon and Pedro Domingos. 2007. Joint inference in information extraction. In *AAAI*, pages 913–918, Vancouver, Canada. AAAI Press.
- David A. Smith and Jason Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 787–794, Morristown, NJ, USA. Association for Computational Linguistics.
- Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.*, 27(4):521–544.
- M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of MUC6*, pages 45–52.