Coreference Resolution for Information Extraction

Dmitry Zelenko and **Chinatsu Aone** and **Jason Tibbetts**
SRA International
4300 Fair Lakes Ct.
Fairfax, VA 22033
{dmitry_zelenko,chinatsu_aone,jason_tibbetts}@sra.com

Abstract

We compare several approaches to coreference resolution in the context of information extraction. We present a loss-based decoding framework for coreference resolution and a greedy algorithm for approximate coreference decoding, in conjunction with Perceptron and logistic regression learning algorithms. We experimentally evaluate the presented approaches using the Automatic Content Extraction evaluation methodology, with promising results.

1 Introduction

Coreference resolution is an important problem of determining whether discourse references in text correspond to the same real world entities (Mitkov, 2002). In this paper, we address a restricted version of the coreference resolution problem in the context of information extraction. The information extraction perspective on coreference resolution imposes a limited scope on the set of entities to be resolved. We are not interested in resolving all coreferences in a document, but only those involving entities to be extracted as part of a specific extraction task. Thus, we can safely ignore coreference resolution of those names, noun phrases, and pronouns that are deemed irrelevant to the extraction task at hand.

The extraction-oriented coreference resolution problem is motivated by the Entity Detection and Tracking (EDT) task of the Automatic Content Extraction evaluation (ACE, 2003). The EDT task requires detecting of named, nominal, and pronominal *mentions* and tracking mentions corresponding to the same real world entities. We adopt the ACE convention of using *mentions* for the names, noun phrases, and pronouns, while reserving *entities* to represent the equivalence classes of mentions, i.e. , the sets of mentions corresponding to the same real world entities.

In this paper, we will take an entity mention extraction component as given (the mention extraction component of (Aone and Ramos-Santacruz, 2000)), and consider coreference resolution algorithms that work with already extracted entity mentions.

A brief outline of the paper follows. In Section 2, we survey previous work on coreference resolution. In Section 3, we present our coreference resolution framework that encompasses a number of standard coreference resolution approaches. In Section 4, we introduce a loss-based coreference decoding methodology and present an approximate greedy coreference decoding algorithm. In Section 5, we experimentally evaluate several coreference resolution architectures, in the context of information extraction.

2 Coreference Resolution Overview

The problem of anaphora resolution is often studied (Mitkov, 2002), which is closely related to the coreference resolution problem. *Anaphora* is a phenomenon of referring to a preceding mention in a document. The reference is then called an *anaphor* and the referred mention is termed an *antecedent*. Anaphora resolution problem is often restricted to nominal and pronominal anaphors, thereby ignoring the problem of name coreference, which is extremely important for information extraction. Additionally, since anaphora addresses (literally) only backward references, the infrequent phenomenon of forward references (termed *cataphora*) is not covered by anaphora resolution. In our presentation, the term "coreference resolution" implies resolution of named, nominal, and pronominal entity mentions that subsumes both backward and forward references.

Let us define the coreference relation $coref$ on a set of document entity mentions. We say that the relation $coref(x, y)$ holds if and only if the mentions x and y are coreferent.

It is frequently helpful to compartmentalize the relation $coref(x,y)$ and, hence, the coreference resolution task into three different subtasks corresponding to different kinds of entities involved. More precisely, if x or y is a pronominal entity, then we obtain a pronoun resolution problem. Otherwise, if x or y is a nominal entity, then we have a noun phrase resolution problem. Finally, if both x and y are named entities, then it is a name resolution problem.

An information extraction system needs to address all three aspects of the coreference resolution problem. Yet different modeling and algorithmic choices may be appropriate for name, noun phrase, and pronoun resolution.

Most early work on coreference and anaphora resolution dealt with pronoun coreference (Lappin and Leass, 1994; Kennedy and Boguraev, 1996). The early approaches identified a set of pronouns in a document, and, for each pronoun, sought to determine the best antecedent. Different definitions of "best" led to different carefully designed and complex rules that were sometimes based on existing discourse theories (Sidner, 1979).

The area of pronoun and noun phrase coreference resolution was greatly revitalized since mid-1990s by application of learning approaches to the problem. We note, amongst many, the work of (Aone and Bennett, 1996; McCarthy and Lehnert, 1995; Ng, 2001; Ng and Cardie, 2002).

A coreference example is a feature-based representation of a pair of mentions that is designed to make manifest the properties of the anaphor and its candidate antecedent that are most helpful in making the decision whether the anaphor indeed refers to the antecedent in question. A coreference example has a binary label reflecting whether the entities that constitute the example are indeed coreferent or not. Most learning-based systems for coreference resolution employed larger hand-crafted feature sets (Ng, 2001).

A number of learning algorithms have been experimentally evaluated on the coreference resolution problem. Many published studies employed a decision tree algorithm (Aone and Bennett, 1996; Ng, 2001; Ng and Cardie, 2002). We also note a few global probabilistic modeling approaches to coreference resolution: the generative probabilistic model of (Charniak et al., 1998) and the conditional random filed model of (McCallum and Wellner, 2003).

The coreference classifiers tha are output by learning algorithms need to be used in conjunction with coreference *decoding algorithms* in order to induce the $coref$ equivalence relation on the set of mentions. A most popular coreference decoding algorithm links an anaphor to the first preceding antecedent predicted as coreferent with the anaphor (Ng, 2001). We will call it the *link-first* decoding algorithm. An alternative decoding algorithm (termed *link-best*) links the anaphor to the most probable preceding antecedent, where the probability of antecedent is taken to the confidence of the coreference classifier prediction (Ng and Cardie, 2002). We will consider both *link-first* and *link-best* decoding algorithms and compare them with the new decoding framework that we introduce in Section 4.

Our decoding framework most resembles the work of (McCallum and Wellner, 2003), where a coreference model represents a conditional random field. The coreference decoding problem for the conditional random field leads to a correlation clustering problem (Bansal et al., 2002). We also reduce the coreference decoding problem to a correlation clustering problem, but use a different approximation algorithm for its solution.

In the absence of training data, we note application of clustering for coreference of noun phrases (Cardie and Wagstaff, 1999). Namely, the noun phrase attributes are used to define a distance function that is used within a heuristic clustering algorithm to produce a clustering of noun phrases that aims to correspond to the coreference partition of the corresponding noun phrase entities.

In addition to the work on coreference resolution within documents, there is an emerging body on a more general *identity uncertainty* problem, which is concerned with determining whether two records describe the same entity (Pasula et al., 2003).

3 Coreference Resolution Architecture

We consider the following components in adaptive approaches to coreference resolution.

- Coreference examples and their feature representation.
- Coreference examples' generation process.
- Learning algorithms for coreference classifiers.

- Decoding algorithm that combines predictions of coreference classifiers into a coherent discourse interpretation.

Let us examine the components in more detail.

3.1 Coreference Examples and Classifiers

We use entity mentions produced by the mention extraction system (Aone and Ramos-Santacruz, 2000) that augments mentions with the following additional attributes (when appropriate): `POS` (a part of speech tag from a restricted set of POS tags), `head` (the mention head, mostly pertinent for named and nominal mentions), `gender` (the gender of the mention, if available), `number` (plurality of the mention), `first name` (the first name of the mention, if available), `last name` (the last name of the mention, if available), `determiner` (the string value of the mention determiner, if available), `personal pronoun type` (for pronouns, the first/second/third person pronoun indicator), `possessive pronoun type` (for pronouns, the possessive pronoun indicator).

For an anaphor/antecedent pair, we will use the following set of features in coreference examples.

- Every conjunction $A = X \wedge B = Y$, where A is an anaphor attribute, B is an antecedent attribute, and X and Y are the corresponding values of the attributes in the given anaphor and the antecedent.
- For every common attribute A of an anaphor and an antecedent, the value of the proposition, $anaphor.A = antecedent.A$ that reflects the same attributes have the same value in both the given anaphor and the antecedent.
- For nominal and pronominal anaphors, we used distance features (number of words/sentences/paragraphs between an anaphor and an antecedent). The distance features are discretized using the entropy-based discretization algorithm with the minimal description length stopping criterion (Dougherty et al., 1995).
- For nominal anaphors, we used the "appositive" feature, if the anaphor is in apposition with the antecedent.
- For named anaphors, we used the substring feature (indicating whether one name is a substring of the other) and the common acronym/alias feature (indicating whether one name is a common acronym/alias of the other).

We note that the extracted mentions are typed, and we will consider only an anaphor/antecedent pair, where the anaphor and the antecedent belong to the same type (e.g., we will not try to corefer a `person` with an `organization`). We also incorporate additional knowledge in the coreference resolution process by imposing constraints on the set of possible anaphor/antecedent pairs. Note that, for the sake of brevity, we use the words "anaphor" (*ana*) and "antecedent" (*ante*) for both anaphora and cataphora phenomena. The constraints are specified by the following relation $IsCandidateAntecedent$:

$$isCandidateAntecedent(anaphora, antecedent) = \\ = \begin{cases} 1, & \text{if } anaphora \text{ is pronominal,} \\ & \text{and } anaphora \text{ follows } antecedent \\ 1, & \text{if } anaphora \text{ is pronominal,} \\ & \text{and } antecedent \text{ is not pronominal} \\ 1, & \text{if } anaphora \text{ is nominal,} \\ & \text{and } antecedent \text{ is not pronominal,} \\ & \text{and it precedes } anaphora \\ 1, & \text{if } ana \text{ is nominal,} \\ & \text{and } ante \text{ is a name} \\ 1, & \text{if } anaphora \text{ is a name,} \\ & \text{and } antecedent \text{ is a name} \\ & \text{that precedes } anaphora \\ 0, & \text{otherwise} \end{cases}$$

The constraints restrict the list of possible antecedents for different classes of anaphora by incorporating coreference knowledge. The knowledge specifies that pronominal anaphora never refer forward to other pronouns, that nominal anaphora refer to preceding nominals or names, and names refer only to preceding names.

We also relax the assumption of the single classifier. Instead, we split the coreference resolution classifier into several distinct *projected* classifiers depending on the type and level of an anaphor. The classifiers are presented in Table 1. The split is a result of the fact that different features are appropriate for different types and levels of an anaphor. For example, while the distance between an anaphor and antecedent (in terms of words, sentences, paragraphs) might be extremely useful for pronominal anaphors, it is not a valuable feature for name coreference resolution. Also, different coreference usage patterns may be prevalent for different kinds of anaphor. For instance, we may expect that the pronouns “I” and “it” behave differently with respect to coreference phenomena. Hence, different models may be appropriate for different kinds of pronominal anaphors.

Anaphor ($Type = t$)	Classifier
Name	$c_{name,t}$
Nominal	$c_{nominal,t}$
1st person pronoun	c_{first}
2nd person pronoun	c_{second}
It pronoun	c_{it}
Plural 3rd person pronoun	c_{they}
Singular 3rd person pronoun	c_{third}

Table 1: Coreference classifiers

Thus, for the 5 entity types used in our evaluation in Section 5, we learn 15 distinct coreference classifiers.

3.2 Coreference Example Generation

We employ two strategies for coreference example generation. For classifiers used in the *link-first* decoding algorithm, we proceed from a fixed anaphora backward (in text), and generate a negative example for each candidate antecedent until an antecedent coreferent with the anaphor is encountered. A positive example is generated for the antecedent, and the process of generating examples for the fixed anaphor stops. We also impose an upper bound M on the number of preceding candidate antecedents that we consider. If none of the M preceding candidate antecedents is coreferent with the anaphor, then we proceed from the anaphor forward in the same fashion until we encounter an antecedent coreferent anaphor or exhaust the M forward candidate antecedents. This is termed a *sequential* example generation process.

For classifiers used in the *link-best* decoding algorithm and the loss-based decoding framework presented in Section 4, we generate an example for every candidate antecedent residing within the window of M candidate antecedents around the anaphor. This is termed an *exhaustive* example generation process.

3.3 Coreference Decoding Algorithms

The coreference decoding procedure combines the predictions of coreference classifiers into a single coherent interpretation.

The most prevalent decoding approach is the *link-first* coreference decoding algorithm (Ng, 2001). The algorithm processes mentions sequentially in order of their appearance in the document and establishes a coreference relation between a mention m_i and the closest preceding candidate antecedent classified as coreferent with m_i by the learned coreference classifier, among the M preceding candidate antecedents. If no such preceding mention is found, the algorithm establishes a coreference relation between the mention m_i and the closest following candidate antecedent classified as coreferent with m_i, among the M following candidate antecedents. After a single pass through the document, the equivalence classes are constructed via the transitive closure of the established coreference relations.

A popular alternative to *link-first* is the *link-best* coreference decoding algorithm (Ng and Cardie, 2002). The algorithm processes mentions sequentially in order of their appearance in the document and establishes a coreference relation between a mention m_i and the most probable candidate antecedent classified as coreferent with m_i by the learned coreference classifier, among the M preceding and M following candidate antecedents. After a single pass through the document, the equivalence classes are again constructed via the transitive closure of the established coreference relations.

3.4 Machine Learning Algorithms

We will use two machine learning algorithms in our experiments: logistic regression (Berkson, 1944) and the (voted) Perceptron algorithm (Freund and Schapire, 1999).

We introduce the following notation. Let $x \in X$ denote a (coreference) example, where $X \subseteq R^N$ is an example space. Let $y \in Y = \{-1, 1\}$ be an example label, where $-1(+1)$ corresponds to a negative (positive) coreference example. We term a pair (x, y) a labeled example.

Let $S = \{(x_1, y_1), \ldots, (x_s, y_s)\}$ be a (training) sample of labeled examples. A learning algorithm A uses the sample S to produce a classifier $c : X \to Y$. Both Perceptron and logistic regression learn linear classifiers[1] in R^N, that is, $c_w(x) = sgn(w^T x)$. Perceptron and logistic regression seek classifiers that minimize particular *loss functions*[2] $l(c, x, y)$ with respect to the

[1] Without loss of generality, we assume that the intercept value of the classifier linear function is equal to 0.

[2] A loss function $l(c, x, y)$ quantifies the error (loss) of the classifier c on the labeled example (x, y).

sample S:

$$c_{w_{opt}} = arg \min_{c_w : w \in R^N} \sum_{i=1}^{s} l(c_w, x_i, y_i) \qquad (1)$$

The Perceptron learning algorithm approximately minimizes the 0-1 loss function $l = l_0 = (sgn(-yw^Tx))_+$, where $(a)_+ = 1$ if $a > 0$ and $(a)_+ = 0$, otherwise. Logistic regression minimizes the logistic loss function $l = l_{log} = ln(1 + e^{-yw^Tx})$.

4 Loss-based Coreference Decoding

Let $M = m_1, \ldots, m_n$ be the set of document mentions of the same type. Let A be a learning algorithm for learning a coreference classifier c mapping a pair of mentions (m_i, m_j) to $\{-1, 1\}$. Let l be a loss function that is being minimized by A.

Let $M_1, M_2, \ldots, M_k$ be an equivalence class partition of M. Define the variable m_{ij} that indicates whether two mentions belong to the same equivalence class, as follows:

$$m_{ij} = \begin{cases} 1, & \text{if } \exists l \in \{1, \ldots, k\}, \text{ so that} \\ & m_i \in M_l \text{ and } m_j \in M_l \\ -1, & \text{otherwise} \end{cases}$$

Let $\mathcal{M} = \{m_{ij}\}$ be an equivalence class partition of the entities M, and let x_{ij} denote the coreference example computed for mentions m_i and m_j. Then, the partition induces the following loss with respect to the classifier c:

$$l(c, \mathcal{M}) = \sum_{i,j} l(c, x_{ij}, m_{ij}) \qquad (2)$$

During decoding, we seek the partition $\mathcal{M}^*$ that minimizes the partition loss (2), given the classifier c.

$$\mathcal{M}^* = argmin_{\mathcal{M}} l(c, \mathcal{M})$$

We note that the classification decisions are not independent of one another, since the equivalence relation is transitive, and $m_{ij} = 1 \wedge m_{jk} = 1$ implies that $m_{ik} = 1$.

Let us denote $w_{ij} = w^T x_{ij}$. In order to make (2) more manageable for our particular loss functions, we observe that

$$\min_{m_{ij}} \sum_{i,j} l(c, x_{ij}, m_{ij}) = \max_{m_{ij}} \sum_{i,j} g(c, x_{ij}, m_{ij})$$

where $g(c, x_{ij}, m_{ij}) = l(c, x_{ij}, -m_{ij}) - l(c, x_{ij}, m_{ij})$. In particular, for the Perceptron 0-1 loss function, we see that $g(c_w, x, y) = l_0(c_w, x, -y) - l_0(c_w, x, y) = sgn(yw^Tx)$, and the objective function (2) becomes

$$\sum_{i,j} sgn(w_{ij} m_{ij}) \rightarrow_{m_{ij}} max \qquad (3)$$

For logistic regression, $g(c_w, x, y) = l_{log}(c_w, x, -y) - l_{log}(c_w, x, y) = ln\left(\frac{1+e^{yw^Tx}}{1+e^{-yw^Tx}}\right) = ln(e^{yw^Tx}) = yw^Tx$, and the objective function (2) becomes

$$\sum_{i,j} w_{ij} m_{ij} \rightarrow_{m_{ij}} max \qquad (4)$$

The optimization problems (3) and (4) are the formulations of the unweighted and weighted versions, respectively, of the correlation clustering problem (Bansal et al., 2002). Both versions are NP-hard, so we have to resort to approximation algorithms for their solution. (Bansal et al., 2002) presents two approximation algorithms for the correlation clustering problem, and (McCallum and Wellner, 2003) use a variant of the `Minimizing Disagreements` algorithm in conjunction with their conditional random field coreference model. We instead introduce a simple greedy decoding algorithm presented in the following section.

4.1 Greedy Coreference Decoding Algorithm

The greedy decoding algorithm incrementally optimizes the (gain) function $\sum_{i,j} g_{ij} m_{ij}$ (where g_{ij} is either w_{ij} or $sgn(w_{ij})$). The logistic regression version of the algorithm is presented as Algorithm 1. The algorithm initially puts each mention into a separate entity and then iteratively merges the entities, while the merge improves the gain function. During each iteration, the pair of entities is selected in a greedy fashion to optimize the gain improvement achieved by the merge. Note that the algorithm iteratively updates the weights between the already merged entities.

We note that no approximation results are known for Algorithm 1. We experimentally evaluate the greedy decoding algorithm in Section 5 and compare it with *link-first* and *link-best* decoding algorithms.

5 Experiments

We evaluate several coreference resolution configurations composed of the different compo-

Algorithm 1 The Greedy Decoding Algorithm

```
$(m_1, m_2, \ldots, m_n)$ is the list of mentions ordered by their location in the document
$c_w$ is the coreference classifier
$\mathcal{M} = \{\{1\}, \{2\}, \ldots, \{n\}\}$
for all $(m_i, m_j)$, $\{i\} \in \mathcal{M}, \{j\} \in \mathcal{M}$ do
  if $isCandidateAntecedent(m_i, m_j)$ then
    $w_{\{i\},\{j\}} = w^T x_{ij}$
  else
    $w_{\{i\},\{j\}} = 0$
  end if
end for
Sort $\{w_{\{i\},\{j\}}\}$
$w_{max} = w_{\mathbf{i}^*,\mathbf{j}^*} = \max_{\mathbf{i} \in \mathcal{M}, \mathbf{j} \in \mathcal{M} \setminus \mathbf{i}} w_{\mathbf{i},\mathbf{j}}$
while $|\mathcal{M}| > 1$ and $w_{max} > 0$ do
  $\mathcal{M} = \mathcal{M} \setminus \{\mathbf{i}, \mathbf{j}\}$
  for all $\mathbf{k} \in \mathcal{M}$ do
    $w_{\mathbf{k},\mathbf{i}^* \cup \mathbf{j}^*} = w_{\mathbf{k},\mathbf{i}^*} + w_{\mathbf{k},\mathbf{j}^*}$
  end for
  $\mathcal{M} = \mathcal{M} \cup \{\mathbf{i}^* \cup \mathbf{j}^*\}$
  $w_{max} = w_{\mathbf{i}^*,\mathbf{j}^*} = \max_{\mathbf{i} \in \mathcal{M}, \mathbf{j} \in \mathcal{M} \setminus \mathbf{i}} w_{\mathbf{i},\mathbf{j}}$
end while
```

nents presented in Section 3. The six different system configurations are shown in Table 2. We set the value of $M = 10$ (the maximum anaphora/antecedent distance) for example generation and *link-best* decoding procedures.

All systems use the same features and the same classifier configuration presented in Section 3.

The data for our experiments comprised a collection of news articles from the last 3 months of 2000 used as development data in the ACE 2003 evaluation (ACE, 2003). The 105 articles were split randomly into the training/testing sets. The training set contained 53 articles, and the testing set contained 52 articles.

We use the evaluation methodology of the Automatic Content Extraction (ACE) program (ACE, 2003). The ACE program quantifies performance of information extraction systems in terms of the value measure that ranges between 100 (a perfect system) to 0 (a system that outputs nothing) to $-\infty$. It is possible for the value to be negative, if a system outputs too many incorrect entities.

We note that our coreference resolution evaluation is somewhat indirect since we measure not the coreference performance per se (e.g., the MUC coreference measure (MUC, 1998)), but the impact on coreference resolution on information extraction performance.

Generation	Algorithm	Decoding
sequential	logistic regression	link-first
exhaustive	logistic regression	link-best
exhaustive	logistic regression	greedy
sequential	Voted Perceptron	link-first
exhaustive	Voted Perceptron	link-best
exhaustive	Voted Perceptron	greedy

Table 2: Coreference Resolution Configurations

	link-first	*link-best*	*greedy*
LR	75.9	74.2	76.4
VP	75.8	75.4	75.8

Table 3: ACE Values for Different Configurations

The ACE evaluation measure is also intrinsically imbalanced, that is, it penalizes for under-merging entities a lot more than it penalizes for over-merging entities. To counter this, we optimized the cost ratios[3] of coreference classifiers on the training sets using the 5-fold cross-validation.

5.1 Coreference Evaluation Results

The Table 3 presents the ACE values for the coreference decoding algorithms combined with the corresponding learning algorithms. It is worth noting that human-level performance for the task is circa 85(LDC, 2003). Therefore, the systems achieve more than 85% of the human-level performance.

The results indicate that the greedy decoding algorithm compares favorably with the more traditional coreference decoding approaches. Surprisingly, for our dataset, the *link-best* decoding algorithm did not perform as well as the competing approaches. The slight boost in the performance obtained by employing the weighted version of the greedy decoding algorithm derived from the logistic loss function indicates that the greedy decoding algorithm is able to take into account calibrated loss values of the logistic function.

[3] The cost ratio of C assigns the value of C to an error made on a positive example, and the value of 1 to an error made on a negative example.

6 Discussion

Our work addresses coreference resolution from the principled decoding perspective. While there has been a lot of work on using machine learning for coreference resolution, the decoding algorithms for coreference resolution were rarely studied, and usually considered separately from the underlying learning problems (with the notable exception of (McCallum and Wellner, 2003)). Our coreference decoding methodology couples the learning algorithm loss functions with the decoding objectives and reformulates the decoding problem as a correlation clustering problem with a learned distance metric. The correlation clustering problem is beginning to be widely studied, and we plan to evaluate experimentally many pertinent algorithms proposed within the theoretical computer science community (Charikar et al., 2003; Demaine and Immorlica, 2003).

We presented coreference resolution in the context of information extraction and evaluated extrinsically the performance of several coreference resolution configurations. Perhaps due to the extrinsic evaluation, the conclusions are not clear-cut, and an additional intrinsic evaluation will be necessary to ascertain our results. Nevertheless, we believe that our framework is useful for designing coreference resolution systems, and our results call for further research.

References

2003. Automatic Content Extraction. http://www.nist.gov/speech/tests/ace/index.htm.

C. Aone and S. W. Bennett. 1996. Applying machine learning to anaphora resolution. In S. Wermter, E. Riloff, and G. Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, volume 1040 of *Lecture Nodes in Artificial Intelligence.* Springer Verlag, Berlin.

C. Aone and M. Ramos-Santacruz. 2000. Rees: A large-scale relation and event extraction system. In *Proceedings of the 6th Applied Natural Language Processing Conference.*

N. Bansal, A. Blum, and S. Chawla. 2002. Correlation clustering. In *Proceedings of The 43rd Annual IEEE Symposium on Foundations of Computer Science.*

J. Berkson. 1944. Application of the logistic function to bio-assay. *Jour- nal of the American Statistical Association*, (9).

C. Cardie and K. Wagstaff. 1999. Noun phrase coreference as clustering. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.*

M. Charikar, V. Guruswami, and A. Wirth. 2003. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 524. IEEE Computer Society.

E. Charniak, N. Ge, and J. Hale. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora.*

E. Demaine and N. Immorlica. 2003. Correlation clustering with partial information. In *Proceedings of the 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems.*

J. Dougherty, R. Kohavi, and M. Sahami. 1995. Supervised and unsupervised discretization of continuous features. In *Proc. 12th International Conference on Machine Learning.* Morgan Kaufmann.

Y. Freund and R. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3).

C. Kennedy and B. Boguraev. 1996. Anaphora for everyone: pronominal anaphora resolution without a parser. In *Proceedings of the 16International Conference on Computational Linguistics.*

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

2003. Personal communication. Linguistic Data Consortium.

A. McCallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proceedings of the IJCAI Workshop on Information Integration on the Web.*

J. McCarthy and W. Lehnert. 1995. Using decision trees for coreference resolution. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence.*

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

1998. *Proceedings of the 6th Message Undertanding Conference (MUC-7).*

V. Ng and C. Cardie. 2002. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics.*

W. Soon; D. Lim; H. Ng. 2001. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 12.

H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. 2003. Identity uncertainty and citation matching. In *Advances in Neural Information Processing Systems 15*. MIT Press.

C. Sidner. 1979. Toward a computational theory of definite anaphora comprehension in English. Technical report, MIT Press.