Proceedings of the 2009 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT), Boulder, CO, June 2009, pp. 575-583.

Graph-Cut-Based Anaphoricity Determination for Coreference Resolution

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

Abstract

Recent work has shown that explicitly identifying and filtering non-anaphoric mentions prior to coreference resolution can improve the performance of a coreference system. We present a novel approach to this task of anaphoricity determination based on graph cuts, and demonstrate its superiority to competing approaches by comparing their effectiveness in improving a learning-based coreference system on the ACE data sets.

1 Introduction

Coreference resolution is the problem of identifying which noun phrases (NPs, or *mentions*) refer to the same real-world entity in a text or dialogue. According to Webber (1979), coreference resolution can be decomposed into two complementary tasks: "(1) identifying what a text potentially makes available for anaphoric reference and (2) constraining the candidate set of a given anaphoric expression down to one possible choice." The first task is nowadays typically formulated as an *anaphoricity determination* task, which aims to classify whether a given mention is anaphoric or not. Knowledge of anaphoricity could improve the precision of a coreference system, since non-anaphoric mentions do not have an antecedent and therefore do not need to be resolved.

Previous work on anaphoricity determination can be broadly divided into two categories (see Poesio et al. (2004) for an overview). Research in the first category aims to identify specific types of non-anaphoric phrases, with some identifying pleonastic *it* (using heuristics [e.g., Paice and Husk (1987), Lappin and Leass (1994), Kennedy and Boguraev (1996)], supervised approaches [e.g., Evans (2001), Müller (2006), Versley et al. (2008)], and distributional methods [e.g., Bergsma et al. (2008)]), and others identifying non-anaphoric definite descriptions (using rule-based techniques [e.g., Vieira and Poesio (2000)] and unsupervised techniques [e.g., Bean and Riloff (1999)]).

On the other hand, research in the second category focuses on (1) determining the anaphoricity of *all* types of mentions, and (2) using the resulting anaphoricity information to improve coreference resolution. For instance, Ng and Cardie (2002a) train an anaphoricity classifier to determine whether a mention is anaphoric, and let an independently-trained coreference system resolve only those mentions that are classified as anaphoric. Somewhat surprisingly, they report that using anaphoricity information adversely affects the performance of their coreference system, as a result of an overly *conservative* anaphoricity classifier that misclassifies many anaphoric mentions as non-anaphoric. One solution to this problem is to use anaphoricity information as *soft* constraints rather than as hard constraints for coreference resolution. For instance, when searching for the best partition of a set of mentions, Luo (2007) combines the probabilities returned by an anaphoricity model and a coreference model to score a coreference partition, such that a partition is penalized whenever an anaphoric mention is resolved. Another, arguably more popular, solution is to "improve" the output of the anaphoricity classifier by exploiting the dependency between anaphoricity determination and coreference resolution. For instance, noting that Ng and Cardie's anaphoricity classifier is too conservative, Ng (2004) first parameterizes their classifier such that its conservativeness can be varied, and then tunes this parameter so that the performance of the coreference system is maximized. As another example, Denis and Baldridge (2007) and Finkel and Manning (2008) perform joint inference for anaphoricity determination and coreference resolution, by using Integer Linear Programming (ILP) to enforce the consistency between the output of the anaphoricity classifier and that of the coreference classifier.

While this ILP approach and Ng's (2004) approach to improving the output of an anaphoricity classifier both result in increased coreference performance, they have complementary strengths and weaknesses. Specifically, Ng's approach can directly optimize the desired coreference evaluation metric, but by treating the coreference system as a black box during the optimization process, it does not exploit the potentially useful pairwise probabilities provided by the coreference classifier. On the other hand, the ILP approach does exploit such pairwise probabilities, but optimizes an objective function that does not necessarily have any correlation with the desired evaluation metric.

Our goals in this paper are two-fold. First, motivated in part by previous work, we propose a graph-cut-based approach to anaphoricity determination that combines the strengths of Ng's approach and the ILP approach, by exploiting pairwise coreference probabilities when co-ordinating anaphoricity and coreference decisions, and at the same time allowing direct optimization of the desired coreference evaluation metric. Second, we compare our cut-based approach with the five aforementioned approaches to anaphoricity determination (namely, Ng and Cardie (2002a), Ng (2004), Luo (2007), Denis and Baldridge (2007), and Finkel and Manning (2008)) in terms of their effectiveness in improving a learning-based coreference system. To our knowledge, there has been no attempt to perform a comparative evaluation of existing approaches to anaphoricity determination. It is worth noting, in particular, that Luo (2007), Denis and Baldridge (2007), and Finkel and Manning (2008) evaluate their approaches on *true* mentions extracted from the answer keys. Since true mentions are composed of all the NPs involved in coreference relations but only a subset of the singleton NPs (i.e., NPs that are not coreferent with any other NPs) in a text, evaluating the utility of anaphoricity determination on true mentions to some extent defeats the purpose of performing anaphoricity determination, which precisely aims to identify non-anaphoric mentions. Hence, we hope that our evaluation on mentions extracted using an NP chunker can reveal their comparative strengths and weaknesses.

We perform our evaluation on three ACE coreference data sets using two commonly-used scoring programs. Experimental results show that (1) employing our cut-based approach to anaphoricity determination yields a coreference system that achieves the best performance for all six data-set/scoring-program combinations, and (2) among the five existing approaches, none performs consistently better than the others.

The rest of the paper is organized as follows. Section 2 describes our learning-based coreference system. In Section 3, we give an overview of the five baseline approaches to anaphoricity determination. Section 4 provides the details of our graph-cut-based approach. Finally, we present evaluation results in Section 5 and conclude in Section 6.

2 Baseline Coreference Resolution System

Our baseline coreference system implements the standard machine learning approach to coreference resolution (see Ng and Cardie (2002b), Ponzetto and Strube (2006), Yang and Su (2007), for instance), which consists of *probabilistic classification* and *clustering*, as described below.

2.1 The Standard Machine Learning Approach

We use maximum entropy (MaxEnt) classification (Berger et al., 1996) in conjunction with the 33 features described in Ng (2007) to acquire a model, P_C, for determining the probability that two mentions, m_i and m_j, are coreferent. Hence,

$$P_C(m_i, m_j) = P(\text{COREFERENT} \mid m_i, m_j).$$

In the rest of the paper, we will refer to $P_C(m_i, m_j)$ as the *pairwise coreference probability* between m_i and m_j. To generate training instances, we employ Soon et al.'s (2001) procedure, relying on the training texts to create (1) a *positive instance* for each anaphoric mention, m_j, and its closest antecedent, m_i; and (2) a *negative instance* for m_j paired with each of the intervening mentions, m_{i+1}, $m_{i+2}, \ldots, m_{j-1}$. When training the feature-weight parameters of the MaxEnt model, we use 100 iterations of the improved iterative scaling (IIS) algorithm (Della Pietra et al., 1997) together with a Gaussian prior to prevent overfitting.

After training, the coreference model is used to select an antecedent for each mention in a test text. Following Soon et al. (2001), we select as the antecedent of each mention, m_j, the *closest* preceding mention that is classified as coreferent with m_j, where mention pairs with pairwise probabilities of at least 0.5 are considered coreferent. If no such mention exists, no antecedent will be selected for m_j. In essence, we use a *closest-first* clustering algorithm to impose a partitioning on the mentions.

3 Baseline Approaches to Anaphoricity Determination

As mentioned previously, we will use five existing approaches to anaphoricity determination as baselines in our evaluation. Common to all five approaches is the acquisition of an anaphoricity model, P_A, for determining the probability that a mention, m_j, is anaphoric. Hence,

$$P_A(m_j) = P(\text{ANAPHORIC} \mid m_j)$$

To train P_A, we again employ MaxEnt modeling, and create one training instance from each mention in a training text. Hence, each instance represents a single mention and consists of 37 features that are potentially useful for distinguishing anaphoric and non-anaphoric mentions (see Ng and Cardie (2002a) for a detailed description of these features).[1]

The classification of a training instance — one of ANAPHORIC or NOT ANAPHORIC — is derived directly from the coreference chains in the associated training text. Like the coreference model, the anaphoricity model is trained by running 100 iterations of IIS with a Guassian prior. The resulting model is then applied to a test text to determine the probability that a mention is anaphoric.

In the rest of this section, we provide an overview of the five baseline approaches to anaphoricity determination. We will characterize each approach along two dimensions: (1) whether it attempts to improve P_A, and if so, how; and (2) whether the resulting anaphoricity information is used as hard constraints or soft constraints by the coreference system.

3.1 Ng and Cardie (2002a)

Ng and Cardie (N&C) do not attempt to improve P_A, simply using the anaphoricity information it provides as hard constraints for coreference resolution. Specifically, the coreference system resolves only those mentions that are determined as anaphoric by P_A, where a mention is classified as anaphoric if the classification threshold is at least 0.5.

3.2 Ng (2004)

P_A may not be "sufficiently" accurate, however, as N&C report a significant drop in the performance of their coreference system after incorporating anaphoricity information, owing in part to their overly *conservative* anaphoricity model that misclassifies many anaphoric mentions as non-anaphoric. To address this problem, Ng (2004) attempts to improve P_A by introducing a threshold parameter c to adjust the conservativeness of P_A as follows. Given a specific c ($0 \leq c \leq 1$), a mention m_j is classified as anaphoric by P_A if and only if $P_A(m_j) \geq c$. It should be easy to see that decreasing c yields progressively less conservative anaphoricity models (i.e., more mentions will be classified as anaphoric). The parameter c is tuned using held-out development data to optimize the performance of the coreference system that employs anaphoricity information (as hard constraints).

In essence, Ng's approach to improving P_A treats the coreference system as a black box, merely selecting the value for c that yields the best score according to the desired coreference evaluation metric on the held-out data. In particular, unlike some of the anaphoricity determination approaches discussed later on, this approach does not attempt to coordinate the anaphoricity decisions and the pairwise coreference decisions. Nevertheless, as mentioned before, a unique strength of this approach lies in its ability to optimize directly the desired coreference

[1] While we train the anaphoricity model using the Ng and Cardie (2002a) feature set, it should be clear that any features that are useful for distinguishing anaphoric and non-anaphoric mentions can be used (e.g., those proposed by Uryupina (2003) and Elsner and Charniak (2007)).

evaluation metric.

3.3 Luo (2007)

Among the five anaphoricity determination approaches, Luo's (2007) is the only one where anaphoricity information is exploited as soft constraints by the coreference model, P_C.

Specifically, Luo's algorithm attempts to find the most probable coreference partition of a given set of mentions. To do so, it scores a partition using the probabilities provided by P_A and P_C. Let us illustrate how this can be done via the following example. Given a document with four mentions, $m_1, \ldots, m_4$, and a partition of the mentions, $\{[m_1, m_3, m_4], [m_2]\}$, automatically produced by some coreference system, Luo's algorithm scores the partition by considering the mentions in the document in a left-to-right manner. As the first mention in the document, m_1 is not anaphoric, and the probability that it is non-anaphoric is 1 – $P_A(m_1)$. Then, the algorithm processes m_2, which according to the partition is non-anaphoric, and the probability of its being non-anaphoric is 1 – $P_A(m_2)$. Next, it processes m_3, which is coreferent with m_1 with probability $P_C(m_1, m_3)$. Finally, it processes m_4, which is coreferent with m_1 and m_3. The probability that m_4 is coreferent with the cluster consisting of m_1 and m_3 is defined to be $\max(P_C(m_1, m_4), P_C(m_3, m_4))$, according to Luo's algorithm. The score of this partition is the product of these four probabilities, two provided by P_A and two by P_C. As can be seen, a partition is penalized whenever a mention that is unlikely to be anaphoric (according to P_A) is being resolved to some antecedent according to the partition.

Nevertheless, it is computationally infeasible to score all possible partitions given a set of mentions, as the number of partitions is exponential in the number of mentions. To cope with this computational complexity, Luo employs the algorithm proposed in Luo et al. (2004) to heuristically search for the most probable partition by performing a beam search through a *Bell tree*. In essence, only the most promising nodes in the tree are expanded at each step of the search process, where the "promise" of a node is defined in terms of the probabilities provided by P_A and P_C, as described above. Details of this process can be found in Luo et al. (2004).

3.4 Denis and Baldridge (2007)

As mentioned before, Denis and Baldridge (D&B) aim to improve the outputs of P_A and P_C by employing Integer Linear Programming (ILP) to perform joint inference for anaphoricity determination and coreference resolution. The ILP approach is motivated by the observation that the outputs of these two models have to satisfy certain constraints. For instance, if P_C determines that a mention, m_j, is not coreferent with any other mentions in the associated text, then P_A should determine that m_j is non-anaphoric. In practice, however, since P_A and P_C are trained independently of each other, this and other constraints cannot be enforced.

ILP provides a framework for *jointly* determining anaphoricity and coreference decisions for a given set of mentions based on the probabilities provided by P_A and P_C, such that the resulting joint decisions satisfy the desired constraints while respecting as much as possible the probabilistic decisions made by the independently-trained P_A and P_C. Specifically, an ILP program is composed of an objective function to be optimized subject to a set of linear constraints, and is created for each test text D as follows. Let M be the set of mentions in D, and P be the set of mention pairs formed from M (i.e., $P = \{(m_i, m_j) \mid m_i, m_j \in M, i < j\}$). Each ILP program has a set of indicator *variables*. In our case, we have one binary-valued variable for each anaphoricity decision and coreference decision to be made by an ILP solver. Following D&B's notation, we use y_j to denote the anaphoricity decision for mention m_j, and $x_{\langle i,j \rangle}$ to denote the coreference decision involving mentions m_i and m_j. In addition, each variable is associated with an *assignment* cost. Specifically, let $c^C_{\langle i,j \rangle} = -\log(P_C(m_i, m_j))$ be the cost of setting $x_{\langle i,j \rangle}$ to 1, and $\bar{c}^C_{\langle i,j \rangle} = -\log(1 - P_C(m_i, m_j))$ be the complementary cost of setting $x_{\langle i,j \rangle}$ to 0. We can similarly define the cost associated with each y_j, letting $c^A_j = -\log(P_A(m_j))$ be the cost of setting y_j to 1, and $\bar{c}^A_j = -\log(1 - P_A(m_j))$ be the complementary cost of setting y_j to 0. Given these costs, we aim to optimize the following objective function:

$$\min \sum_{(m_i, m_j) \in P} c^C_{\langle i,j \rangle} \cdot x_{\langle i,j \rangle} + \bar{c}^C_{\langle i,j \rangle} \cdot (1 - x_{\langle i,j \rangle}) + \sum_{m_j \in M} c^A_j \cdot y_j + \bar{c}^A_j \cdot (1 - y_j)$$

subject to a set of manually-specified *linear* constraints. D&B specify four types of constraints: (1) each indicator variable can take on a value of 0 or 1; (2) if m_i and m_j are coreferent ($x_{\langle i,j \rangle}$=1), then m_j is anaphoric (y_j=1); (3) if m_j is anaphoric (y_j=1), then it must be coreferent with some preceding mention m_i; and (4) if m_j is non-anaphoric, then it cannot be coreferent with any mention. Note that we are *minimizing* the objective function, since each assignment cost is expressed as a negative logarithm value. We use *lp_solve*[2], an ILP solver, to solve this program.

It is easy to see that enforcing consistency using ILP amounts to employing anaphoricity information as hard constraints for the coreference system. Since transitivity is not guaranteed by the above constraints, we follow D&B and use the *aggressive-merge* clustering algorithm to put any two mentions that are posited as coreferent into the same cluster.

3.5 Finkel and Manning (2008)

Finkel and Manning (F&M) present one simple extension to D&B's ILP approach: augmenting the set of linear constraints with the transitivity constraint. This ensures that if $x_{\langle i,j \rangle}$=1 and $x_{\langle j,k \rangle}$=1, then $x_{\langle i,k \rangle}$=1. As a result, the coreference decisions do not need to be co-ordinated by a separate clustering mechanism.

4 Cut-Based Anaphoricity Determination

As mentioned in the introduction, our graph-cut-based approach to anaphoricity determination is motivated by Ng's (2004) and the ILP approach, aiming to combine the strengths of the two approaches. Specifically, like Ng (2004), our approach allows direct optimization of the desired coreference evaluation metric; and like the ILP approach, our approach co-ordinates anaphoricity decisions and coreference decisions by exploiting the pairwise probabilities provided by a coreference model. In this section, we will introduce our cut-based approach, starting by reviewing concepts related to minimum cuts.

4.1 The Minimum Cut Problem Setting

Assume that we want to partition a set of n objects, $\{x_1, x_2, \ldots, x_n\}$, into two sets, Y_1 and Y_2. We have two types of scores concerning the x's and the Y's: *membership* scores and *similarity* scores. The membership score, $mem_{Y_i}(x_j)$, is a non-negative quantity that approximates the "affinity" of x_j to Y_i. On the other hand, the similarity score, $sim(x_j, x_k)$, is a non-negative quantity that provides an estimate of the similarity between x_j and x_k.

Informally, our goal is to maximize each object's net happiness, which is computed by subtracting its membership score of the class it is *not* assigned to from its membership score of the class it is assigned to. However, at the same time, we want to avoid assigning similar objects to different classes. More formally, we seek to minimize the partition cost:

$$\sum_{x_j \in Y_1, x_k \in Y_2} sim(x_j, x_k) + \sum_{x \in Y_1} mem_{Y_2}(x) + \sum_{x \in Y_2} mem_{Y_1}(x)$$

There exists an efficient algorithm for solving this seemingly intractable problem when it is recast as a graph problem. So, let us construct a graph, G, based on the available scores as follows. First, we create two nodes, s and t (called the *source* and the *sink*, respectively), to represent the two classes. Then, we create one "object" node for each of the n objects. For each object, x_j, we add two directed edges, one from s to x_j (with weight $mem_{Y_1}(x_j)$) and the other from x_j to t (with weight $mem_{Y_2}(x_j)$). Moreover, for each pair of object nodes, x_j and x_k, we add two directed edges (one from x_j to x_k and another from x_k to x_j), both of which have weight $sim(x_j, x_k)$. A *cut* in G is defined as a partition of the nodes into two sets, S and T, such that $s \in S$, $t \in T$; and the cost of the cut, $cost(S, T)$, is the sum of the weights of the edges going from S to T. A minimum cut is a cut that has the lowest cost among all the cuts of G. It can be proved that finding a minimum cut of G is equivalent to minimizing the partition cost defined as above. The main advantage of recasting the above minimization problem as a graph-cut problem is that there exist polynomial-time maxflow algorithms for finding a minimum cut.

4.2 Graph Construction

Next, we show how to construct the graph to which the mincut-finding algorithm will be applied. The ultimate goal is to use the mincut finder to partition a given set of mentions into two subsets, so that our coreference system will attempt to resolve only those mentions that are in the subset corresponding to ANAPHORIC. In other words, the resulting

[2] Available from http://lpsolve.sourceforge.net/

anaphoricity information will be used to identify and filter non-anaphoric mentions prior to coreference resolution. The graph construction process, which takes as input a set of mentions in a test text, is composed of three steps, as described below.

Step 1: Mimicking Ng and Cardie (2002a)

To construct the desired graph, G, we first create the source, s, and the sink, t, that represent the classes ANAPHORIC and NOT ANAPHORIC, respectively. Then, for each mention m_n in the input text, we create one node, n, and two edges, sn and nt, connecting n to s and t. Next, we compute w_{sn} and w_{nt}, the weights associated with sn and nt. A natural choice would be to use $P_A(m_n)$ as the weight of sn and $(1 - w_{sn})$ as the weight of nt. (We will assume throughout that w_{nt} is always equal to $1 - w_{sn}$.) If we apply the mincut finder to the current G, it should be easy to see that (1) any node n where $w_{sn} > 0.5$ will be assigned to s, (2) any node n where $w_{sn} < 0.5$ will be assigned to t, and (3) any remaining node will be assigned to one of them. (Without loss of generality, we assume that such nodes are assigned to s.) Hence, the set of mentions determined as anaphoric by the mincut finder is identical to the set of mentions classified as anaphoric by P_A, thus yielding a coreference system that is functionally equivalent to N&C's. This also implies that G shares the same potential weakness as P_A: being overly conservative in determining a mention as anaphoric.

Step 2: Mimicking Ng (2004)

One way to "improve" G is to make it functionally equivalent to Ng's (2004) approach. Specifically, our goal in Step 2 is to modify the edge weights in G (without adding new edges or nodes) such that the mincut finder classifies a node n as anaphoric if and only if $P_A(m_n) \geq c$ for some $c \in [0, 1]$. Now, recall from Step 1 that the mincut finder classifies a node n as anaphoric if and only if $w_{sn} \geq 0.5$. Hence, to achieve the aforementioned goal, we just need to ensure the property that $w_{sn} \geq 0.5$ if and only if $P_A(m_n) \geq c$. Consequently, we compute w_{sn} using a sigmoid function:

$$w_{sn} = \frac{1}{1 + e^{-\alpha \times (P_A(m_n) - c)}}$$

where α is a constant that controls the "steepness" of the sigmoid.[3] It should be easy to verify that the sigmoid satisfies the aforementioned property. As noted before, $w_{nt} = 1 - w_{sn}$ for each node n. Inspired by Ng (2004), the value of the parameter c will be tuned based on held-out development data to maximize coreference performance.

Step 3: Incorporating coreference probabilities

Like Ng's (2004) approach, the current G suffers from the weakness of not exploiting the pairwise probabilities provided by P_C. Fortunately, these probabilities can be naturally incorporated into G as similarity scores. To see why these pairwise probabilities are potentially useful, consider two mentions, m_i and m_j, in a text D that are coreferent and are both anaphoric. Assume that the graph G constructed from D has these edge weights: $w_{si} = 0.8$, $w_{sj} = 0.3$, and $w_{ij} = w_{ji} = 0.8$. Without the similarity scores, the mincut finder will correctly determine m_i as anaphoric but incorrectly classify m_j as non-anaphoric. On the other hand, if the similarity scores are taken into account, the mincut finder will correctly determine both mentions as anaphoric.

The above discussion suggests that it is desirable to incorporate edges between two nodes, i and j, when m_i and m_j are likely to be coreferent (i.e., $P_C(m_i, m_j) \geq c_2$ for some constant c_2). In our implementation, we tune this new parameter, c_2, jointly with c (see Step 2) on development data to maximize coreference performance. While it is possible to imagine scenarios where incorporating pairwise probabilities is not beneficial, we believe that these probabilities represent a source of information that could be profitably exploited via learning appropriate values for c and c_2.[4]

[3] One of the main reasons why we use a sigmoid function (rather than a linear function) is that the weights will still fall within the $[0, 1]$ interval after the transformation, a property that will turn out to be convenient when the pairwise coreference probabilities are incorporated (see Step 3). α is chosen so that the difference between two weights after the transformation is as close as possible to their difference before the transformation. With this criterion in mind, we set α to 0.42 in our experiments.

[4] Incorporating the coreference probabilities can potentially identify some of the anaphoric mentions that would be misclassified otherwise. However, note that the minimum cut algorithm does not maintain the notion of directionality that would allow one to determine that a discourse-new mention (i.e., the first mention of a coreference chain) is not anaphoric. In particular, the algorithm tends to classify all members of a coreference chain, including the first mention, as anaphoric. We did not ex-

5 Evaluation

5.1 Experimental Setup

For evaluation, we use the ACE Phase II coreference corpus, which is composed of three sections: Broadcast News (BNEWS), Newspaper (NPAPER), and Newswire (NWIRE). Each section is in turn composed of a training set and a test set. For each section, we train an anaphoricity model, P_A, and a coreference model, P_C, on the training set, and evaluate P_C (when used in combination with different approaches to anaphoricity determination) on the test set. As noted before, the mentions used are extracted automatically using an in-house NP chunker. Results are reported in terms of recall (R), precision (P), and F-measure (F), obtained using two coreference scoring programs: the MUC scorer (Vilain et al., 1995) and the CEAF scorer (Luo, 2005).

5.2 Results and Discussions

“No Anaphoricity” baseline. Our first baseline is the learning-based coreference system described in Section 2, which does not employ any anaphoricity determination algorithm. Results using the MUC scorer and the CEAF scorer are shown in row 1 of Tables 1 and 2, respectively. As we can see, MUC F-score ranges from 55.0 to 61.7 and CEAF F-score ranges from 55.3 to 61.2.

Duplicated Ng and Cardie (2002a) baseline. Next, we evaluate our second baseline, which is N&C’s coreference system. As seen from row 2 of Tables 1 and 2, MUC F-score ranges from 50.5 to 60.0 and CEAF F-score ranges from 54.5 to 59.4. In comparison to the first baseline, we see drops in F-score in all cases as a result of considerable precipitation in recall, which can in turn be attributed to the misclassification of many anaphoric mentions by the anaphoricity model. More specifically, MUC F-score decreases by 1.7–5.5%, whereas CEAF F-score decreases by 0.5–1.8%. These trends are consistent with those reported in N&C’s paper.

Duplicated Ng (2004) baseline. Our third baseline is Ng’s (2004) coreference system. Recall that this resolver requires the tuning of the conservativeness parameter, c, on held-out data. To ensure a fair comparison between different resolvers, we do not rely on additional data for parameter tuning. Rather, we reserve $\frac{1}{3}$ of the available training data for tuning c, for which we tested values from 0 to 1 in steps of 0.01, and use the remaining $\frac{2}{3}$ of the data for training P_A and P_C. Results are shown in row 3 of Tables 1 and 2, where MUC F-score ranges from 57.0 to 61.9 and CEAF F-score ranges from 55.5 to 60.6. In comparison to the first baseline, we obtain mixed results: MUC F-score increases by 2.0% and 0.2% for BNEWS and NPAPER, respectively, but drops by 0.1% for NWIRE; CEAF F-score increases by 0.2% and 1.1% for BNEWS and NPAPER, respectively, but drops by 0.6% for NWIRE.

Duplicated Luo (2007) baseline. Results of our fourth baseline, in which the anaphoricity and pairwise coreference probabilities are combined to score a partition using Luo’s system, are shown in row 4 of Tables 1 and 2. Here, we see that MUC F-score ranges from 55.8 to 62.1 and CEAF F-score ranges from 56.3 to 61.5. In comparison to the first baseline, performance improves, though insignificantly,[5] in all cases: MUC F-score increases by 0.2–0.8%, whereas CEAF F-score increases by 0.3–1.0%.

Duplicated Denis and Baldridge (2007) baseline. Our fifth baseline performs joint inference for anaphoricity determination and coreference resolution using D&B’s ILP approach. Results are shown in row 5 of Tables 1 and 2, where MUC F-score ranges from 56.2 to 63.8 and CEAF F-score ranges from 56.9 to 61.5. In comparison to the first baseline, MUC F-score always increases, with improvements ranging from 1.2% to 2.1%. CEAF results are mixed: F-score increases significantly for BNEWS, drops insignificantly for NPAPER, and rises insignificantly for NWIRE. The difference in performance trends between the two scorers can be attributed to the fact that the MUC scorer typically under-penalizes errors due to over-merging, which occurs as a result of D&B’s using the aggressive-merge clustering algorithm. In addition, we can see that D&B’s approach performs at least as good as Luo’s approach in all but one case (NPAPER/CEAF).

Duplicated Finkel and Manning (2008) baseline. Our sixth baseline is F&M’s coreference system,

plicitly address this issue, simply letting the coreference clustering algorithm discover that first mentions are non-anaphoric.

[5] Like the MUC organizers, we use Approximate Randomization (Noreen, 1989) for significance testing, with p=0.05.

	Approach to Anaphoricity Determination	Broadcast News R	P	F	Newspaper R	P	F	Newswire R	P	F
1	No Anaphoricity	57.7	52.6	55.0	60.8	62.6	61.7	59.1	58.1	58.6
2	Duplicated Ng and Cardie (2002a)	40.3	67.7	50.5†	52.1	70.6	60.0	43.0	69.3	53.1†
3	Duplicated Ng (2004)	51.9	63.2	57.0	60.0	63.8	61.9	59.3	57.7	58.5
4	Duplicated Luo (2007)	55.4	56.1	55.8	60.6	63.7	62.1	58.4	59.2	58.8
5	Duplicated Denis and Baldridge (2007)	57.3	55.1	56.2*	63.8	63.7	63.8*	60.4	59.3	59.8*
6	Duplicated Finkel and Manning (2008)	56.4	55.3	55.8	63.8	63.7	63.8*	59.7	59.2	59.5
7	Graph Minimum Cut	53.1	67.5	**59.4***	57.9	71.2	**63.9***	54.1	69.0	**60.6***

Table 1: MUC scores for the three ACE data sets. F-scores that represent statistically significant gains and drops with respect to the "No Anaphoricity" baseline are marked with an asterisk (*) and a dagger (†), respectively.

	Approach to Anaphoricity Determination	Broadcast News R	P	F	Newspaper R	P	F	Newswire R	P	F
1	No Anaphoricity	63.2	49.2	55.3	64.5	54.3	59.0	67.3	56.1	61.2
2	Duplicated Ng and Cardie (2002a)	55.9	53.3	54.5	60.7	56.3	58.5	60.6	58.2	59.4
3	Duplicated Ng (2004)	62.5	49.9	55.5	63.5	57.0	60.1	65.6	56.3	60.6
4	Duplicated Luo (2007)	62.7	51.1	56.3	64.6	55.4	59.6	67.0	56.8	61.5
5	Duplicated Denis and Baldridge (2007)	63.8	51.4	56.9*	62.6	53.6	57.8	67.0	56.8	61.5
6	Duplicated Finkel and Manning (2008)	63.2	51.3	56.7*	62.6	53.6	57.8	66.7	56.7	61.3
7	Graph Minimum Cut	61.4	57.6	**59.4***	64.1	59.4	**61.7***	65.7	61.9	**63.8***

Table 2: CEAF scores for the three ACE data sets. F-scores that represent statistically significant gains and drops with respect to the "No Anaphoricity" baseline are marked with an asterisk (*) and a dagger (†), respectively.

which is essentially D&B's approach augmented with transitivity constraints. Results are shown in row 6 of Tables 1 and 2, where MUC F-score ranges from 55.8 to 63.8 and CEAF F-score ranges from 56.7 to 61.3. In comparison to the D&B baseline, we see that F-score never improves, regardless of which scoring program is used. In fact, recall slightly deteriorates, and this can be attributed to F&M's observation that transitivity constraints tend to produce smaller clusters. Overall, these results suggest that enforcing transitivity for coreference resolution is not useful for improving coreference performance.

Our graph-cut-based approach. Finally, we evaluate the coreference system using the anaphoricity information provided by our cut-based approach. As before, we reserve $\frac{1}{3}$ of the training data for *jointly* tuning the two parameters, c and c_2, and use the remaining $\frac{2}{3}$ for training P_A and P_C. For tuning, we tested values from 0 to 1 in steps of 0.1 for both c and c_2. Results are shown in row 7 of Tables 1 and 2. As we can see, MUC F-score ranges from 59.4 to 63.9 and CEAF F-score ranges from 59.4 to 63.8, representing a significant improvement over the first baseline in all six cases: MUC F-score rises by 2.0–4.4% and CEAF F-score rises by 2.6–4.1%. Such an improvement can be attributed to a large gain in precision and a smaller drop in recall. This implies that our mincut algorithm has successfully identified many non-anaphoric mentions, but in comparison to N&C's approach, it misclassifies a smaller number of anaphoric mentions. Moreover, our approach achieves the best F-score for each data-set/scoring-program combination, and significantly outperforms the best baseline (D&B) in all but two cases, NPAPER/MUC and NWIRE/MUC.

6 Conclusions

We have presented a graph-cut-based approach to anaphoricity determination that (1) directly optimizes the desired coreference evaluation metric through parameterization and (2) exploits the probabilities provided by the coreference model when coordinating anaphoricity and coreference decisions. Another major contribution of our work is the empirical comparison of our approach against five existing approaches to anaphoricity determination in terms of their effectiveness in improving a coreference system using automatically extracted mentions. Our approach demonstrates effectiveness and robustness by achieving the best result on all three ACE data sets according to both the MUC scorer and the CEAF scorer. We believe that our cut-based approach provides a flexible mechanism for coordinating anaphoricity and coreference decisions.

Acknowledgments

We thank the three anonymous reviewers for their invaluable comments, Kazi Saidul Hasan for his help on using *lp_solve*, and NSF for its gracious support of this work under Grant IIS-0812261. The description of the minimum cut framework in Section 4.1 was inspired by Pang and Lee (2004).

References

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

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

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

S. Della Pietra, V. Della Pietra, and J. Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):380–393.

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

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

R. Evans. 2001. Applying machine learning toward an automatic classification of *it*. *Literary and Linguistic Computing*, 16(1):45–57.

J. R. Finkel and C. Manning. 2008. Enforcing transitivity in coreference resolution. In *Proc. of ACL-08:HLT Short Papers (Companion Volume)*, pages 45–48.

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

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

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

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

X. Luo. 2007. Coreference or not: A twin model for coreference resolution. In *Proc. of NAACL-HLT*, pages 73–80.

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

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

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

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

V. Ng. 2004. Learning noun phrase anaphoricity to improve conference resolution: Issues in representation and optimization. In *Proc. of the ACL*, pages 151–158.

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

C. Paice and G. Husk. 1987. Towards the automatic recognition of anaphoric features in English text: the impersonal pronoun 'it'. *Computer Speech and Language*, 2.

B. Pang and L. Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proc. of the ACL*, pages 271–278.

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

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

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

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

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

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

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

B. L. Webber. 1979. *A Formal Approach to Discourse Anaphora*. Garland Publishing, Inc.

X. Yang and J. Su. 2007. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proc. of ACL*, pages 528–535.