A Ranking Approach to Pronoun Resolution

Pascal Denis and **Jason Baldridge**

Department of Linguistics

University of Texas at Austin

{denis,jbaldrid}@mail.utexas.edu

Abstract

We propose a supervised maximum entropy ranking approach to pronoun resolution as an alternative to commonly used classification-based approaches. Classification approaches consider only one or two candidate antecedents for a pronoun at a time, whereas ranking allows all candidates to be evaluated together. We argue that this provides a more natural fit for the task than classification and show that it delivers significant performance improvements on the ACE datasets. In particular, our ranker obtains an error reduction of 9.7% over the best classification approach, the twin-candidate model. Furthermore, we show that the ranker offers some computational advantage over the twin-candidate classifier, since it easily allows the inclusion of more candidate antecedents during training. This approach leads to a further error reduction of 5.4% (a total reduction of 14.6% over the twin-candidate model).

1 Introduction

Pronoun resolution concerns the identification of the antecedents of pronominal anaphors in texts. It is an important and challenging subpart of the more general task of coreference, in which the entities discussed in a given text are linked to all of the textual spans that refer to them. Correct resolution of the antecedents of pronouns is important for a variety of other natural language processing tasks, including –but not limited to– information retrieval, text summarization, and understanding in dialog systems.

The last decade of research in coreference resolution has seen an important shift from rule-based, hand-crafted systems to machine learning systems (see [Mitkov, 2002] for an overview). The most common approach has been a *classification* approach for both general coreference resolution (e.g., [McCarthy and Lehnert, 1995; Soon *et al.*, 2001; Ng and Cardie, 2002]) and pronoun resolution specifically (e.g., [Morton, 2000; Kehler *et al.*, 2004]). This choice is somewhat surprising given that pronoun resolution does not directly lend itself to classification; for instance, one cannot take the different antecedent candidates as classes since there

would be far too many overall and also the number of potential antecedents varies considerably for each anaphor.

Despite this apparent poor fit, a classification approach can be made to work by assuming a *binary* scheme in which pairs of candidate antecedents and anaphors are classified as either COREF or NOT-COREF. Many candidates usually need to be considered for each anaphor, so this approach potentially marks several of the candidates as coreferent with the anaphor. A separate algorithm must choose a unique antecedent from that set. The two most common techniques are "Best-First" or "Closest-First" selections (see [Soon *et al.*, 2001] and [Ng and Cardie, 2002], respectively).

A major drawback with the classification approach outlined above is that it forces different candidates for the same pronoun to be considered independently since only a *single* candidate is evaluated at a time. The probabilities assigned to each candidate-anaphor pair merely encode the likelihood of that particular pair being coreferential, rather than whether that candidate is the best with respect to the others. To overcome this deficiency, Yang et al. [2003] propose a *twin-candidate* model that directly compares pairs of candidate antecedents by building a *preference* classifier based on triples of NP mentions. This extension provides significant gains for both coreference resolution [Yang *et al.*, 2003] and pronoun resolution [Ng, 2005].

A more straightforward way to allow direct comparison of different candidate antecedents for an anaphor is to cast pronoun resolution as a *ranking* task. A variety of discriminative training algorithms –such as maximum entropy models, perceptrons, and support vector machines– can be used to learn pronoun resolution rankers. In contrast with a classifier, a ranker is directly concerned with comparing an entire set of candidates at once, rather than in a piecemeal fashion. Each candidate is assigned a conditional probability (or score, in the case of non-probabilistic methods such as perceptrons) with respect to the entire candidate set. Ravichandran et al. [2003] show that a ranking approach outperforms a classification approach for question-answering, and (re)rankers have been successfully applied to parse selection [Osborne and Baldridge, 2004; Toutanova *et al.*, 2004] and parse reranking [Collins and Duffy, 2002; Charniak and Johnson, 2005].

In intuitive terms, the idea is that while resolving a pronoun one wants to compare different candidate antecedents,

rather than consider each in isolation. Pronouns have no inherent lexical meaning, so they are potentially compatible with many different preceding NP mentions provided that basic morphosyntactic criteria, such as number and gender agreement, are met. Looking at pairs of mentions in isolation gives only an indirect, and therefore unreliable, way to select the correct antecedent. Thus, we expect pronoun resolution to be particularly useful for teasing apart differences between classification and ranking models.

Our results confirm our expectation that comparing all candidates together improves performance. Using exactly the same conditioning information, our ranker provides error reductions of 4.5%, 7.1%, and 13.7% on three datasets over the twin-candidate model. By taking advantage of the ranker's ability to efficiently compare many more previous NP mentions as candidate antecedents, we achieve further error reductions. These reductions cannot be easily matched by the twin-candidate approach since it must deal with a cubic increase in the number of candidate pairings it must consider.

We begin by further motivating the use of rankers for pronoun resolution. Then, in section (3), we describe the three systems we are comparing, providing explicit details about the probability models and training and resolution strategies. Section (4) lists the features we use for all models. In section (5), we present the results of experiments that compare the performance of these systems on the Automatic Content Extraction (ACE) datasets.

2 Pronoun Resolution As Ranking

The twin-candidate approach of Yang et al. [2003] goes a long way toward ameliorating the deficiencies of the single-candidate approach. Classification is still binary, but probabilities are conditioned on triples of NP mentions rather than just a single candidate and the anaphor. Each triple contains: (i) the anaphor, (ii) an antecedent mention, and (iii) a non-antecedent mention. Instances are classified as positive or negative depending on which of the two candidates is the true antecedent. During resolution, all candidates are compared pairwise. Candidates receive points for each contest they win, and the one with the highest score is marked as the antecedent.

The twin-candidate model thus does account for the *relative* goodness of different antecedent candidates for the same pronoun. This approach is similar to error-correcting output coding [Dietterich, 2000], an ensemble learning technique which is especially useful when the number of output classes is large. It can thus be seen as a group of models that are individual experts on teasing apart two different candidates. Nonetheless, the approach is still hampered by the fact that these models' probability estimates are only based on two candidates rather than all that are available. This means that unjustified independence assumptions made during model training and usage may still hurt performance.

While it is a common and often necessary strategy to adapt a task to fit a particular modeling approach, pronoun resolution has in fact been unnecessarily *coerced* into classification approaches. While the twin-candidate strategy is an improvement over the single-candidate approach, it does not address the fundamental problem that pronoun resolution is not characterized optimally as a classification task. The nature of the problem is in fact much more like that of parse selection. In parse selection, one must identify the best analysis out of some set of parses produced by a grammar. Different sentences of course produce very different parses and very different numbers of parses, depending on the ambiguity of the grammar. Similarly, we can view a text as presenting us with different analyses (candidate antecedents) which each pronoun could be resolved to. (Re)ranking models are standardly used for parse selection (e.g., [Osborne and Baldridge, 2004]), while classification has never been explored, to our knowledge.

In classification models, a feature for machine learning is actually the combination of a contextual predicate[1] combined with a class label. In ranking models, the features simply are the contextual predicates themselves. In either case, an algorithm is used to assign weights to these features based on some training material. For rankers, features can be shared across different outcomes (e.g., candidate antecedents or parses) whereas for classifiers, every feature contains the class label of the class it is associated with. This sharing is part of what makes rerankers work well for tasks that cannot be easily cast in terms of classification: features are not split across multiple classes and instead receive their weights based on how well they predict correct outputs rather than correct labels. The other crucial advantage of rankers is that all candidates are trained together (rather than independently), each contributing its own factor to the training criterion. Specifically, for the maximum entropy models used here the computation of a model's expectation of a feature (and the resulting update to its weight at each iteration) is directly based on the probabilities assigned to the different candidates [Berger *et al.*, 1996]. From this perspective, the ranker can be viewed as a straightforward generalization of the twin-candidate classifier.

The idea of ranking is actually present in the linguistic literature on anaphora resolution. It is at the heart of Centering Theory [Grosz *et al.*, 1995] and the Optimality Theory account of Centering Theory provided by Beaver [2004]. Ranking is also implicit in earlier hand-crafted approaches such as Lappin and Leass [1994], wherein various factors are manually given weights, and goes back at least to [Hirst, 1981].

3 The Three Systems

Here, we describe the three systems that we compare: (1) a single-candidate classification system, (2) a twin-candidate classification system, and (3) a ranking system. For each system, we give the probability models and the training and resolution procedures. All model parameters are estimated using maximum entropy [Berger *et al.*, 1996]. Specifically, we estimate parameters with the limited memory variable metric algorithm implemented in the Toolkit for Advanced Discriminative Modeling[2] [Malouf, 2002]. We use a Gaussian prior

---

[1]Examples of contextual predicates are whether the antecedent is a proper name in coreference or whether an $S \rightarrow NP\ VP$ expansion occurs in a parse tree in parse selection.

[2]Available from `tadm.sf.net`.

with a variance of 1000 — no attempt was made to optimize the prior for each data set or system.

Maxent models are well-suited for the coreference task, because they are able to handle many different, potentially overlapping features without making independence assumptions. Previous work on coreference using maximum entropy includes [Kehler, 1997; Morton, 1999; 2000]. In principle, other discriminative algorithms such as perceptrons and support vector machines could be used for each of the systems, though the output would not be probabilistic for these.

The systems are trained and tested on data originally annotated with coreference chains. This means that in principle, an anaphoric pronoun can have several antecedents. Since pronouns show a strong tendency to take very local antecedents, we take only the *closest antecedent* as an anchor when creating training instances.

We use the following notation for all models: π is an anaphoric pronoun and $\mathcal{A} = \{\alpha_1, \ldots, \alpha_k\}$ is a set of antecedent candidates. The task of pronoun resolution is to predict the correct antecedent $\widehat{\alpha}$ for π out of $\mathcal{A}$.

3.1 The Single-candidate Classifier

For the single-candidate classifier, we use the model, training and test procedures of [Soon *et al.*, 2001].

Model

The single-candidate classification approach tackles coreference in two steps by: (i) estimating the probability, $P_c(\text{COREF}|\langle \pi, \alpha_i \rangle)$, of having a coreferential outcome given a pair of mentions $\langle \pi, \alpha_i \rangle$, and (ii) applying a selection algorithm that will single out a unique candidate out of the subset of candidates α_k for which the probability $P_c(\text{COREF}|\langle \pi, \alpha_k \rangle)$ reaches a particular value (typically .5). Note that in this case, the number of events created for a given pronoun is just the cardinality of the set of candidates.

$$P_c(\text{COREF}|\langle \pi, \alpha_i \rangle) = \frac{exp(\sum_{i=1}^{n} \lambda_i f_i(\langle \pi, \alpha_i \rangle, \text{COREF}))}{\sum_c exp(\sum_{i=1}^{n} \lambda_i f_i(\langle \pi, \alpha_i \rangle, c))} \quad (1)$$

Training

Training instances are constructed based on pairs of mentions of the form $\langle \pi, \alpha_i \rangle$, where π and α_i are the descriptions for an anaphoric pronoun and one of its candidate antecedents, respectively. Each such pair is assigned either a label COREF (i.e. a positive instance) or a label NOT-COREF (i.e. a negative instance) depending on whether or not the two mentions corefer. In generating the training data, we create for each anaphoric pronoun: (i) a *positive instance* for the pair $\langle \pi, \alpha_i \rangle$ where α_i is the closest antecedent for π, and (ii) a *negative instance* for each pair $\langle \pi, \alpha_j \rangle$ where α_j intervenes between α_i and π.

Resolution

Once trained, the classifier is used to select a unique antecedent for each anaphoric pronoun in the test documents. In the Soon et al. [2001] system, this is done for each pronoun π by scanning the text right to left, and pairing π with each preceding mention α_i. Each test instance $\langle \pi, \alpha_i \rangle$ thus formed is then evaluated by the classifier, which returns a probability representing the likelihood that these two mentions are coreferential. Soon et al. [2001] use "Closest-First" selection: that is, the process terminates as soon as an antecedent (i.e., a test instance with probability $> .5$) is found or the beginning of the text is reached.

3.2 The Twin-candidate Classifier

The twin-candidate model was proposed by Yang et al. [2003] in the context of coreference resolution. Ng [2005] more recently used it specifically for the pronoun resolution task; for this reason, we adopt his training and test procedures.

Model

With the twin-candidate approach, resolving anaphoric pronouns is also a two step process. The first step involves estimating the probability $P_{tc}(\text{FIRST}|\langle \pi, \alpha_i, \alpha_j \rangle)$, of the pronoun π corefering to the *first* antecedent candidate α_i. Since this is still binary classification, we have the dual probability $P_{tc}(\text{SECOND}|\langle \pi, \alpha_i, \alpha_j \rangle)$, which expresses the likelihood of the pronoun π being coreferential with the *second* antecedent candidate α_j. As with the single-candidate classifier, the selection of the correct antecedent is done in a separate step based on the parameters learned by the model. But with the twin-candidate approach, the antecedent selection algorithm involves comparing candidates in a pairwise manner.

$$P_{tc}(\text{FIRST}|\langle \pi, \alpha_i, \alpha_j \rangle) = \frac{exp(\sum_{i=1}^{n} \lambda_i f_i(\langle \pi, \alpha_i, \alpha_j \rangle, \text{FIRST}))}{\sum_c exp(\sum_{i=1}^{n} \lambda_i f_i(\langle \pi, \alpha_i, \alpha_j \rangle, c))} \quad (2)$$

Training

Training instances are constructed based on *triples* of mentions of the form $\langle \pi, \alpha_i, \alpha_j \rangle$, where π describes a pronominal anaphor and α_i and α_j are the descriptions for two of its candidate antecedents and α_i is stipulated to be closer to π than α_j. These instances are labeled either FIRST if α_i is the correct antecedent or SECOND if α_j is the correct antecedent. For this to work, one has to add an additional constraint on the creation of instances, namely: exactly one and only one of the two candidates can be coreferential with the pronoun. Note that the number of instances created is rather large; it is in fact cubic (since each triple generates two instances) in the number of mentions in the document if one assumes that all mentions preceding a pronoun are potential candidates. In order to obviate this problem, Ng [2005] suggests using a window of 4 sentences including the sentence of the pronoun, and the immediately preceding three sentences.

Resolution

Once trained, the twin-candidate classifier is used to select a unique antecedent for the given anaphoric pronoun π. Like Yang et al. [2003] and Ng [2005], we use a round robin algorithm to compare the members of the candidate set for π. More specifically, test instances are created for each pair of candidates, α_i and α_j, where α_j precedes α_i. These instances

are presented to the classifier, which determines which one of the candidates is preferred; the winner of the comparison gets one point. Finally, the candidate with the most points at the termination of the round robin competition gets selected as the antecedent for π. We use a window of three sentences as we did in training.

3.3 The Ranker

The following describes our training and resolution procedures for the ranking system.

Model

Viewed as ranking task, pronoun resolution is done in a single step, by computing the probability $P_r(\alpha_i|\pi)$, which is the conditional probability of a particular candidate α_i being the antecedent of the anaphoric pronoun π. Here, a unique event is created for each pronoun and its *entire* candidate set $\mathcal{A}$. Finally, selecting the correct antecedent merely boils down to picking the most likely candidate in this set.

$$P_r(\alpha_i|\pi) = \frac{exp(\sum\limits_{i=1}^{n} \lambda_i f_i(\pi, \alpha_i))}{\sum\limits_{k} exp(\sum\limits_{i=1}^{n} \lambda_i f_i(\pi, \alpha_k))} \qquad (3)$$

Training

The training instances for the ranker system are built based on an anaphoric pronoun π and the set of its antecedent candidates $\mathcal{A}$. The candidate set is composed of: (i) the closest antecedent for π, which is singled out as such, and (ii) a set of non-antecedents. The construction of the latter set proceeds by taking the closest antecedent as an anchor and adding all the non-antecedents that occur in a window of 4 sentences around it (including the current sentence of the antecedent, the preceding sentence, and the two following sentences). In contrast with the previous models, note that the comparison between the different candidates in $\mathcal{A}$ is here directly part of the training criterion; these are used in the denominator of the above equation.

Resolution

Once trained, the ranker is used to select a unique antecedent for each anaphoric pronoun. Given preference shown by pronouns for local resolutions and in order to reduce testing time, we build our candidate set by taking only the preceding mentions that occur in a window of 4 sentences, including the pronoun's sentence and the 3 sentences preceding it. The ranker provides a probability distribution for the entire candidate set, and the candidate with the highest conditional probability is chosen as the antecedent. In cases of ties, the alternative that is the closest to the pronoun is chosen.

4 Feature selection

In this study, we focused on features obtainable with very limited linguistic processing. Our features fall into three main categories: (i) features of the anaphoric pronoun, (ii) features of antecedent candidate NP, and (iii) relational features (i.e., features that describe the relation between the two mentions). The detailed feature set is summarized in table 1.

Features of the pronoun	
PERS_PRO	T if π is a personal pronoun; else F
POSS_PRO	T if π is a possessive pronoun; else F
THIRD_PERS_PRO	T if π is 3rd person pronoun; else F
SPEECH_PRO	T if π is 1st, 2nd person pronoun; else F
REFL_PRO	T if π is a reflexive pronoun; else F
PRO_FORM	T if π is lower-cased pronoun; else F
PRO_LCONX	POS tag of word on the left of π
PRO_RCONX	POS tag of word on the right of π
PRO_SCONX	POS tags of words around π
Features of the antecedent candidate	
ANTE_WD_LEN	the number of tokens in α's string
PRON_ANTE	T if α is a pronoun; else F
PN_ANTE	T if α is a proper name; else F
INDEF_ANTE	T if α is a indefinite NP; else F
DEF_ANTE	T if α is a definite NP; else F
DEM_ANTE	T if α is a demonstrative NP; else F
QUANT_ANTE	T if α is a quantified NP; else F
ANTE_LCONX	POS tag of word on the left of α
ANTE_RCONX	POS tag of word on the right of α
ANTE_SCONX	POS tags of words around α
ANTE_M_CT	number of times α's string appears previously in the text
NEAREST_ANTE	T if α is the nearest NP candidate compatible in gender, person, and number; else F
EMBED_ANTE	T if α is embedded in another NP; else F
Relational features	
S_DIST	Binned values for sentence distance between π and α
NP_DIST	Binned values for mention distance between π and α
NUM_AGR	T if π and α agree in number; F if they disagree; UNK if either NP's number cannot be determined
GEN_AGR	T if π and α agree in gender; F if they disagree; UNK if either NP's gender cannot be determined

Table 1: Feature selection for pronoun resolution

For the pronoun features, we encoded into our features information about the particular type of pronoun (e.g., personal, possessive, etc.) and the syntactic context of the anaphoric pronoun. The syntactic context is here approximated as POS tags surrounding the pronoun. For the antecedent candidates, we also use information about the type of NP at hand as well as POS context information. Other features encode the salience of a given antecedent: whether the candidate NP string has been seen up to the current point, whether it is the nearest NP, and whether it is embedded in another larger NP. Finally, we use features describing the relation between the anaphoric NP and its candidate antecedent, namely distance features (in terms of sentences and in terms of NP mentions) and compatibility features (i.e., number and gender agreement). In addition to the simple features described above, we used various composite features. More specifically, we used features combining: (i) distances and the type of the pronoun (e.g., reflexive, possessive), (ii) the named entity for the antecedent with various information on the pronoun, such as the pronoun form and the pronoun gender, (iii) the last three char-

acters in the antecedent head word and the pronoun form and gender.

5 Experiments and Results

5.1 Corpus and evaluation

For evaluation, we used the datasets from the ACE corpus (Phase 2). This corpus is divided into three parts, corresponding to different genres: newspaper texts (NPAPER), newswire texts (NWIRE), and broadcasted news transcripts (BNEWS). Each of these is split into a `train` part and a `devtest` part. We used the `devtest` material only once, namely for testing. Progress during the development phase was estimated only by using cross-validation on the training set for the NPAPER section.

In our experiments, we used all forms of personal (all persons) and possessive pronouns that were annotated as ACE "markables", i.e., the pronouns associated with the following named entity types: FACility, GPE (geo-political entity), LOCation, ORGanization, PERson, VEHicle, WEApons. This excludes pleonastics and references to eventualities or to non-ACE entities. Together, the three ACE datasets contain 7263 and 1866 such referential pronouns, for training and testing, respectively.

Finally, note that in building our antecedent candidate sets, we restricted ourselves to the *true* ACE mentions since our focus is on evaluating the classification approaches versus the ranking approach rather than on building a full pronoun resolution system. It is worth noting that previous work tends to be vague in both these respects: details on mention filtering or providing performance figures for markable identification are rarely given.

No human-annotated linguistic information is used in the input. The corpus text was preprocessed with the OpenNLP Toolkit[3] (i.e., a sentence detector, a tokenizer, a POS tagger, and a Named Entity Recognizer).

Following common practice in pronoun resolution, we report results in terms of *accuracy*, which is simply the ratio of correctly resolved anaphoric pronouns. Since the ACE data is annotated with coreference *chains*, we assumed that correctly resolving a pronoun amounts to selecting one of the previous elements in chain as the antecedent.

5.2 Comparative results

The results obtained for the three systems on the three ACE datasets are summarized in table (2).

System	BNEWS	NPAPER	NWIRE
SCC	62.2	70.7	68.3
TCC	68.6	74.6	71.1
RK	72.9	76.4	72.4

Table 2: Accuracy scores for the single-candidate classifier (SCC), the twin-candidate classifier (TCC), and the ranker (RK).

As shown by this table, the ranker system significantly outperforms the two classifier systems, with an overall f-score of 74.0%. This corresponds to average (weighted) improvements of 7.2% (i.e., an error reduction of 21%) over the single-candidate classifier and of 2.8% (i.e., an error reduction of 9.7%) over the twin-candidate classifier. The scores obtained for the first dataset NPAPER are substantially better than for the two other datasets; we suspect that this difference is due to the fact that we only did development on that dataset.

5.3 Additional results

In this section, we discuss an additional experiment aimed at getting additional insight into the potential of the ranker. In the previous experiments, we provided a rather limited context for training: we only considered mentions in a window of 4 sentences around the correct antecedent. Our main motivation for doing this was to stay as close as possible to the training conditions given in [Ng, 2005] for the twin-candidate approach, thereby giving it the fairest comparison possible. An open question is to what extent can widening the window of antecedent candidates help the ranker to learn better parameters for pronoun resolution. To answer this question, we ran an experiment on the same three ACE datasets and widened the window of sentences by collecting, in addition to the closest antecedent, all non-antecedents preceding the anaphor up to 10 sentences before the antecedent:[4] The results for this experiment are reported in table (3):

System	BNEWS	NPAPER	NWIRE
RK ($w = 10$)	73.0	77.6	75.0

Table 3: Accuracy scores for the ranker (RK) with a window of 10 sentences.

These figures show a significant improvement on the first two datasets, with an average score of 75.4%. This translates into an average gain of 1.4% or an error reduction of 5.4%.

6 Conclusions

We have demonstrated that using a ranking model for pronoun resolution performs far better than a classification model. On the three ACE datasets, the ranker achieves an error reductions of 9.7% over the twin-candidate classifier, even when both have exactly the same features and experimental settings (e.g., number of sentences from which to consider candidates). Our results thus corroborate Ravichandran et al.'s [2003] similar finding that ranking outperforms classification for question-answering. Clearly, the ability to consider all potential antecedents together, rather than independently, provides the ranker with greater discriminating power.

The round robin nature of the pairwise contests in the twin-candidate approach imposes a restrictive computational cost on its use which limits the number of NP mentions that can be considered in a candidate set. The ranker does not suffer from this limitation, and we show that the ranker achieves a further error reduction of 5.4% (or total of 14.6% over the twin-candidate model) by increasing the size of the candidate set used in training.

---

[3]Available from `opennlp.sf.net`.

[4]As for the other experiments, we use a Gaussian prior of variance 1000, and we maintain the window of 4 sentences for testing.

The most direct comparison with previous results is with Ng [2005], who obtained 76.6% and 81.9% on the newspaper and newswire parts of ACE. Our best results for these parts were 77.6% and 75.0%. Though our focus is on comparing classification versus ranking, it is nonetheless interesting that we match Ng's model on the newspaper texts since we use a much simpler feature set and only a single model rather than a complex ensemble. We would thus expect the use of a ranker in place of the twin-candidate classifier would achieve further improvements for his set-up.

The main difference between the twin-candidate approach and the ranking approach is that under the former, candidates are compared by pairs (the best candidate is the one that has won the most times), whereas in the latter an ordering is imposed on the entire set at once. A potential advantage of the ranking approach is that it could allow one to define features on the candidate set itself. Another advantage of the ranker over the preference classifier is how ranking is obtained: only the ranker guarantees a global winner.

While our ranker outperforms the classifiers outright, some benefit could be gained by using both approaches together. It would be straightforward to integrate classifiers and rankers in an ensemble model. For example, a ranker could use the results of the classifier as features in its model.

Acknowledgments

References

[Beaver, 2004] David Beaver. The optimization of discourse anaphora. *Linguistics and Philosophy*, 27(1), 2004.

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

[Charniak and Johnson, 2005] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of ACL*, Ann Arbor, MI, 2005.

[Collins and Duffy, 2002] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures and the voted perceptron. In *Proceedings of ACL*, pages 263–270, Philadelphia, PA, 2002.

[Dietterich, 2000] Thomas Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 1–15, New York, 2000. Springer Verlag.

[Grosz *et al.*, 1995] Barbara Grosz, Aravind Joshi, and Scott Weinstein. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 2(21), 1995.

[Hirst, 1981] Graeme Hirst. *Anaphora in Natural Language Understanding: A Survey*. Springer-Verlag, 1981.

[Kehler *et al.*, 2004] A. Kehler, D. Appelt, L. Taylor, and A. Simma. The (non)utility of predicate-argument frequencies for pronoun interpretation. In *Proceedings of HLT/NAACL*, pages 289–296, 2004.

[Kehler, 1997] Andrew Kehler. Probabilistic coreference in information extraction. In *Proceedings of EMNLP*, pages 163–173, 1997.

[Lappin and Leass, 1994] Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.

[Malouf, 2002] Robert Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Workshop on Natural Language Learning*, pages 49–55, Taipei, Taiwan, 2002.

[McCarthy and Lehnert, 1995] Joseph F. McCarthy and Wendy G. Lehnert. Using decision trees for coreference resolution. In *Proceedings of IJCAI*, pages 1050–1055, 1995.

[Mitkov, 2002] Ruslan Mitkov. *Anaphora Resolution*. Longman, Harlow, UK, 2002.

[Morton, 1999] Thomas Morton. Using coreference for question answering. In *Proceedings of ACL Workshop on Coreference and Its Applications*, 1999.

[Morton, 2000] Thomas Morton. Coreference for NLP applications. In *Proceedings of ACL*, Hong Kong, 2000.

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

[Ng, 2005] Vincent Ng. Supervised ranking for pronoun resolution: Some recent improvements. In *Proceedings of AAAI)*, 2005.

[Osborne and Baldridge, 2004] Miles Osborne and Jason Baldridge. Ensemble-based active learning for parse selection. In *Proceedings of HLT/NAACL*, pages 89–96, Boston, MA, 2004.

[Ravichandran *et al.*, 2003] Deepak Ravichandran, Eduard Hovy, and Franz Josef Och. Statistical QA - classifier vs re-ranker: What's the difference? In *Proceedings of the ACL Workshop on Multilingual Summarization and Question Answering–Machine Learning and Beyond*, 2003.

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

[Toutanova *et al.*, 2004] Kristina Toutanova, Penka Markova, and Christopher Manning. The leaf projection path view of parse trees: Exploring string kernels for HPSG parse selection. In *Proceedings of EMNLP*, pages 166–173, Barcelona, 2004.

[Yang *et al.*, 2003] X. Yang, G. Zhou, J. Su, and C.L. Tan. Coreference resolution using competitive learning approach. In *Proceedings of ACL*, pages 176–183, 2003.