CS224n Final Project: Anaphora Resolution

Ben Handy 05376717
Tyler Hicks-Wright 05142387
Eric Schkufza 05059547

June 10, 2006

Abstract

This paper presents an overview of the data-structures and algorithms used by our group in the implementation of a statistical anaphora resolver based on the one described by Ge, et al in 1998. Additionally, we provide numerical and graphical comparisons of the relative performance of our implementation and theirs. The remainder of this paper is organized as follows:

Section 1 summarizes the base algorithm that our group developed, as described in "A Statistical Approach to Anaphora Resolution (Ge, et al. 1998). Section 2 provides motivation for extensions to that base algorithm, along with the algorithmic enhancements that we implemented to meet those goals. Section 3 reconsiders the material presented in Sections 1 and 2, but in greater depth. Specifically, we discuss the specific data-structures and optimization techniques that we applied in the implementation of those algorithms. Section 4 describes the experiments that we performed to gauge the efficacy of our system, along with qualitative and quantitative summaries of the results that we obtained. Section 5 examines some possible explanations for the errors exposed in Section 4. And finally, Section 6 summarizes our team's findings, as well as suggests avenues for future research. Additionally, Appendix A presents the figures and data referenced by Section 4.

1 Statistical Anaphora Resolution

The problem of Anaphora Resolution refers to the problem of determining the relationship between pronouns (such as he, she, or it) and the antecedent nouns or noun phrases (such as 'Mr. Harris,' 'Ms. Lewis,' or 'the dog') that they refer to in a set of logically related sentences, hereafter referred to as 'stories.' For instance, in the following story

> *Mr. Harris arrived in Munich on Saturday afternoon. However, he did not meet Ms. Lewis until the following morning.*

there is an anaphora between the pronoun 'he' and the noun 'Mr. Harris.'

Statistical anaphora resolution, is a branch of statistical NLP that relies on large corpora of training data to determine statistical relationships between words, for the purpose of gauging the relationship between pronouns and antecedents in the absence of any higher level expert knowledge of language.

In their landmark 1998 paper, "A Statistical Approach to Anaphora Resolution," Ge, Hale, and Charniak describe a probabilistic architecture for considering written works and identifying the antecedents that the pronouns therein refer to. The algorithm that they present for doing so approximates the probability that a candidate antecedent is associated with a particular pronoun,

$$P(A(p)) = a|p, h, \bar{W}, t, l, s_p, \bar{d}, \bar{M})$$

the semantics of which are described at length in that paper, and omitted here for brevity, into the product of four conditional probabilities, which can be accurately calculated, or approximated, given a large enough body of training data. By generating those probabilities for each of a set of candidate antecedents, and selecting from that set, the one that maximizes its approximated probability, or score, they demonstrate that it is possible to achieve a success rate of over 80% for anaphora resolution.

Motivations for, along with descriptions of those four approximated probabilities: Hobb's score, gender-animaticity score, mention-count score, and parse score, are presented below.

1.1 Hobbs Score

The Hobbs Score derives from Hobb's algorithm, which was first proposed by Hobbs in his 1976 paper, "Pronoun Resolution."

Hobb's algorithm is essentially a method for traversing the parse trees generated by a story, beginning from a node used to represent a pronoun and ending at a node used to represent a noun phrase, which is in turn proposed as an antecedent for that pronoun.

Although Hobb's algorithm, as originally proposed, has stood the test of time, its original formulation relied crucially on assumptions about the structure of parse trees, that although at the time were generally adhered to, have since fallen out of style. Most notably, Hobb's algorithm, assumed the existence of $\bar{N}$ nodes, which are conspicuously absent from the Treebank data set, which both this paper, and Ge's use as an experimental testbed.

Fortunately, Hobb's algorithm has since been modified to work, as nearly as was originally intended, given the conventions of the Treebank dataset. Presented below is an algorithmic description of a modified version of Hobb's algorithm as suggested by Tetreault in "A Corpus-Based Evaluation of Centering and Pronoun Resolution."

1. Begin at the NP node immediately dominating the pronoun being considered.

2. Walk up the tree that that node appears in to the first NP or S node encountered. Call this node X, and call the path used to reach it p.

3. Traverse all branches below X to the left of path p in a left-to-right, breadth-first manner. Propose as the antecedent any NP node that is encountered which has an NP or S node between it and X. If no antecedent is found, proceed to Step 4.

4. If node X is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the story being considered in order of recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, is proposed as an antecedent. If X is not the highest S node in the sentence, continue to Step 5.

5. From node X, go up the tree to the first NP or S node encountered. Call this new node X, and call the path traversed to reach it p.

6. If X is an NP node and if the path p to X did not pass through the N node that X immediately dominates, propose X as an antecedent.

7. Traverse all branches below node X to the left of path p in a left-to-right, breadth-first manner. Propose any NP node encountered as an antecedent.

8. If X is an S node, traverse all branches of node X to the right of path p in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as an antecedent.

9. Go to Step 4.

In addition to identifying antecedents, Hobb's algorithm also provides a method for gauging their distance from the pronouns that it considers. Because Hobb's algorithm returns antecedents in increasing order of distance from a pronoun, if it is allowed to produce a list of antecedents (by continuing execution after discovering an antecedent, rather than by returning immediately) position in that list can be used as a rough measure of distance.

By running Hobb's algorithm against an annotated training corpus, and allowing it to propose a list of candidate antecedents for every pronoun that it encounters, and then recording the number of times that the $i-th$ item in those lists corresponded to the correct antecedent for a given pronoun, it is possible to establish a statistical model of the expected number of items that must appear in those lists before the correct antecedent for a given pronoun appears, hereafter referred to as 'Hobb's distance.'

Accordingly, Hobb's score, which is designed to capture the expected distance between pronouns and the antecedents that they refer to, is defined as follows: The probability that a candidate antecedent occurring at Hobb's distance i in a list of candidate antecedents proposed by Hobb's algorithm for a given pronoun is the ration of the number of times that a correct antecedent was observed at that Hobb's distance in some body of training data, divided by the number of times that Hobb's algorithm generated the correct antecedent for a pronoun, at any Hobb's distance:

$$P(d_H = i|a) = \frac{|\text{correct antecedents at Hobbs dist. i}|}{|\text{correct antecedents}|}$$

Hobb's distance is perhaps a more telling metric of distance than intermittent words, when considering a potential antecedent for a given pronoun. Consider the following two examples:

> *Mr. Harris stayed at home, whereas Ms. Lewis went out. He did his laundry.*

> *Mr. Harris stayed at home, whereas Ms. Lewis went out, after having spent the entire day indoors. He did his laundry.*

Whereas the number of intermittent words between 'Mr. Lewis' and 'he' varies significantly between the two, Hobb's algorithm would identify Mr. Harris at the same Hobb's distance in both cases. This fact suggests that Hobb's algorithm is able to correctly ignore subordinate clauses that appear between antecedents and pronouns, and yet have no bearing on their relationship.

1.2 Gender-Animaticity Score

Gender-Animaticity score represents a rough attempt to determine the likelihood that a candidate antecedent belongs to class of nouns or noun phrases that the pronoun being considered is associated with. For example, in the following sentence,

> *Mr. Harris and Ms. Lewis are getting married. He doesn't seem happy with the idea.*

The pronoun 'he' is understood to refer to 'Mr. Harris' because it is associated with the class of masculine nouns or noun phrases, to which 'Ms. Lewis,' as opposed to 'Mr. Harris,' does not belong.

Similarly, in the following sentence, 'it' clearly refers to the inanimate object that is referenced.

> *Mr. Harris was sitting on the table next to the lamp. It hadn't been dusted in weeks.*

Additionally, the gender-animaticity score can also be used to discriminate between the plurality of the nouns referenced by pronouns such as 'he' and 'they,' which might respectively refer to 'Mr. Harris,' and 'his family,' but not vice-versa.

The gender-animaticity score is calculated by dividing the number of times that a pronoun-antecedent anaphora pair appeared in training data, divided by the total number of times that that antecedent was observed:

$$P(p|w_a) = \frac{|w_a \text{ in the antecedent for } p|}{|w_a|}$$

Oftentimes, however, data for the gender-animaticity score is quite sparse, as most antecedents tend to appear rather infrequently, and it is often the case that antecedents observed in training data do not appear in test data at all. Accordingly, smoothing is applied to compensate for that fact. Specifically, when a previously unobserved antecedent is considered as the referent of a given pronoun, its gender-animaticity score is defined as the prior probability of that pronoun.

In the equation presented above, we interpret $|w_a|$ to be the number of times that the antecedent phrase w_a was the referent of any pronoun, rather than the number of times that it was observed. The motivation behind this assumption is explored further in Section 3.

1.3 Mention-Count Score

Mention-Count score is designed to model the importance of an antecedent in a given story. Essentially, antecedents that appear frequently in a story, are more likely to be the referent of a particular pronoun than those that are only mentioned once or twice. Because the number of times that an antecedent has appeared by the time that the pronoun with which it should be associated appears is a function of how far into a story that pronoun appears, calculation of the mention-count score involves determining how many sentences into the story being considered that that pronoun appears:

$$P(a|m_a, j) = \frac{|\text{antecedent } a \text{ seen } m_a \text{times by sentence } j|}{|w_a|}$$

As was the case for the gender-animaticity score, data tends to be quite spare for the mention-count score, and a certain amount of smoothing is necessary to compensate for that fact.

First, because the number of times that antecedents appear in stories varies wildly, rather than explicitly represent the number of times, $|m_a|$ that an antecedent has appeared, that count is instead bucketed. Doing so allows antecedents that are mentioned in test data, a number of times that had previously never been observed, to be treated identically to those that appeared in training data, a nearly identical number of times.

Second, because most antecedents tend to appear rather infrequently, and it is often the case that a previously unobserved antecedent is encountered during testing, the mention-count score for such antecedents is approximated as one, divided by the number of times that any antecedent was seen $|m_a|$ times by the j'th sentence in a story.

1.4 Parse Score

Parse score is designed to capture the fact that certain language patterns tend to repeat themselves more often than others, and that those patterns can be exploited to determine the relationship between pronouns and antecedents. Specifically, if an antecedent is referred to by a pronoun when it occurs in a certain syntactic relationship, which is defined in terms of the nodes that immediately surround that antecedent in a parse tree, then it is likely to be the referent of the pronoun that is most often associated with that pattern.

To be precise, parse score is defined as follows:

$$P(w_a|h,t,l) = \frac{P(w_a|h,t,l)}{P(w_a|t,l)}$$

where w_a is an antecedent, and after turning the sentence that it appears in into a parse tree, t is the type of the node that it is associated with, l is that node's parent's type, and h is the head, or the somewhat loosely defined, 'most significant' component of that node's parent node.

A simple, yet effective technique for determining a node's head, is taken from the OpenNLP project (http://opennlp.sourceforge.net/), and is described algorithmically, below:

1. If the node being considered has more than two children, the first two of which are NP nodes, and the second of which is a pre-terminal node, return that node's third child.

2. If the node being considered has more than one child,

    (a) If that node's first child is an NP node, perform a left-right depth-first traversal beginning from that node and return the last node encountered.

    (b) If any of that node's children are pre-terminal, and are of type CC, then return the left-most such node that is encountered.

    (c) If any of that node's children are NP nodes, return the left-most such node that is encountered.

3. Return that node.

As parse score tends to be even sparser than the two previously mentioned score, extra measures are taken to ensure that it is smoothed in a reasonable way. As suggested by Charniak in "Statistical Parsing with a Context-free Grammar and Word Statistics," that smoothing is achieved by the equation

$$\begin{aligned} p(a|h,t,l) &= \lambda_1 p(a|h,t,l) \\ &+ \lambda_2 p(a|c_h,t,l) \\ &+ \lambda_3 p(a|t,l) \\ &+ \lambda_4 p(a|t) \end{aligned}$$

where c_h is used to represent the class of node types that the head that is being considered refers to. In practice, head classes are even more loosely defined than heads themselves, and although several techniques have been proposed for their classification, most notably by Charniak, there appears to be no clear consensus on how to do so.

1.5 Anaphora Resolution

Given the four conditional probability estimates presented above, Ge, et al, propose the following method for determining the correct antecedent to associate with a given pronoun.

Beginning from the pronoun being considered, Hobb's algorithm is run repeatedly until an acceptably large number of candidate antecedents are proposed. For each of those antecedents, the four approximate probabilities mentioned above, are multiplied together as an estimate of the true probability that a given antecedent is correct, given the pronoun being considered. Ultimately, the candidate antecedent that produces the largest probability is proposed as a result.

What is important to note about Ge's method is that because the four conditional probabilities described above are only approximations, there is no guarantee that they, or their product, will represent true probability distributions. Fortunately, Ge's method concerns itself only with the maximization of their product. Accordingly, although originally proposed an approximation to a true probability distribution, deriving from independence assumptions, the four probabilities described above, can equally well be though of as scores.

The extensions described in the following section depend largely on that intuition.

2 Extensions

The algorithm presented above has proved to be an effective method for resolving anaphoras. The four component scores that it makes use of correspond to key intuitions that fit together into a natural approximation for the probability that a pronoun is related to an antecedent.

However, while that approximation is somewhat elegant, it perhaps places an unnecessary restriction upon its use in the context of more general scoring functions. Specifically, if one's goal is to correctly resolve as many anaphora as possible, it perhaps makes sense to move away from the intuition that those four component scores be multiplied together to approximate a true probability, and to instead consider them in a more general form, simply as score functions.

Doing so allows one to use the component score described above as a base for a more involved scoring function that itself incorporates additional scoring methods, that although no longer an accurate approximation to a legitimate probability, produces enhanced performance characteristics.

Two such extensions to those component scores, along with a method for combining them with Ge's scoring functions, are described below.

2.1 MaxEnt Antecedent Classification

One possible extension to Ge's scoring functions is a more extensive pronoun-antecedent classification system, which could be used to address what Ge described as an inability to accurately capture the pronoun-antecedent relationships that the gender-animaticity score was aimed at describing. By providing an additional method for capturing an antecedents gender and animaticity, it is perhaps possible to obtain more accurate information about an antecedent's traits than could be obtained by simply counting sparse training data instances.

Such classification could be obtained through the use of a maximum entropy classifier, based on character n-gram models. Training might be accomplished by considering each of the matching pronoun-antecedent pairs in some body of training data, and treating the pronoun being considered as the 'class' of the antecedent. Similarly, during testing, whenever a candidate antecedent is encountered, that classifier could be used to suggest an appropriate pronoun class. If that class is identical to the pronoun being considered, a scoring boost might be awarded. Similarly, if the two differ, than a penalty might instead be applied. The amount of boost or punishment to apply based on the result of such classification could be tuned using a validation set.

2.2 Language Model Scoring

Another possible extension one might choose to implement is the use of a language model. Specifically, given a candidate antecedent for a particular pronoun, a language model could be used to gauge how 'natural' it might sound if that antecedent were substituted for that pronoun.

Such a scoring function is motivated by the intuition that only when the correct antecedent is selected for a particular pronoun, can the sentence that that pronoun appears in be read, given the antecedent substitution, without sounding awkward or grammatically incorrect. Accordingly, because language models attach higher probabilities to grammatically correct, natural sounding sentences, such a probability could be employed as an additional scoring boost or penalty to a candidate antecedent.

A natural choice for such a language model is the Trigram Language Model based on smoothing that uses Absolute Discounting, which presented during the first three weeks of the quarter, and proved empirically to be quite effective at detecting sentence correctness.

2.3 Combining Scoring Functions

Having developed the scoring functions described above, the question remains how they might be combined with those proposed by Ge.

One possible approach is to multiply the two new scores with Ge's original four. However, because doing so would completely remove their products' interpretability as a legitimate probability, there is perhaps little incentive to do so. Specifically, the scoring functions proposed above each operate on drastically different scales, whereas language model score is in the range between zero and one, MaxEnt score is simply a binary yes-no that might be converted into an arbitrary boost-punishment.

And alternate approach would be to add the two new scores to Ge's original four. The advantage of addition over multiplication is that it affords the opportunity to apply different weights to each scoring function, based on their experimentally determined efficacy. However, such an approach perhaps only makes sense if an appropriately large amount of time might be devoted to the tuning of such parameters.

Something of a hybrid approach between the two techniques mentioned might be a product where each term is weighted by an exponent. However, what advantages, if any, such an approach might offer over a simple weighted sum, are unclear.

A final approach is would be something of a voting algorithm. Essentially, each scoring algorithm would be permitted to rank each of the candidate antecedents that they are presented in order of increasing score. Having done so, final scores could be determined by performing a possibly weighted addition of the ranks generated by each function. The most notable flaw in such a design is that the MaxEnt score, as proposed above, does not generate continuous values, but rather a fixed boost or punishment score. Accordingly, if such a technique were to be applied, the MaxEnt score would either have to be incorporated differently, or modified to produce a continuous range of values, neither of which seem very appealing.

Ultimately, given the time constraints related to this project, incorporating the two new scores developed above, with those developed by Ge through simple multiplication is perhaps the most reasonable approach to take. Although the results produced by doing so might ultimately prove inferior to those that could be obtained by deciding otherwise, there is regrettably too little time to allocate the alternative techniques the time that they deserve.

3 Implementation Details

Our group chose to use Java to build our anaphora resolution system. This allowed us to leverage several NLP components that our group developed previously over the course of the quarter. Additionally, the decision to do so allowed our group to make use of NLP data structures such as CounterMap, which were provided on the course web space.

Details of how our group used those tools to implement the algorithms described in Sections 2 and 3, are described in the sections below.

3.1 Hobbs Implementation

Our group's implementation of Hobb's algorithm was essentially identical to the one presented in Section 2, that is attributed to Tetreault. At the suggestion of Chris Manning and Bill MacCartney, given the semantics of the annotations used in the Treebank data set, S nodes were taken as any of the following: S, SBAR, SBARQ, SINV, or SQ. Similarly, N nodes were taken as any of the following: NN, NNS, NNP, NNPS, NX, or PRP.

Even in light of Tetreault's modifications to Hobb's algorithm, there still exist instances in the Treebank data set, that fail to satisfy some subset of the assumptions made by the algorithm as it is written. When such instances presented themselves, which was rarely, our group simply chose to discard the pronoun being considered at the time, so as not to introduce erroneous data into our experiments.

3.2 Gender-Animaticity Implementation

The implementation of the gender-animaticity score was relatively straight-forward, as it simply required a count of antecedents and the pronouns they occur with. Wherever multiple antecedents related to the same pronoun, each of those instances were recorded as data points, rather than just the one that was closest to that pronoun.

Although doing so perhaps somewhat mitigated the data sparsity problem mentioned in Section 2, by introducing as many antecedents into our count repositories as possible, ultimately data sparsity remained a serious issue, given the relative infrequency of antecedent types.

Although the smoothing that our group chose to implement followed Ge's suggestion to simply substitute the prior probability of the pronoun being considered, there appeared to be little intuitive reason to do so. Ultimately the decision was made for lack of a better idea.

3.3 Mention-Count Implementation

The mention count score was similarly easy to implement, as it only required keeping counts of the number of times that an antecedent has been seen in a given story. Our group defined a repeat mention to be an exact string match between two labeled antecedents.

A cursory inspection of the training data that our group ran our algorithms against suggested that most antecedent strings occur exactly once in a story, while many others occur two or three times. Other numbers of occurrences are generally unusual and sporadic. Accordingly, to bucket those counts appropriately, we assigned the counts 1, 2, and 3, their own buckets, 4, 5, and 6, to a single bucket, and all higher counts into a final bucket. Although there was insufficient time to perhaps learn a better bucketing function given validation data, we believe that our decision was a reasonable one given our observations.

While several other bucketing schemes were attempted, a brief, qualitative inspection of the distribution of counts in those buckets suggests that at least among the schemes that we examined, the one that we settled upon proved the most effective.

3.4 Parse-Score Implementation

Implementing the parse probability statistic did not provide much of a challenge given that our training and test data were already marked up with full parse tree information. To determine the type of an antecedent, we simply examined its tree label. To make it easier to retrieve the type of a node's parent in a parse tree (and to make the implementation of Hobbs algorithm easier) we transformed all Tree objects into MTrees, which are identical to Trees, with the exception that they also contain pointers to their parents.

To extract an antecedent's head, we used the algorithm presented in Section 2, that is used in the OpenNLP project. Although doing so made implementation rather straightforward, we believe that such a rule-based approximation is perhaps not entirely accurate, and might ultimately prove to be source of error in our methods.

Once heads and types were extracted, producing parse probability estimates was simply a matter of dividing counts. Some additional smoothing did however, need to be applied to the smoothing components described in Charniak's paper, "Statistical Parsing with a Context-free Grammar and Word Statistics" when appropriate data was unavailable. Specifically, it was necessary to approximate the final statistic $p(a|t)$, when the antecedent being considered had not been seen in combination with the type being considered, as $1/|antecedenttypes|$.

While applying the smoothing described in Section 2, we chose simple lambda values of $\frac{1}{3}$ for each statistic, other than the clustering statistic, which was assigned a weight of 0. The decision to do so was a function of time constraints, which prevented our group from being able to implement some form of parameter tuning based on validation data, and from locating an appropriate resource that described a method for performing head classification. It is undoubtedly the case that had the effort been devoted to some form of intelligent parameter tuning, a substantial improvement in experimental results would have been realized.

3.5 MaxEnt and Language Model Scores

Our group chose to reuse the MaxEnt classifier that we developed during the fourth and fifth weeks of the quarter as the MaxEnt scoring component described in Section 3.

As suggested earlier, each unique pronoun that appeared in training data was taken to represent a unique class to which antecedents might be fit. As our group previously demonstrated that an n-gram based classification system proved to be an effective technique for accurate classification, we opted to classify antecedents based on a combination of their 2, 3, and 4-gram character characteristics.

Our group similarly chose to reuse the Trigram Language model that we developed during the second and third weeks of the quarter as the Language Model scoring component described in Section 3.

The modifications that our group made to that model's original architecture, which permitted application to the BLLIP dataset, allowed for integration with a minimal amount of modification.

3.6 The BLLIP-WSJ Dataset

The dataset that our group chose to use for experimentation was the BLLIP 1987-1989 Wall Street Journal Corpus, Release 1, created by the Brown Laboratory for Linguistic Information Processing. It is organized according to the standard Penn Treebank conventions, with the following exceptions:

- Certain auxiliary verbs (e.g., "have", "been" etc.) are deterministically labeled AUX or AUXG (e.g., "having").

- Root nodes are given the new non-terminal label S1 (as opposed to the empty string in the treebank).

- Number attached to non-terminals indicating coreference are preceded by "#" (as opposed to "-" in the treebank).
- Two new grammatical function tags, PLE and DEI, have been added. These tags are used to mark two forms of non-coreferential pronouns, deictic and pleonastic.

The corpus is also broken up into separate story files, where several sentences on a related topic comprise a story. Coreferences between pronouns and antecedents occur only within stories, and never between them, even if the topic being discussed within both is identical.

During experimentation, it became clear that despite its widespread usage, the BLLIP dataset appears to contain multiple instances of annotations that appear incorrect. Specifically, pronouns are often tagged as referring to antecedents with which they appear to have no logical relation. Similarly, it is often the case that gender and animaticity are incoherent for a given pronoun-antecedent pair. The consequences of this fact are discussed further in Section 5.

3.7 High-Level Details

Training for the system described in Sections 2 and 3 is accomplished by parsing a set of stories from the BLLIP data set. For each story, pronoun-antecedent pairs are extracted and combined into counts for each of the scoring functions described above.

Once training is complete, a separate body of stories are read in from the BLLIP dataset for testing purposes. For each story, pronouns are extracted, as are the antecedents that they are annotated as being associated with. As suggested by Ge and Hobb, sentences often contain dangling pronouns that do not relate to any particular antecedent. For instance, in the sentence

It is raining

there is no antecedent to which 'it' directly refers. Whenever such pronouns are encountered, that is, those that have no associated antecedent in a story, they are discarded. Otherwise, Hobb's algorithm is used to generate a set of fifteen candidate antecedents, to which the scoring functions described above are applied. Whichever of those antecedents achieves the greatest score is returned as a result.

Those returned antecedents are then examined to determine whether or not they are correct, given the annotations of the story that they appear in. The algorithm is applied to a sufficiently large body of test data, and the fraction of correct antecedent selections over all pronouns considered is presented as a measure of its efficacy.

4 Experiments

This section summarizes the experiments used to further test our anaphora resolution system.

4.1 Experiment 1

Our first experiment was to run each of the heuristics listed above individually to see how they perform on their own. In doing this, we chose the 15 top antecedents according to Hobb's algorithm, then rank them by the score given by each heuristic. We used a test set of 1,635 pronouns, over 98 stories to evaluate the percentage that the classifier labeled correctly. The results are displayed in Figure 1. Hobb's algorithm, which by itself finds over 54% of the correct antecedents, provides a very stable footing on which to improve.

4.2 Experiment 2

In order to hopefully improve on the results from Experiment 1, we tested combining the heuristics. A sample of our results can be found in Figure 2. Unfortunately, it seems that combining features as discussed in Ge, et. al. was not as effective as we had initially hoped. At best, other heuristics do not change the ranking achieved by the Hobb's distance. This issue is further discussed in the following section.

4.3 Experiment 3

In an attempt to gain more insight into what might be causing the poor performance, we found the rank of each antecedent according to our scoring functions, and kept a running total of how many of these antecedents were placed in each rank. The output, running only Hobb's score, is displayed in Figure 3. 54.9% of the time, the correct antecedent is ranked first, 76% are placed in the top 3, and 83% in the top 5. From this, it is evident that Hobbs is being effective at suggesting the correct antecedents.

When we run this experiment with more scoring functions, however, we get a more dispersed spread. Using Mention Count, Parse Probability, Gender and MaxEnt we get the graph in Figure 4. In this graph, the correct antecedent is ranked first only 45% of the time, top three only 72%, and top five only 80%. Clearly, the other scoring functions are not adding much value in the current implementation.

5 Error Analysis

To analyze the sources of our anaphora resolution errors, we printed out each candidate proposed by Hobbs, followed by each component score, and finally the composite score and a special marking for the correct candidates. Examination of this output yielded several interesting Examination of that output lead to several interesting observations about the sources of error for each scoring mechanism, along with what improvements might most benefit the composite score generated by our algorithm, and as a result, lead to more accurate identification of antecedents.

5.1 Data Sparsity

The first and most profound source of error that we observed was clearly due to sparsity of training data. While we initially assumed that fact to be a function of the size of our training data, we saw little improvement when that data set was augmented to encompass the entire 1987 portion of the BLLIP-WSJ dataset. The problem instead appears to be a function of the fact that antecedents are exactly the kinds of words that are not commonly repeated, even in large bodies of text. Instead, most often they are proper nouns, referring to very specific entities. Another factor that exaggerates that problem is that those specific entities are often be referred to in a variety of ways, and seeing an entity named one way affords no information related to the different namings that it takes on.

Contributing to the problem even further is the fact that the BLLIP dataset often tags larger phrases than the true antecedent that it presumably intends to refer to. As an example, the antecedent, 'Mr. Harris, who is chairman of the board,' might often be tagged as the referent of 'he,' when clearly, 'Mr. Harris' would suffice. Ultimately this may be a consequence of the BLLIP dataset's convention of only tagging a small set of grammatical types as antecedents. If that proved to be the case, then some method of extracting the most significant part of those antecedents might result in the extraction of less unique, more salient antecedents, and as a result, produce data sets that did not require such extensive smoothing.

The data sparsity problem manifested itself in gender-animaticity score, mention count score, and parse-probability score, as they all relied on counts related to specific antecedents.

Our first attempt at addressing that issue was to extract a head from each antecedent, and to then use that head in all further calculations, instead of the full antecedent itself. This however, did not lead to consistent improvement, possibly because of the inadequacy of our primitive head-finding function.

Our second attempt at addressing the data sparsity problem was to compute statistics separately for each word in a given antecedent. Accordingly, the score that a candidate antecedent was given was defined as the average score of each of its component words. While this technique did have the effect of spreading data out more evenly and resulted in a decreased need for smoothing, it also seemed to dilute the value of each of the original scoring functions, and added significant noise in the form of 'filler' words that by themselves conveyed no useful information relating to the antecedent that they appeared in.

5.2 Smoothing Techniques

As suggested repeatedly in Section 4, the anaphora resolution algorithm that our group implemented left substantial room for the possibility of training smoothing weights based on validation data. Unfortunately, given time constraints, there proved to be little opportunity to adequately explore intelligent techniques for doing so. As the odds that the weights that our group happened upon in an ad-hoc fashion were actually optimal, given the datasets that we considered, are astronomically small, it is almost certainly the case that we might achieve a substantial reduction in misclassifications if we were to do so.

5.3 Combining Scores

As Section 4 indicates, each of the scoring functions that our group implemented, proved moderately successful; no one function was completely unable to correctly identify the appropriate antecedent for some percentage of the pronouns that our algorithm was applied to. Moreover, the figures in Section 4 suggest that while our algorithm only achieves a correct antecedent identification of slightly more than 50%, it is almost never the case that any of the three highest scoring antecedents that it considers are incorrect, given the pronoun being considered.

Such results strongly suggest that perhaps the most significant source of error in our algorithm was the method that we chose to combine the scoring functions that we developed. Clearly, if our group were able to develop a method for performing a closer examination of the three highest-scoring candidate antecedents for a given pronoun, and from that pool, ultimately selecting an antecedent, we might be able to achieve a success rate much nearer to that obtained by Ge.

6 Conclusion

Somewhat surprisingly, our group's initial implementation of the algorithm described by Ge for anaphora resolution, was unable to produce results that agreed with those presented in his paper. Specifically, whereas Ge was able to claim a successful classification rate of over 80%, our group's implementation topped out somewhere in the mid fifties.

As suggested above, presumably the most likely reasons for such a discrepancy were the sparsity of the data that our group considered, the techniques that were applied to smooth that data, and an ambiguity in many of the algorithms and methods applied by Ge in his paper. Accordingly, future research in the anaphora resolution domain might begin with a correspondence with Ge regarding some of the major implementation details that remain unclear in his publication. Where such clarifications still do not resolve the discrepancy between classification rates, a closer inspection of the techniques that were applied for smoothing, along with a more intelligent method for training the weights that those techniques involve might be applied.

Although the extensions to Ge's algorithm investigated by our group did prove rather promising, insofar as their implementation resulted in an anaphora resolution system that rarely ranked the correct antecedent for a given pronoun below the top 20% of those candidate antecedents that it considered.

As suggested in Section 4, a natural extension to the work reported in this paper would be research into more effective methods for combining and interpreting the scores generated by our algorithm. Experimentation suggests that the correct results are there; they want only to be found.

A Graphical Appendix

All figures referred to in this paper are provided below.

B Contribution

Our group compiled this project using pair programming over several late nights in our dorm rooms.

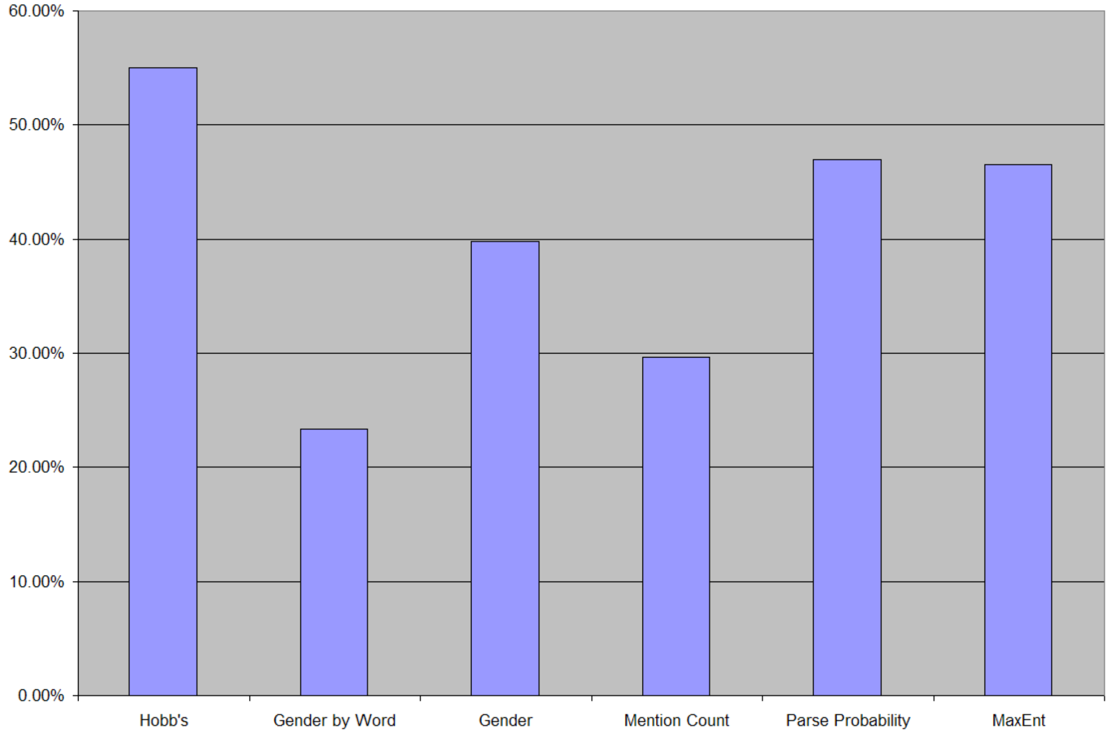
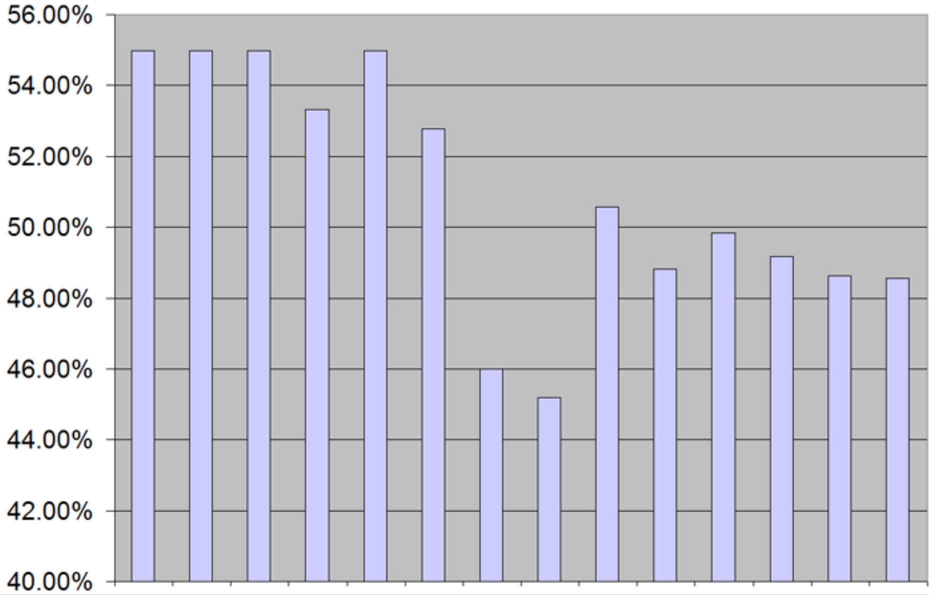


Figure 1: Anaphora resolution accuracy with for each heuristic individually.



Hobbs	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Gender	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Gender by Word	0	0	0	0	0	0	0	0	1	1	1	1	1	1
Mention Count	0	0	0	0	1	1	0	1	0	0	1	1	1	1
Parse Probability	0	0	0	1	0	1	0	1	0	1	0	1	1	1
Language Model	0	0	1	0	0	0	0	0	0	0	0	0	0	1
Max Ent	0	1	0	0	0	0	0	1	0	1	1	0	1	1

Figure 2: Anaphora resolution accuracy with for combined heuristics.

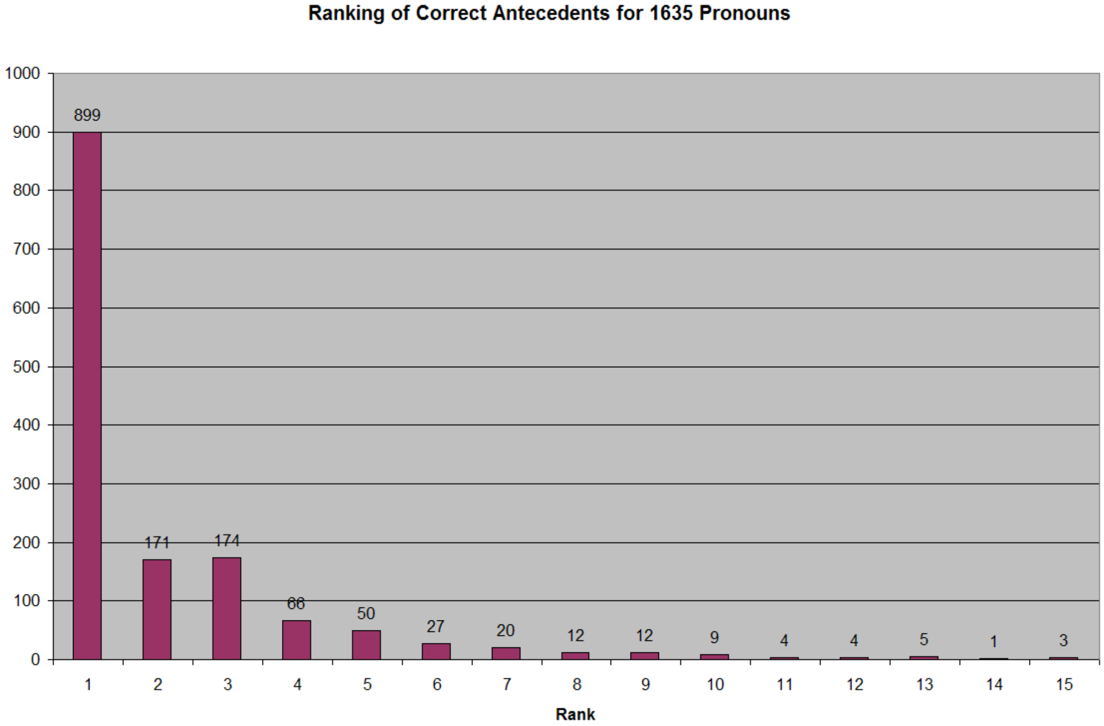


Figure 3: Number of correct antecedents ranked first, second, third, etc. by Hobb's Score.

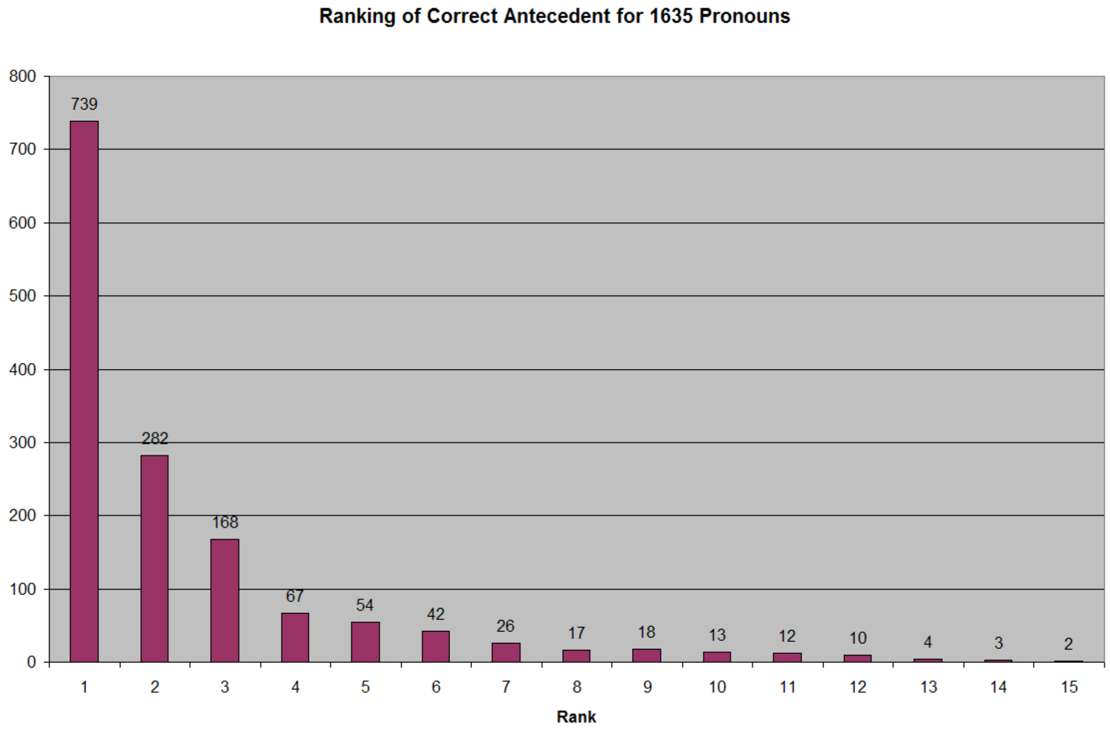


Figure 4: Number of correct antecedents ranked first, second, third, etc. by Hobb's, Gender, Mention Count, Parse Probability and MaxEnt.