Coreference for NLP Applications

Thomas S. Morton
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
tsmorton@cis.upenn.edu

Abstract

This paper presents several techniques for performing automatic coreference annotation and performance results for each of them. To demonstrate that they can be applied to real-world data, we have built a simple question-answering system which uses the techniques. A system using coreference is compared to a baseline system with the result that the addition of the coreference annotation improves performance.

1 Introduction

The fact that two pieces of text specify the same thing in the world can be very helpful in a variety of natural language processing tasks. While there is a vast body of literature on anaphora resolution (Hobbs, 1976; Lappin and Leass, 1994; Mitkov, 1997; Ge et al., 1998), many of these techniques require handcrafted resources, which are difficult to construct, or unrealistic sources of information as input to their algorithms. Here we present a series of techniques for performing coreference between noun phrases which require a limited amount of phrase structure as input and no domain-specific knowledge engineering.

2 Coreference

Coreference annotation involves determining whether or not two noun phrases are used to refer to the same thing. While this is a single task, different types of noun phrases behave differently in terms of how they co-refer. This leads us to use different approaches depending on the type of noun phrase under consideration. Here we consider pronouns, proper nouns, definite nouns, and appositives. For each of these classes we determine what the set of possible antecedents is, a set of factors which influence whether the two nouns corefer with one another, and a decision process for determining which nouns are coreferring.

2.1 Pre-processing

Before coreference on noun phrases can be annotated, the noun phrases, themselves, must first be determined. This processing is common to most NLP applications and includes sentence detection, tokenization, part-of-speech tagging and noun-phrase chunking. The sentence detector we use is described in Reynar and Ratnaparkhi (1997), the tokenizer in Reynar (1998), and the part-of-speech tagger in Ratnaparkhi (1996). The noun-phrase chunking is also done as a tagging task in which tokens are tagged as the start of a noun phrase, the continuation of a noun phrase or other. It employs modeling techniques similar to those used for the part-of-speech tagger. These tools were all trained automatically with data from the Penn Treebank (Marcus et al., 1994).

2.2 Pronoun Model

Pronoun resolution is the most well-studied of the types of coreference discussed here. The pronouns we wish to resolve are all forms of the singular pronouns, *he*, *she*, and *it*. The set of possible antecedents we have chosen is all the basal noun phrases which occur before the pronoun in the same sentence or in the previous two sentences. For these pronouns,

and the data we used, a look-back of two sentences makes the antecedent available 98.7% of the time. A look-back of only one sentence makes the antecedent available 96.9% of the time, which concurs with the findings in Hobbs (1976).

The factors we take into account when determining what noun phrases a pronoun is coreferent with are, in broad terms, locality, gender, syntax, and accessibility or salience. These factors however are not independent and the specific features we use often model more than one factor. Table 1 lists each of the features used and which factors we were attempting to model with them.

Features 1–2 indicate how close the antecedent is to the pronoun. The first feature counts NPs from right to left and the second counts them from left to right. We refer to the latter as the Hobbs distance as it approximates the naive syntax-based ranking presented in Hobbs (1976). Feature 3 gives the antecedent's distance in sentences. It is paired with the pronoun, which allows the model to learn that reflexives should be resolved in the same sentence. Feature 4 provides a crude measure of whether or not the NP is a subject, as the first NP in a sentence is often its subject. Feature 5 models the syntactic context of the antecedent; it can indicate that the antecedent is within a prepositional phrase or modified by a prepositional phrase or relative clause. Feature 6 gives a notion of salience, as an entity which is mentioned repeatedly is likely to be pronominalized. Features 7–10 help determine whether the candidate antecedent is of the correct gender. Thus, the model can learn that a pairing of *Mr.* and male gender makes a good antecedent for *he*, *him*, or *his*. The last two features are only used when determining whether or not the pronoun has a referent. These cases include pleonastic *it* or any other time a pronoun does not refer to a noun phrase.

Using these features and a collection of annotated data we have trained a statistical model to decide when a pronoun is coreferring with another noun phrase or has no referent. Specifically, we employ the maximum

Features	S	G	L	A
1. The distance in NPs between the pronoun and the antecedent			x	
2. Hobbs distance in NPs between the pronoun and the antecedent, and the pronoun's gender	x	x	x	x
3. The distance in sentences between the pronoun and the antecedent and the pronoun	x	x	x	
4 The NP's position in the sentence	x			x
5 The word and POS-tag preceding and following the antecedent	x			x
6 The number of times the antecedent has been mentioned				x
7 The head word of antecedent and the pronoun's gender		x		
8 The head POS-tag of the antecedent and the pronoun's gender		x		
9 The modifier words and POS-tags of antecedent and the pronoun's gender		x		
10 The modifier POS-tags of antecedent and the pronoun's gender		x		
11 The word and POS-tag preceding and following the pronoun	x			
12 The pronoun	x			

Table 1: Features for a Maximum Entropy Pronoun Model and their Motivation. (S=Syntax, G=Gender, L=Locality, A=Accessibility)

entropy framework, which allows us to use a set of binary features, and provides a probability distribution over the set of possible outcomes.[1] The task of resolving pronouns is not naturally a classification task, since the set of possible antecedents differs depending on context. Here we have the model make a binary decision on pronoun/antecedent pairs, and then select the pair with the highest probability. Included as a possible pairing is that the pronoun is non-referential. For this pairing only Features 11–12 are used. If this pairing has the highest probability then the pronoun is left unresolved.

The decision to resolve a particular pronoun is not independent of other coreference decisions. One of the ways these decisions interact is in the computation of Feature 6, the number of times an antecedent has been mentioned. The accuracy of this feature depends on the accuracy of previous coreference decisions including those which do not involve pronouns. Another example of this dependence involves the computation of gender for an antecedent. When a person or company is introduced, often their name is accompanied by an honorific or corporate designator which makes the gender of the entity clear. Later references to this entity might only include a single term, sometimes making determination of gender impossible unless the results of previous coreference decisions are known. One way to capture this phenomenon is by merging antecedents and referents when any coreference relation is posited. When this happens the features of each noun phrase are merged in the following way: The head and modifier words and tags are added to a set of such words and tags for the entity. Thus features 7–10 generate contextual predicates, used by the maximum entropy model, based on every head and modifier word which has been found to be coreferent with the antecedent in question. In contrast, only a single distance measure is kept, which is based on the referent with the lowest Hobbs distance. Feature 6, the number of times this entity has been mentioned is also incremented.

[1] Ratnaparkhi (1997) provides a good introduction on how to use generalized iterative scaling, GIS, to compute the parameters of these models.

Model	P	R	F
w/o entity merging	94.8	71.5	81.5
with entity merging	94.4	76.0	84.2

Table 2: Precision, Recall, and F-Measure for Pronoun Evaluation

2.3 Pronoun Evaluation

We evaluated our model on the data used in Ge et al. (1998). There are 1307 pronouns in this corpus which are forms of *he*, *she*, or *it*. We trained our model on 90% of the data and tested on the other 10%. A feature had to occur at least 5 times in order to be included in the model, and the model parameters were computed using 100 iterations of GIS. The task here was to determine which noun phrase the pronoun referred to or that it was non-referential, given all the previous coreference relationships. While this task is not representative of how the model would be used in practice, namely the other coreference relations are not given and must be computed, the task allows us to evaluate the pronoun model in isolation. Average results for ten-fold cross-validation are presented in Table 2 for our model with and without the entity merging described above. We find that the entity merging improves performance.

2.4 Proper Noun Rules

To apply the pronoun model in practice we need to compute the other types of coreference that contribute to the mention count statistics and word or tag/gender pairs. Coreference between proper nouns is very common in newswire domains and accounts for approximately one third of all coreference relationships (Baldwin et al., 1995). Here we are concerned only with the coreference relationships between two proper nouns and not how proper nouns interact with other nouns or pronouns. A noun is considered a proper noun if its last word has the tag "NNP" or "NNPS". Proper nouns do not have the same locality constraints that pronouns have.

Model	P	R	F
string matching	92.1	88.0	90.0

Table 3: Precision, Recall, and F-Measure for Proper Noun Evaluation

When searching for a possible referent we may have to consider all previous proper nouns as candidates. Determining these types of relationships can be done quite accurately with simple string-matching techniques. The approach we take is that a proper noun is coreferent with a previously occurring proper noun if the subsequent proper noun is a substring of the previous one. The later proper noun is normalized by only considering tokens occurring after and including the first token tagged as a proper noun which does not match the patterns "series of letters ending in a period" or "capital letter followed by sequence of non-vowels". This has the effect of generally removing non-proper noun modifiers and honorifics. The same patterns are also applied to the end of the proper noun to remove corporate designators. Proper nouns can also occur as modifiers to common nouns such as in the example, "A Hitachi spokesman". These can be treated in the same way as other proper nouns. To do this we simply add any sequence of words tagged with "NNP" or "NNPS" that modify a head noun tagged with "NN" or "NNS" to our list of proper nouns and proceed as we did before.

2.5 Proper Noun Evaluation

We performed an evaluation of the above techniques using 80 hand-annotated Wall Street Journal articles. The articles contained 726 coreferring proper nouns. Performance for the above techniques is presented in Table 3. This evaluation does not include modifiers that are proper nouns but we suspect that such an evaluation would produce similar results. Most precision errors involved type mismatches between entities and could likely be addressed with a named-entity detector. Recall was primarily hurt by a lack of treatment of acronyms.

2.6 Common Nouns: Rules and Model

Coreference between common nouns and other nouns is a difficult class in general, but a subset of these cases can be identified with reasonable precision. The cases we consider are definite noun phrases and appositives. Definite noun phrases indicate that the referent should be known to the reader and thus we are likely to find an antecedent for this noun phrase. We consider a noun phrase definite if it is modified by the determiner *the*. For determining coreference between these noun phrases we employ the approach of Vieira and Poesio (1997). The first noun phrase that occurs within the previous five sentences has a string-equivalent head word, and no additional modifiers is considered coreferent with the definite noun phrase.

Another case of common noun coreference is appositives. The coreference between *The asbestos fiber* and *crocidolite* in the following sentence is an example of an appositive:

> The asbestos fiber, crocidolite, is unusually resilient once it enters the lungs, with even brief exposures to it causing symptoms that show up decades later, researchers said.

We consider any two noun phrases separated by a comma as candidates for being an appositive and use the words the noun phrases contain as well as their syntactic context to determine whether they are in fact appositives. To weight these factors we again employ a maximum entropy model. The features used by our model are described in Table 4.

2.7 Common Noun Evaluation

For the definite nouns we hand-tagged 80 Wall Street Journal articles containing 568 referring definite nouns. To train and test our appositive model, we hand-tagged 1000 examples of noun phrases, which were separated by a comma, as either appositives or non-appositives. A feature had to occur at least 5 times in order to be included in the model, and the model parameters were computed using 100 iterations of GIS. For testing,

type	P	R	F
definite NPs	82.5	47.4	60.2
appositives	88.1	79.9	83.8

Table 5: Precision, Recall, and F-Measure for Common Noun Evaluation

1. The token or tag preceding the left noun phrase.
2. The token or tag following the right noun phrase.
5. Either noun phrase contains the head word w_h.
6. Either noun phrase contains the head tag t_h.
7. Either noun phrase contains the modifier word w_m.
8. Either noun phrase contains the modifier tag t_m.
9. The left noun phrase contains the token w_l and the right noun phrase contains the token w_r.
10. The left noun phrase contains the token w_l and the right noun phrase contains the tag t_r.
11. The left noun phrase contains the tag t_l and the right noun phrase contains the token w_r.
12. The left noun phrase contains the tag t_l and the right noun phrase contains the tag t_r.

Table 4: Features for a Maximum Entropy Appositive Model

ten-fold cross-validation was used with 90% of the data used for training and the other 10% left for testing. The results for both types of common nouns are given in Table 5. Recall for the definite nouns is quite low. Attention to definite nouns which do not share the same head as their antecedent could improve this. Specifically, modeling "the company", which occurs frequently in these articles, would help considerably. Appositives suffer from the fact that only adjacent basal noun phrases are considered.

2.8 MUC Evaluation

The different types of coreference described above interact with one another as each one asserts coreference relationships. This occurs because asserting a coreference relationship implies that an entity has been mentioned again. This subsequent mention also effects how local the entity appears to all the coreference models. Failing to assert a coreference relationship may cause an entity to become inaccessible to a pronoun or common noun which is coreferent with it. In order to determine how these components would perform together, we integrated them, and evaluated this system using the MUC-6 Coreference Task (Def, 1995). The two data sets each consist of 30 Wall Street Journal articles that have been hand-annotated for coreference relationships. In these articles almost all coreference between noun phrases has been marked. Results for this task are presented in Table 6. These results are based on the scoring algorithm used for MUC-6 and described in Vilain et al. (1995). Recall for these results is low because there are many types of coreference relationships which we have not attempted to annotate. Precision has remained high, so we can be fairly confident in the coreference relationships that have been

data	P	R	F
dry-run	79.3	41.2	54.2
evaluation	79.6	44.5	57.1

Table 6: Precision, Recall, and F-Measure for MUC Evaluation

found.

2.9 Related Work in Coreference

Our approach for pronoun resolution is similar to that of Ge et al. (1998) in the following ways. Both approaches use statistical modeling to rank antecedents and then select the top ranked one. The features used by both these models include the Hobbs distance and the mention count statistics. They mainly differ in the following ways. Our approach requires noun phrases for input while Ge et al. (1998) uses full parse trees. Ge et al. (1998) does not employ entity merging which we find helps our model. Most importantly, our model handles non-referential pronouns while Ge et al. (1998) excludes them from the data. These differences are primarily the result of our desire to use this model in applications. For an application, full parsing is computationally expensive and non-referential pronouns cannot be excluded from the data. These differences also lead to the exclusion of some helpful features which are employed by Ge et al. (1998) such as a more accurate computation of the Hobbs distance, the probability that an antecedent can occur in the same syntactic position as the pronoun, and the use of unsupervised techniques for gender determination. The result of these trade-offs is poorer performance at pronoun resolution when compared to Ge et al. (1998).[2] The techniques we used for proper noun coreference are similar to those used by Baldwin et al. (1995) and others. For definite nouns we use the approach of Vieira and Poesio (1997). To our knowledge, the approach used for determining appositives is novel.

[2]Evaluating using only referential pronouns gives us an accuracy of 79.1% compared to the 84.2% reported in Ge et al. (1998).

3 Applications

There are numerous applications which use coreference as a base annotation. These include work in summarization (Baldwin and Morton, 1998; Azzam et al., 1999) and question answering (Morton, 1999; Breck et al., 1999).

3.1 Question Answering

Here we present results for a question answering task. Specifically, a system is given a query and then asked to find a 250-byte answer string from a collection of documents. Systems generate a ranked list of the top 5 answer strings. This task is the same as the question answering task used in TREC-8 and described in Voorhees and Tice (1999). As a baseline, we have implemented the system used by AT&T as described in Singal et al. (1999). A description of how that system ranks sentences is given in Table 7.

When we apply this approach to documents with coreference relationships annotated we consider any additional terms which are in coreference chains with noun phrases in a sentence to have occurred in that sentence as well. The terms added via a coreference chains are given 90% of the weight of regularly occurring terms. When presenting the top 5 passages, referents are indicated in parentheses next to the referring terms.

3.2 Question Answering Evaluation

We evaluated both approaches using the 200 questions used in TREC-8 and an evaluation script designed for these questions and provided by NIST. The output is scored using the mean reciprocal rank (MRR).

$$\sum_{q=1}^{n} \frac{1}{rank(q)}$$

Here n is the number of questions processed, and $rank(q)$ is the ranking of the first 250-byte string which answers the question q. The results for each approach are presented in Table 8.

Passage Ranking:
1. The top 50 documents for a question are retrieved using a straight vector match (no query expansion).[3] 2. Each section of these 50 documents is broken into sentences and each sentence is assigned a score based on the following algorithm. 3. The query term weight of every question word that appears in the sentence is added to the sentence score, the passage size is set to the sentence size (in bytes). 4. If a query word bigram appears in the sentence, extra credit[4] is assigned to the sentence. 5. If an adjoining sentence contains a question word not contained in this sentence, and if by adding this adjoining sentence to the passage, the passage size doesn't exceed 500 bytes, half the query term weight for this word is added to the sentence score. 6. If the next adjoining sentence contains a question word not covered yet, and if by adding this adjoining sentence to the passage, the passage size doesn't exceed 500 bytes, a quarter of the query-term weight for this word is added to the sentence score.
Answer Selection:
1. A single top ranked sentence is selected from each document. Ties are broken in favor of longer sentences. 2. Near-duplicate passages are removed. If a low-scoring passage has a cosine-similarity of over 0.50 with a highly ranked passage, the low-scoring passage is removed. 3. The top five sentences from the remaining ones are printed in order of their scores. If a sentence is under 250 bytes the later bytes of the previous sentence are included and then the earlier bytes of the next sentence.

Table 7: AT&T Passage Ranking Algorithm

Model	MRR
baseline	52.3%
coreference	53.8%

Table 8: Mean Reciprocal Rank for TREC-8 Question Answering Task

4 Discussion

The coreference annotation produces a small increase in the performance of the question-and-answer system. Since the baseline estimates coreference by including terms with partial weights from the surrounding sentences, the coreference system will only outperform the baseline when it includes query terms that only occur many sentences away. This scenario occurs infrequently, which is why we only see a slight improvement in a 200 question evaluation. Coreference is essential for finding answers to some of the questions in this set. For the question, "When did Beethoven die?" the system with coreference was able to select the sentence:

> Still, as news spread of his (Beethoven's) final illness (he would die of jaundice and ascites on March 26, 1827), the Philharmonic Society of London sent him a get-well gift of 100 pounds.

having resolved the "his" and "he" to Beethoven while the baseline opted only for sentences which actually contain the term "Beethoven". The coreference annotation described here can be used in other ways which would likely be beneficial to an application. Filling in referents in the extracted text can make text much more coherent to the reader. We see this has been done by the system in the above example. For question-answering this may provide an answer, and in summarization this may allow the reader to determine what the document is about. Coreference between appositives can help determine candidate answers for question-answering as

[3] These documents were provided by AT&T

[4] 0.25 × (lower of the two component query term weights)

they often provide a category for the noun phrase they are coreferent with. We plan to further explore these areas in the future.

5 Conclusion

We have presented several techniques for coreference resolution and evaluated each of them. We have demonstrated that these techniques can be integrated to provide automatic coreference annotation on real-world data and that that annotation can be used to improve natural language processing applications.

References

Saliha Azzam, Kevin Humphreys, and Robert Gaizauskas. 1999. Using coreference chains for text summarization. In *Proceedings of the Workshop on Coreference and Its Applications*, College Park, Maryland, June.

Breck Baldwin and Thomas Morton. 1998. Dynamic coreference-based summarization. In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing*, Granada, Spain, June.

Breck Baldwin, Jeff Reynar, Mike Collins, Jason Eisner, Adwait Ratnaparki, Joseph Rosenzweig, Anoop Sarkar, and Srinivas Bangalore. 1995. Description of the University of Pennsylvania system used for MUC-6. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland.

E. Breck, J. Burger, L. Ferro, D. House, M. Light, and I. Mani. 1999. A Sys called Qanda. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, November. NIST.

Defense Advanced Research Projects Agency (DARPA). 1995. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland, November.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *Proceedings of the Sixth Workshop on Very Large Corpora*, Montreal, Quebec, Canada, August.

Jerry R. Hobbs. 1976. Pronoun resolution. Technical Report 76-1, City College, New York.

Shalom Lappin and Herbert J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, pages 535–561.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1994. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ruslan Mitkov. 1997. Factors in anaphora resolution: They are not the only things that matter. A case study based on two different approaches. In *Proceedings of the ACL '97/EACL '97 Workshop on Operational Factors in Practical Robust Anaphora Resolution.*

Thomas Morton. 1999. Using coreference in question answering. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, November. NIST.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Part of Speech Tagger. In *Proceeding of the Conference on Empirical Methods in Natural Language Processing*, University of Pennsylvania, May 17-18.

Adwait Ratnaparkhi. 1997. A simple introduction to maximum entropy models for natural language processing. Technical Report 97-08, Institute for Research in Cognitive Science, University of Pennsylvania.

Jeffrey C. Reynar and Adwait Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19, Washington, D.C., April.

Jeff Reynar. 1998. *Topic Segmentation: Algorithms and Applications.* Ph.D. thesis, University of Pennsylvania.

Amit Singal, Steve Abney, Michiel Bacchiani, Michael Collins, Donald Hindle, and Fernando Pereira. 1999. AT&T at TREC-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, November. NIST.

Renata Vieira and Massimo Poesio. 1997. Processing definite descriptions in corpora. In *Corpus-based and Computational Approaches to Discourse Anaphora.* UCL Press.

Marc Vilain, John Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, Columbia, Maryland.

E. Voorhees and D. Tice. 1999. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Maryland, November. NIST.