Coreference as the Foundations for Link Analysis over Free Text Databases

Breck Baldwin
Institute for Research in Cognitive Science
University of Pennsylvania
3401 Walnut St. 400C
Philadelphia, PA 19104. USA
Phone: (215) 898-0329
Email: breck@linc.cis.upenn.edu

Amit Bagga
Box 90129
Dept. of Computer Science
Duke University
Durham, NC 27708-0129. USA
Phone: (919) 660-6507
Email: amit@cs.duke.edu

Abstract

Coreference annotated data has the potential to substantially increase the domain over which link analysis can be applied. We have developed coreference technologies which relate individuals and events within and across text documents. This in turn leverages the first step in mapping the information in those texts into a more data-base like format suitable for visualization with link driven software.

1 Introduction

Coreference is in some sense nature's own hyperlink. For example, the phrase 'Alan Turing', 'the father of modern computer science', or 'he' can refer to the same individual in the world. The communicative function of coreference is the ability to link information about entities across many sentences and documents. In data base terms, individual sentences provide entry records which are organized around entities, and the method of indicating which entity the record is about is coreference.

Link analysis is well suited to visualizing large structured databases where generalizations emerge from macro observations of relatedness. Unfortunately, free text is not sufficiently organized for similar fidelity observations. Coreference in its simplest form has the potential to organize free text sufficiently to greatly expand the domain over which link analysis can be fruitfully applied.

Below we will illustrate the kinds of coreference that we currently annotate in the CAMP software system and give an idea of our system performance. Then we will illustrate what kinds of observations could be pulled via visualization from coreference annotated document collections.

2 CAMP Natural Language Processing Software

The foundation of our system is the CAMP NLP system. This system provides an integrated environment in which one can access many levels of linguistic information as well as world knowledge. Its main components include: named entity recognition, tokenization, sentence detection, part-of-speech tagging, morphological analysis, parsing, argument detection, and coreference resolution as described below. Many of the techniques used for these tasks perform at or near the state of the art and are described in more depth in (Wacholder 97), (Collins 96), (Baldwin 95), (Reynar 97), (Baldwin 97), (Bagga, 98b).

3 Within Document Coreference

We have been developing the within document coreference component of CAMP since 1995 when the system was developed to participate in the Sixth Message Understanding Conference (MUC-6) coreference task. Below we will illustrate the classes of coreference that the system annotates.

Coreference breaks down into several readily identified areas based on the form of the phrase being resolved and the method of calculating coreference. We will proceed in the approximate ordering of the systems execution of components. A more detailed analysis of the classes of coreference can be found in (Bagga, 98a).

3.1 Highly Syntactic Coreference

There are several readily identified syntactic constructions that reliably indicate coreference. First are appositive relations as holds between 'John Smith' and 'chairman of General Electric' in:

> John Smith, chairman of General Electric, resigned yesterday.

Identifying this class of coreference requires some syntactic knowledge of the text and property analysis of the individual phrases to avoid finding coreference in examples like:

> John Smith, 47, resigned yesterday.
> Smith, Jones, Woodhouse and Fife announced a new partner.

To avoid these sorts of errors we have a mutual exclusion test that applies to such positings of coreference to prevent non-sensical annotations.

Another class of highly syntactic coreference exists in the form of predicate nominal constructions as

between 'John' and 'the finest juggler in the world' in:

> John is the finest juggler in the world.

Like the appositive case, mutual exclusion tests are required to prevent incorrect resolutions as in:

> John is tall.
>
> They are blue.

These classes of highly syntactic coreference can play a very important role in bridging phrases that we would normally be unable to relate. For example, it is unlikely that our software would be able to relate the same noun phrases in a text like

> The finest juggler in the world visited Philadelphia this week. John Smith pleased crowds every night in the Annenberg theater.

This is because we do not have sufficiently sophisticated knowledge sources to determine that jugglers are very likely to be in the business of pleasing crowds. But the recognition of the predicate nominal will allow us to connect a chain of 'John Smith', 'Mr. Smith', 'he' with a chain of 'the finest juggler in the world', 'the juggler' and 'a juggling expert'.

3.2 Proper Noun Coreference

Names of people, places, products and companies are referred to in many different variations. In journalistic prose there will be a full name of an entity, and throughout the rest of the article there will be ellided references to the same entity. Some name variations are:

- Mr. James Dabah <- James <- Jim <- Dabah
- Minnesota Mining and Manufacturing <- 3M Corp. <- 3M
- Washington D.C. <- WASHINGTON <- Washington <- D.C. <- Wash.
- New York <- New York City <- NYC <- N.Y.C.

This class of coreference forms a solid foundation over which we resolve the remaining coreference in the document. One reason for this is that we learn important properties about the phrases in virtue of the coreference resolution. For example, we may not know whether 'Dabah' is a person name, male name, female name, company or place, but upon resolution with 'Mr. James Dabah' we then know that it refers to a male person.

We resolve such coreferences with partial string matching subroutines coupled with lists of honorifics, corporate designators and acronyms. A substantial problem in resolving these names is avoiding overgeneration like relating 'Washington' the place with the name 'Consuela Washington'. We control the string matching with a range of salience functions and restrictions of the kinds of partial string matches we are willing to tolerate.

3.3 Common Noun Coreference

A very challenging area of coreference annotation involves coreference between common nouns like 'a shady stock deal' and 'the deal'. Fundamentally the problem is that very conservative approaches to exact and partial string matches overgenerate badly. Some examples of actual chains are:

- his dad's trophies <- those trophies
- those words <- the last words
- the risk <- the potential risk
- its accident investigation <- the investigation

We have adopted a range of matching heuristics and salience strategies to try and recognize a small, but accurate, subset of these coreferences.

3.4 Pronoun Coreference

The pronominal resolution component of the system is perhaps the most advanced of all the components. It features a sophisticated salience model designed to produce high accuracy coreference in highly ambiguous texts. It is capable of noticing ambiguity in text, and will fail to resolve pronouns in such circumstances. For example the system will not resolve 'he' in the following example:

> Earl and Ted were working together when suddenly he fell into the threshing machine.

We resolve pronouns like 'they', 'it', 'he', 'hers', 'themselves' to proper nouns, common nouns and other pronouns. Depending on the genre of data being processed, this component can resolve 60-90% of the pronouns in a text with very high accuracy.

3.5 The Overall Nexus of Coreference in a Document

Once all the coreference in a document has been computed, we have a good approximation of which sentences are strongly related to other sentences in the document by counting the number of coreference links between the sentences. We know which entities are mentioned most often, and what other entities are involved in the same sentences or paragraphs. This sort of information has been used to generate very effective summaries of documents and as a foundation for a simple visualization interface to texts.

4 Cross Document Coreference

Cross-document coreference occurs when the same person, place, event, or concept is discussed in more than one text source. Figure 1 shows the architecture of the cross-document module of CAMP.

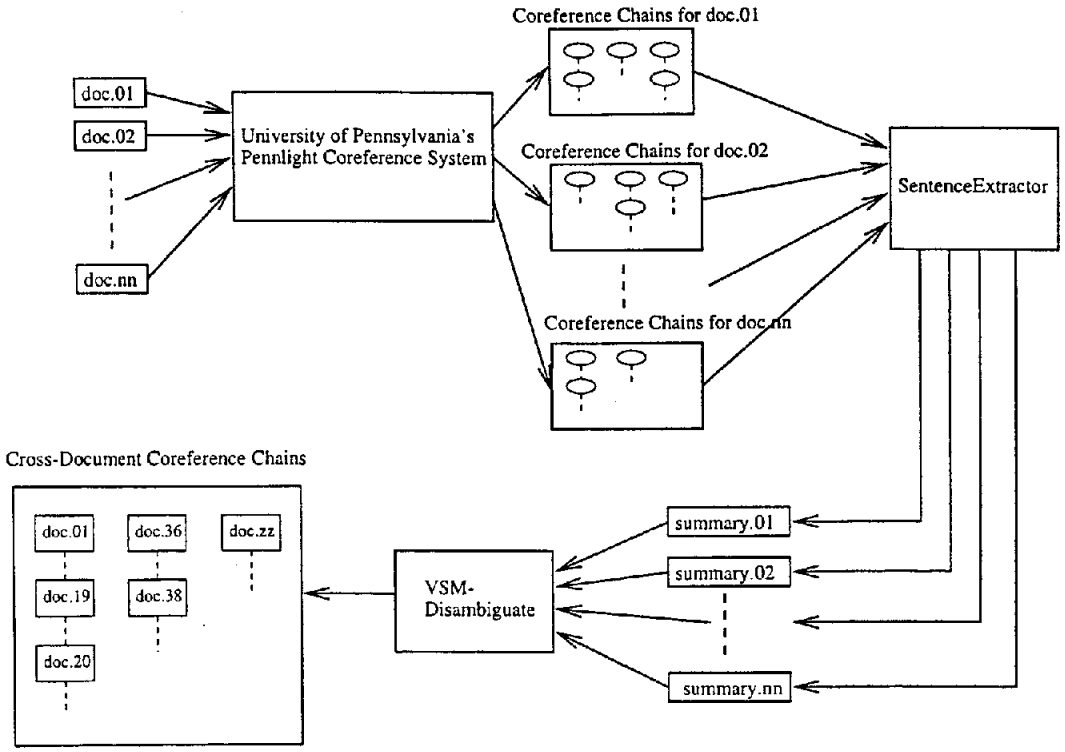

Figure 1: Architecture of the Cross-Document Coreference System

John Perry, of Weston Golf Club, announced his resignation yesterday. He was the President of the Massachusetts Golf Association. During his two years in office, Perry guided the MGA into a closer relationship with the Women's Golf Association of Massachusetts.

Figure 2: Extract from doc.36

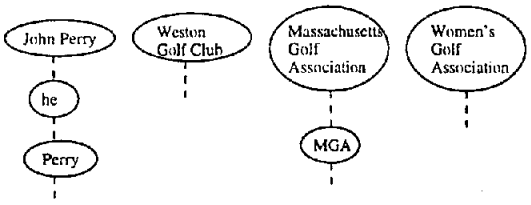

Figure 3: Coreference Chains for doc.36

This module takes as input the coreference chains produced by CAMP's within document coreference module. Details about each of the main steps of the cross-document coreference algorithm are given below.

- First, for each article, the within document coreference module of CAMP is run on that article. It produces coreference chains for all the entities mentioned in the article. For example, consider the two extracts in Figures 2 and 4. The coreference chains output by CAMP for the two extracts are shown in Figures 3 and 5.

- Next, for the coreference chain of interest within each article (for example, the coreference chain that contains "John Perry"), the Sentence Extractor module extracts all the sentences that contain the noun phrases which form the coreference chain. In other words, the SentenceExtractor module produces a "summary" of the article with respect to the entity of interest. These summaries are a special case of the query sensitive techniques being developed at Penn using CAMP. Therefore, for doc.36 (Figure 2), since at least one of the three noun phrases ("John Perry," "he," and "Perry") in the coreference chain of interest appears in each of the three sentences in the extract, the summary produced by SentenceExtractor is the extract itself. On the other hand, the summary produced by SentenceExtractor for the coreference chain of interest in doc.38 is only the first sentence of the extract because the only element of the coreference chain appears in this sentence.

- Finally, for each article, the VSM-Disambiguate module uses the summary extracted by the SentenceExtractor and computes its similarity with

Oliver "Biff" Kelly of Weymouth succeeds John Perry as president of the Massachusetts Golf Association. "We will have continued growth in the future," said Kelly, who will serve for two years. "There's been a lot of changes and there will be continued changes as we head into the year 2000."

Figure 4: Extract from doc.38

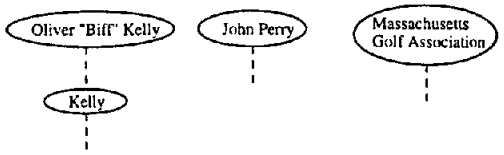

Figure 5: Coreference Chains for doc.38

the summaries extracted from each of the other articles. The VSM-Disambiguate module uses a standard vector space model (used widely in information retrieval) (Salton, 89) to compute the similarities between the summaries. Summaries having similarity above a certain threshold are considered to be regarding the same entity.

4.1 Experiments and Results

We tested our cross-document system on two highly ambiguous test sets. The first set contained 197 articles from the 1996 and 1997 editions of the New York Times, while the second set contained 219 articles from the 1997 edition of the New York Times. The sole criteria for including an article in the two sets was the presence of a string matching the "/John.*?Smith/", and the "/resign/" regular expressions respectively.

The goal for the first set was to identify cross-document coreference chains about the same *John Smith*, and the goal for the second set was to identify cross-document coreference chains about the same "resign" event. The answer keys were manually created, but the scoring was completely automated.

There were 35 different *John Smiths* in the first set. Of these, 24 were involved in chains of size 1. The other 173 articles were regarding the 11 remaining *John Smiths*. Descriptions of a few of the *John Smiths* are: Chairman and CEO of General Motors, assistant track coach at UCLA, the legendary explorer, and the main character in Disney's Pocahontas, former president of the Labor Party of Britain. In the second set, there were 97 different "resign" events. Of these, 60 were involved in chains of size 1. The articles were regarding resignations of several different people including Ted Hobart of ABC Corp., Dick Morris, Speaker Jim Wright, and the possible resignation of Newt Gingrich.

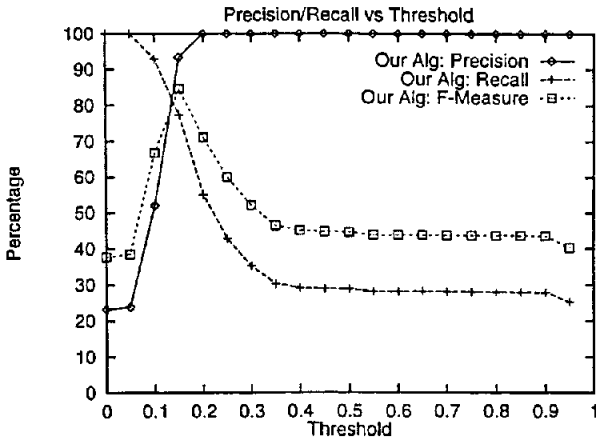

Figure 7: Precision, Recall, and F-Measure Using Our Algorithm for the *John Smith* Data Set

4.2 Scoring and Results

In order to score the cross-document coreference chains output by the system, we had to map the cross-document coreference scoring problem to a within-document coreference scoring problem. This was done by creating a meta document consisting of the file names of each of the documents that the system was run on. Assuming that each of the documents in the two data sets was about a single *John Smith*, or about a single "resign" event, the cross-document coreference chains produced by the system could now be evaluated by scoring the corresponding within-document coreference chains in the meta document.

Precision and recall are the measures used to evaluate the chains output by the system. For an entity, i, we define the precision and recall with respect to that entity in Figure 6.

The final precision and recall numbers are computed by the following two formulae:

$$\text{Final Precision} = \sum_{i=1}^{N} w_i * Precision_i$$

$$\text{Final Recall} = \sum_{i=1}^{N} w_i * Recall_i$$

where N is the number of entities in the document, and w_i is the weight assigned to entity i in the document. For the results discussed in this paper, equal weights were assigned to each entity in the meta document. In other words, $w_i = \frac{1}{N}$ for all i. Full details about the scoring algorithm can be found in (Bagga, 98).

Figure 7 shows the Precision, Recall, and the F-Measure (the average of precision and recall with equal weights for both) statistics for the *John Smith* data set. The best precision and recall achieved by

$$\text{Precision}_i = \frac{\textit{number of correct elements in the output chain containing entity}_i}{\textit{number of elements in the output chain containing entity}_i}$$

$$\text{Recall}_i = \frac{\textit{number of correct elements in the output chain containing entity}_i}{\textit{number of elements in the truth chain containing entity}_i}$$

Figure 6: Definitions for Precision and Recall for an Entity i

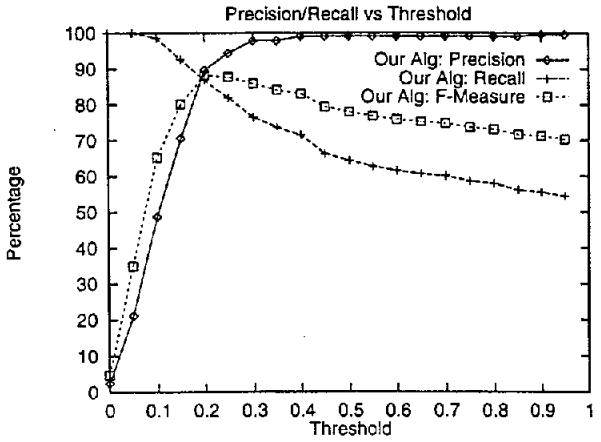

Figure 8: Precision, Recall, and F-Measure Using Our Algorithm for the "resign" Data Set

the system on this data set was 93% and 77% respectively (when the threshold for the vector space model was set to 0.15). Similarly, Figure 8 shows the same three statistics for the "resign" data set. The best precision and recall achieved by the system on this data set was 94% and 81% respectively. This occurs when the threshold for the vector space model was set to 0.2. The results show that the system was very successful in resolving cross-document coreference.

5 Possible Generalizations About Large Data Collections Derived From Coreference Annotations

Crucial to the entire process of visualizing large document collections is relating the same individual or event across multiple documents. This single aspect of our system establishes its viability for large collection analysis. It allows the drops of information held in each document to be merged into a larger pool that is well organized.

5.1 The Primary Display of Information

Two display techniques immediately suggest themselves for accessing the coreference annotations in a document collection, the first is to take the identified entities as atomic and link them to other entities which co-occur in the same document. This might reveal a relation between individuals and events, or individuals and other individuals. For example, such a linking might indicate that no newspaper article ever mentioned both Clark Kent and Superman in the same article, but that most all other famous individuals tended to overlap in some article or another. On the positive case, individuals, over time, may tend to congregate in media stories or events may tend to be more tightly linked than otherwise expected.

The second technique would be to take as atomic the documents and relate via links other documents that contain mention of the same entity. With a temporal dimension, the role of individuals and events could be assessed as time moved forward.

5.2 Finer Grained Analysis of the Documents

The fact that two entities coexisted in the same sentence in a document is noteworthy for correlational analysis. Links could be restricted to those between entities that co-existed in the same sentence or paragraph. Additional filterings are possible with constraints on the sorts of verbs that exist in the sentence.

A more sophisticated version of the above is to access the argument structure of the document. CAMP software provides a limited predicate argument structure that allows subjects/verbs/objects to be identified. This ability moves our annotation closer to the fixed record data structure of a traditional data base. One could select an event and its object, for instance 'X sold arms to Iraq' and see what the fillers for X were in a link analysis. There are limitations to predicate argument structure matching–for instance getting the correct pattern for all the selling of arms variations is quite difficult.

In any case, there appear to be a myriad of applications for link analysis in the domain of large text data bases.

6 Conclusions

The goal of this paper has been to articulate a novel input class for link based visualization techniques–coreference. We feel that there is tremendous potential for collaboration between researchers in visualization and in coreference annotation given the new

space of information provided by coreference analysis.

7 Acknowledgments

The second author was supported in part by a Fellowship from IBM Corporation, and in part by the University of Pennsylvania. Part of this work was done when the second author was visiting the Institute for Research in Cognitive Science at the University of Pennsylvania.

References

Bagga, Amit, and Breck Baldwin. Algorithms for Scoring Coreference Chains. *Proceedings of the the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation*, May 1998.

Bagga, Amit. Evaluation of Coreferences and Coreference Resolution Systems. *Proceedings of the First Language Resource and Evaluation Conference*, pp. 563-566, May 1998.

Bagga, Amit, and Breck Baldwin. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. To appear at the *17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics* (COLING-ACL'98), August 1998.

Baldwin, Breck. CogNIAC: A Discourse Processing Engine. University of Pennsylvania Department of Computer and Information Sciences Ph.D. Thesis, 1995.

Baldwin, B., C. Doran, J. Reynar, M. Niv, and M. Wasson. EAGLE: An Extensible Architecture for General Linguistic Engineering. *Proceedings RIAO, Computer-Assisted Information Searching on Internet*, Montreal, Canada, 1997.

Collins, Michael. A New Statistical Parser Based on Bigram Lexical Dependencies. *Proceedings of the* 34^{th} *Meeting of the Association for Computational Linguistics*, 1996.

Ratnaparkhi, Adwait. A Maximum Entropy Model for Part-Of-Speech Tagging. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 133–142, May 1996.

Wacholder, Nina, Yael Ravin, and Misook Choi. Disambiguation of Proper Names in Text. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 202-208, 1997.

Reynar, Jeffrey, and Adwait Ratnaparkhi. A Maximum Entropy Approach to Identifying Sentence Boundaries. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pp. 16-19, 1997.

Salton, Gerard. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, 1989, Reading, MA: Addison-Wesley.