

Using Coreference in Question Answering

Thomas S. Morton

Department of Computer and Information Science

University of Pennsylvania

`tsmorton@linc.cis.upenn.edu`

1 Introduction

In this paper we present an overview of the system used by the GE/Penn team for the the Question Answering Track of TREC-8. Our system uses a simple sentence ranking method which is enhanced by the addition of coreference annotated data as its input. We will present an overview of our initial system and its components. Since this was the first time this track has been run, we made numerous additions to our initial system. We will describe these additions and what motivated them as a series of lessons learned after which the final system used for our submission will be described. Finally we will discuss directions for future research.

2 Initial System Overview

The input to our system is a small set of candidate documents and a query. To get a set of candidate documents we employed a search engine over the TREC-8 document collection and further processed the top 20 documents returned by it for each query. In order for our system to annotate coreference relations, a variety of linguistic annotation is required. This includes accounting for SGML tags in the original documents, performing sentence detection, tokenization, noun phrase detection, and named-entity categorization. With this annotation complete, the coreference system annotates coreference relations between noun phrases. The coreference-annotated document is then passed to a sentence ranker which ranks each of the sentences, merging these ranked sentences with the sentences from previously processed documents. Finally the top 5 sentences are presented to the user.

2.1 Search Engine

In order to get a small collection of candidate documents, we installed and indexed the TREC-8 data set with the PRISE 2.0 search engine, developed by NIST. Indexing and retrieval were done using the default configuration with no attempts made to tune the ranking to the Question Answering task. From this we took the top 20 ranked documents and performed further processing on each of them.

2.2 Preprocessing

Determining coreference between noun phrases requires that the noun phrases in the text have been identified. This processing begins by preprocessing the SGML to determine likely boundaries between segments of text, sentence-detecting these segments using a sentence detector described in (Reynar and Ratnaparkhi, 1997), and tokenizing those sentences using a tokenizer described in (Reynar, 1998). The text can then be part-of-speech-tagged using the tagger described in (Ratnaparkhi, 1996), and finally noun phrases are determined using a maximum entropy model trained on the Penn Treebank (Marcus et al., 1994). The output of Nymble (Bikel et al., 1997), a named-entity recognizer which determines which words are people's names, organizations, locations, etc., is also used to aid in determining coreference relationships.

2.3 Coreference

Once preprocessing is completed, the system iterates through each of the noun phrases to determine if it refers to a noun phrase which has occurred previously. Only proper noun phrases, definite noun phrases, and non-possessive third person pronouns are considered. Proper noun phrases are determined by the part of speech assigned to the last word in the noun phrase. A proper noun phrase is considered coreferent with a previously occurring noun phrase if it is a substring of that noun phrase, excluding abbreviations and words which are not proper nouns. A noun phrase is considered definite if it begins with the determiner "the" or begins with a possessive pronoun or a past-participle verb. A definite noun phrase is considered coreferent with another noun phrase if the last word in the noun phrase matches the last word in a previously occurring noun phrase. The mechanism for resolving pronouns consists of a maximum entropy model which examines two noun phrases and produces a probability that they corefer. The 20 previously occurring noun phrases are considered as possible referents. The possibility that the pronoun refers to none of these noun phrases is also examined. The pair with the highest probability are considered coreferent, or the pronoun is left

unresolved when the model predicts this as the most likely outcome. The model considers the following features:

1. The category of the noun phrase being considered as determined by the named-entity recognizer.
2. The number of noun phrases that occur between the candidate noun phrase and the pronoun.
3. The number of sentences that occur between the candidate noun phrase and the pronoun.
4. Which noun phrase in a sentence is being referred to (first, second, ...).
5. In which noun phrase in a sentence the pronoun occurred (first, second, ...).
6. The pronoun being considered.
7. If the pronoun and the noun phrase are compatible in number.
8. If the candidate noun phrase is another pronoun, is it compatible with the referring pronoun?
9. If the candidate noun phrase is another pronoun, is it the same as the referring pronoun?

The model is trained on nearly 1200 annotated examples of pronouns which refer to or fail to refer to previously occurring noun phrases.

2.4 Sentence Ranking

Sentences are ranked based on the sum of the *idf* weights (Salton, 1989) for each unique term which occurs in the sentence and also occurs in the query. The *idf* weights are computed based on the documents found on TREC discs 4 and 5 (Voorhees and Harman, 1997). No additional score is given for tokens occurring more than once in a sentence. If a sentence contains a coreferential noun phrase then the terms contained in any of the noun phrases with which it is coreferent are also considered to be contained in the sentence.

A secondary weight was also used to resolve ties in the first weight ranking and to determine how sentences longer than 250 bytes should be truncated. The secondary weight was computed for each noun phrase based on the sum of the *idf* weights for each unique term where words occurring farther away from the noun phrase were discounted. This was done by adding the product of the *idf* weight for a word and the reciprocal of the distance, in words, between the noun phrase and the word. For example, a word three tokens to the left of a noun phrase would only receive a third of its *idf* weight with respect to that noun phrase. This weight was used to select a “most central” noun phrase and the weight of this noun phrase was used to resolve ties between

sentences equally ranked by the first score. In cases where a sentence was longer than 250 bytes, this noun phrase was used to determine where the sentence would be truncated.

3 Lessons Learned

3.1 Lesson 1

Our first goal was to develop a baseline with which we could compare our system’s output. The simplest baseline we could imagine would be to simply rank segments of text based on the common $tf \cdot idf$ measure. Since these segments were small, having a maximum of 250 bytes, we ignored the term frequency component, and the query and segment were treated as a set of terms rather than a bag. Each segment was ranked based on the sum of the *idf* weights for the words in that segment which, once stemmed, matched those found in the query. Each segment was a 250-byte window centered on a term which was also found in the query. On the development set provided, this produced an answer in the top five sentences for nearly half (17/38) of the questions provided for development. This allowed us to better assess the added value various types of linguistic annotation would provide.

3.2 Lesson 2

Performing linguistic annotation of the documents in the collection is computationally expensive. While we only examined the top 20 returned documents, some of these documents were very long, often exceeding 2MB. To combat this, each document was reduced to a 20K segment using the 250-byte segment ranked first by the baseline as the center of the 20K segment. This sped up processing considerably but had no noticeable effect on system output. This may be because some question generation was based in part on reading the documents and creating questions which were answered by that document. This may have lead to a bias for shorter documents.

3.3 Lesson 3

Many questions indicate a semantic category that the answer should fall in based on the Wh-word the sentence uses. For Wh-words such as “Who”, “Where”, and “When”, a fairly specific category is specified while the category for “What” and “Which” is usually specified by the noun phrase following it. “How” can be used to specify a variety of types; however, when it is followed by words such as “many”, “long”, “fast”, the answer will likely contain a number of some sort. When the semantic type of the answer could be determined, and this could be mapped to a category that was determined by the named-entity recognizer or some other recognizable pattern, then only sentences which evoked an entity of the same category were considered. This

included sentences which contained pronouns which referred to entities of the correct type in preceding sentences. Sentences which contained the correct entity type, but all entities of this type were present in the query, were also ignored. This processing helped exclude sentences which only used terms in the query and would be highly ranked even though they did not contain a possible answer.

3.4 Lesson 4

The semantic category for questions which ask for a date can usually be determined. These are also categories that the named-entity recognizer identified and so the system was quite effective at finding candidate answers to these sorts of questions. However, the form of the answer often did not meet the needs of the user. Within the context of a newspaper article, relative date terms such as today, Tuesday, last week, or next month, can be interpreted by a reader based on context; however, when this context is removed, the meaning of these terms is often unclear. All the articles in this collection contain datelines which often make it possible to automatically resolve such terms for the user. For these terms we used the dateline as a base reference for when the article was written and then used a small set of heuristics to determine a complete description of a date term. Additional terms introduced by the more complete description of the relative date term were also considered to be in that sentence. This was especially helpful when these terms were in the query and would not have matched this sentence without such processing. This processing was also helpful when presenting sentences to the user. When sentences contained relative date terms, the parts of the description which were not present in the relative date term were inserted after it to improve the user understanding of the text without context.

3.5 Lesson 5

While linguistic processing is helpful in determining answers to a variety of questions, some information needs can be satisfied with much simpler means. For questions asking “Where is X” or “What is the capital of X”, a good online dictionary will usually provide the answer within a few keystrokes. For these two types of questions, we automatically extracted a set of probable answers and added these to the query. This improved system performance and did a better job of addressing the user’s intentions than the system without this information. Specifically, the dictionary provided answers with a better level of generalization than the system did without these additional query terms.

4 Final System

Our final system examined the query and added terms from an online dictionary when applicable.

This expanded query was then passed to the search engine, and the top 20 documents returned by it were collected for further processing. The baseline system was run on these documents to find a central passage, and a 20K window around this passage was kept for further processing. Preprocessing was performed on these segments and coreference relations between entities and dates were automatically annotated. Finally, sentences which weren’t excluded by the semantic-category filtering were ranked using the simple *idf* weighting described above. The top-ranked sentences were augmented to include complete descriptions of coreferential terms such as definite noun phrases, proper nouns, pronouns, and dates, which were not already present in the sentence. These augmented sentences were then presented to the user.

5 Results

The part of the TREC-8 Question Answering Track evaluation in which we participated allowed 5 answers to be submitted, each of which could be at most 250 bytes long. For the 198 questions in the evaluation, our system was able to answer 126 of them, or 63.3%. If answers are weighted by rank, our mean reciprocal rank was 0.510. This compared favorably with other systems; of the 20 participants, our system ranked 4th overall.

6 Discussion and Future Work

Attending TREC-8 provided us with additional insights for future work. The most significant of these is that in the future more attention needs to be paid to indexing. Specifically, we discovered that the search engine we used, PRISE 2.0, was significantly below the state-of-the-art in performance at the ad-hoc task. To compare its performance at the Question Answering task, we considered all the documents in which some participant had found a correct answer. This is likely not the complete set of documents which contain the answer, but it serves as a reasonable approximation. We then compared the number of these documents that PRISE found compared to AT&T’s search engine. The result is that the AT&T search engine returned 146 more documents, over all queries, in which some system found the answer than the PRISE search engine. For 38 queries, the PRISE system returned no documents in which an answer was found, while the AT&T system did this for only 35 documents. We should also explore the possibility of examining more than 20 documents. This is evidenced by the fact that if all 200 documents returned by the AT&T system are considered, then a document containing an answer was provided for at least 187 of the 198 queries. In a similar vein, we also hope to look at alternate indexing schemes such as paragraph indexing, which was

used in Southern Methodist University's system.

7 Conclusion

Here we present a system for performing question answering on a large collection of text. This system uses a simple ranking method, which is aided by determining coreference relations to add terms to a sentence and by determining the semantic category of the answer to exclude some sentences from consideration. We believe coreference plays an important role in question answering, as it allows a system to extract answers from text which refers to but doesn't explicitly mention an entity. It also provides a means to make text presented to the user without its original context easier to understand. Determining the semantic category that the answer will be in, and the entities which fall into that category, is also useful: it allows sentences which do not contain a possible answer to be excluded from consideration. This system performed well at the evaluation and we look forward to improving its performance for future evaluations.

References

- D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*.
- M. Marcus, B. Santorini, and M. Marcinkiewicz. 1994. Building a large annotated corpus of english: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.
- Adwait Ratnaparkhi. 1996. A Maximum Entropy Part of Speech Tagger. In Eric Brill and Kenneth Church, editors, *Conference on Empirical Methods in Natural Language Processing*, University of Pennsylvania, May 17-18.
- 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.
- Gerald Salton. 1989. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Publishing Company, Inc.
- Ellen M. Voorhees and Donna Harman. 1997. Overview of the fifth Text REtrieval Conference (TREC-5). In *Proceedings of the Fifth Text REtrieval Conference (TREC-5)*, pages 1–28. NIST 500-238.