Drexel at TREC 2007: Question Answering

Protima Banerjee
College of Information Science
and Technology
Drexel University
Philadelphia, PA 19104 USA
protima.banerjee@drexel.edu

Hyoil Han
College of Information Science
and Technology
Drexel University
Philadelphia, PA 19104 USA
hyoil.han@acm.org

ABSTRACT
The TREC Question Answering Track presented several distinct challenges to participants in 2007. Participants were asked to create a system which discovers the answers to factoid and list questions about people, entities, organizations and events, given both blog and newswire text data sources. In addition, participants were asked to expose interesting information nuggets which exist in the data collection, which were not uncovered by the factoid or list questions. This year is the first time the Intelligent Information Processing group at Drexel has participated in the TREC Question Answering Track. As such, our goal was the development of a Question Answering system framework to which future enhancements could be made, and the construction of simple components to populate the framework. The results of our system this year were not significant; our primary accomplishment was the establishment of a baseline system which can be improved upon in 2008 and going forward.

1. INTRODUCTION

The TREC Question Answering Track presented several distinct challenges to participants in 2007. Participants were asked to create a system which discovers the answers to factoid and list questions, given both blog and newswire data sources. The targets for each question series could be people, entities, events or organizations, and the questions themselves varied in the types of information required to construct a correct response. Each question set ended with an "Other" question; the answer to the "Other" question is an interesting information nugget which was not previously uncovered by any of the other questions in the series. Each group was permitted to submit three runs, and no manual intervention was allowed for any results submitted. All participating systems were required to freeze their code and configuration once the 2007 question set had been downloaded from the TREC Question Answering website.

This paper describes the first time participation of Drexel University's Intelligent Information Processing group in the TREC Question Answering Track. Our primary goal this year was the development of a Question Answering system framework to which future enhancements can be applied. As a result, our results this year were not significant; our primary accomplishment was the establishment of a baseline system which can be improved upon in 2008 and going forward.

2. BACKGROUND

Hirschman [1] describes a generic architecture for a Question Answering system, which is depicted (with slight variation) in Figure 1. According to Hirschman, five core steps exist in the QA process. They are:

1. Question Analysis, during which a question is interpreted and decomposed.
2. Document Collection Pre-processing, during which stemming, part of speech

tagging, passage identification, entity extraction and indexing may occur.

3. Candidate Document Selection, during which a set of documents which is likely to include the candidate answer are selected from the processed document corpus.
4. Answer Extraction, during which a set of candidate answers are extracted from the candidate documents.
5. Response Generation, during which the set of candidate answers are reduced to a single system response.

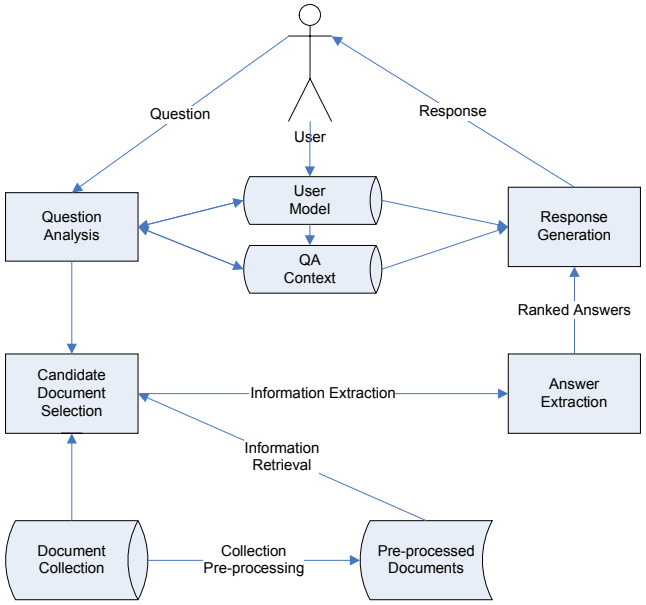

Figure 1: Hirschman's Architecture for a Generic QA System [1]

This architecture can be thought of as a two-stage model, represented in Figure 2. Here, the first stage is an Information Retrieval stage, which outputs a set of candidate (answer-bearing) documents. The second stage is then primarily an Information Extraction stage, in which answers are mined from the candidate documents. The final step in the second stage of the Question Answering process is then to formulate a response which can be returned back to the user.

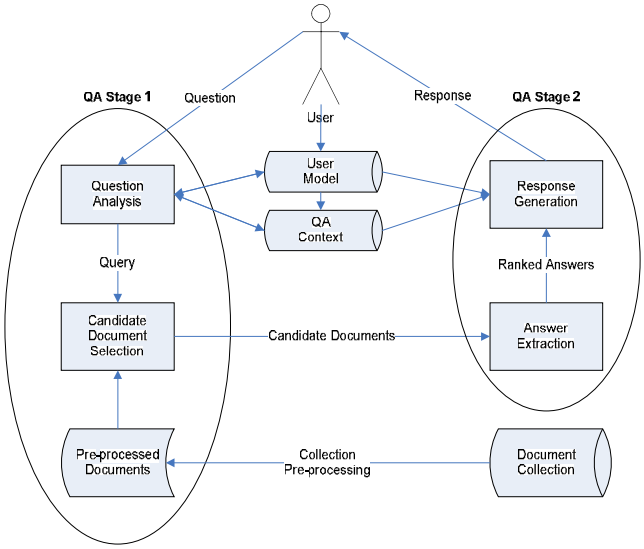

Figure 2: Two-stage Model for a Generic QA Architecture

The remainder of this paper will describe our Question Answering architecture and framework in the context of this two-stage model.

3. SYSTEM ARCHITECTURE

In this section we will describe the components of our system architecture. We will explain these components in a generic sense, as well as their specific implementations in our system this year. Our system architecture is shown in Figure 3 below.

As previously stated, the first stage of our architecture is primarily an Information Retrieval (IR) stage. In this stage, the AQUAINT2 documents were pre-processed using stop-word removal, stemming, and indexing using tools provided as a part of the LEMUR toolkit [2] (http://www.lemurproject.org). Once the corpus pre-processing was completed, we used the Indri [3] search engine to perform document retrieval. Indri queries were created from the TREC questions and targets using Indri's query-likelihood retrieval method which is based on the Ponte and Croft language modeling approach [4] using Dirichlet priors for smoothing [5]. The top documents returned from Indri were then considered to be the set of Candidate (answer-bearing) Documents. These documents were then sent on to the second Question Answering stage for further processing. We considered Candidate Documents sets in sizes of 5, 10 and

25 documents for the three runs submitted this year.

It should be noted here that due to time and budgetary constraints we utilized only the AQUAINT2 corpus for our experiments. No Blog data was included as a part of our submission this year.

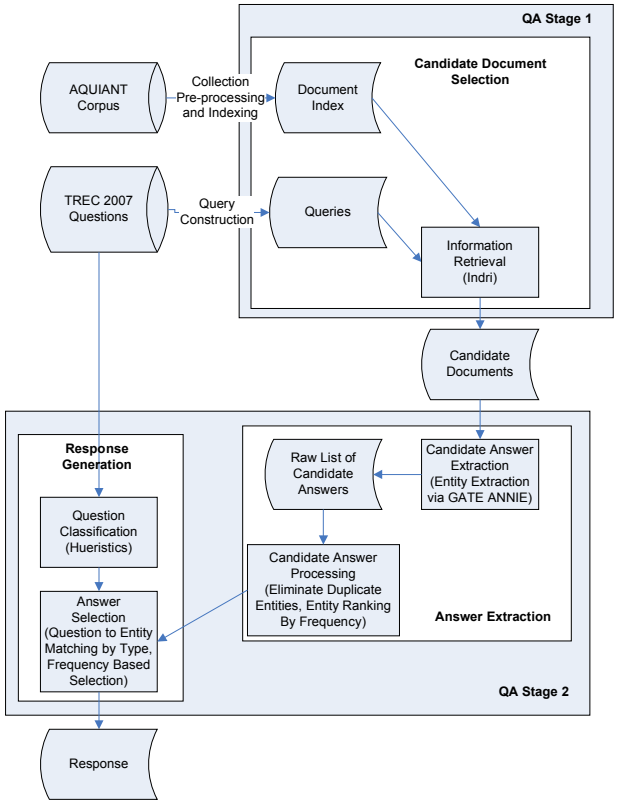

Figure 3: System Architecture

The set of Candidate Documents were sent to a Candidate Answer Extraction module. Our current implementation of this module made use of the Information Extraction (IE) resources provided by the GATE ANNIE Toolkit [6] (http://gate.ac.uk/ie/annie.html). ANNIE was used to extract all entities of type "Person," "Location," "Organization," "Date," "Currency," and "Percentage" from the Candidate Documents. These entities, which we considered in a generic sense to be the Raw List of Candidate Answers, were stored in an Oracle database for further processing.

We then performed two types of basic processing on the Raw List of Candidate Answers. First, we eliminated duplicates using simple string matching functions provided by Oracle. Then, we grouped the answers by entity type (eg. "Person," "Location," "Organization") and ranked the entities according to their frequency of occurrence in the set of Candidate Documents.

At this point, we returned to the original query to try to understand what type of entity would best answer the question that was posed. We created a simple set of heuristics based on the existence of keywords within the query such as "who," "what," "when," "where," "why," "how many" and "how much." Having reviewed the query classification scheme proposed by Lehnert [7], we recognize that this is an area where our approach can use significant improvement. The output of our simple classification scheme was an Answer Type, which is represents the entity type that will correctly answer the question.

Knowing the Answer Type and the set of Candidate Answers allowed us to implement a simple matching algorithm. Our final response to the question was the top ranked Candidate Answer of the proper Answer Type. We then went back to our Raw Answer List to determine the document from which the top-ranked Candidate Answer came. Our final response to the question consisted of the top-ranked Candidate Answer and its associated Document Number.

The above approach was designed primarily with factoid questions in mind. We were able to re-use the same approach for list questions. In the case of list questions, however, we selected a set Candidate Answers to be returned as the query response. We selected this set by considering answers which ranked above a fixed frequency threshold. Our approach does not yet incorporate a mechanism to respond to "Other" questions. We are currently evaluating a methodology for modeling Question Answering Context that will

solidify our approach to answering “Other” questions in 2008.

4. RESULTS

As might be expected, our results this year were not significant. We submitted three runs, using Candidate Document sets of 5, 10 and 25 documents, and achieved accuracy results of 0.016, 0.019, and 0.022 for these runs respectively. The accuracy results of our system were thus directly influenced by the quantity of data returned by the IR stage of our system. However, our results are far below the accuracy results for state-of-the-art systems. We expect significant improvements in these results in 2008 as we continue to experiment and enhance the components within our framework.

5. CONCLUSION

In conclusion, our primary goal this year was to establish a framework for our future work in Question Answering. We started with a holistic approach to the Question Answering process, using the generic Question Answering architecture proposed by Hirschman [1] as a foundation. Over the next year, our work in Question Answering will focus on several key areas: development and management of a Question Answering Context using statistical language modeling methods, development of robust algorithms for Candidate Answer Selection and Response Generation, and the incorporation of knowledge sources into the Question Answering process. We plan to participate in the 2008 TREC Question Answering Track to evaluate our research against other systems and approaches.

6. REFERENCES

[1] L. Hirschman and R. Gaizauskas, "Natural language question answering: the view from here." vol. 7: Cambridge Univ Press, 2002, pp. 275-300.

[2] P. Ogilvie and J. P. Callan, "Experiments Using the Lemur Toolkit," Text REtrieval Conference, 2001.

[3] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri: A language model-based search engine for complex queries," Proceedings of International Conference on New Methods in Intelligence Analysis, 2004.

[4] J. M. Ponte and W. B. Croft, A language modeling approach to information retrieval: ACM Press New York, NY, USA, 1998.

[5] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," ACM Transactions on Information Systems (TOIS), vol. 22, pp. 179-214, 2004.

[6] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham, "Evolving GATE to meet new challenges in language engineering." vol. 10: Cambridge Univ Press, 2004, pp. 349-373.

[7] W. Lehnert, The Process of Question Answering: Lawrence Erlbaum Associates, 1977.