University of Lethbridge's Participation in TREC-2004 QA Track

Yllias Chali Stephen Dubien
Department of Mathematics and Computer Science
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, Canada, T1K 3M4
E-mail: {chali,dubiens}@cs.uleth.ca

1 Introduction

Text Retrieval Conference (TREC), organised by National Institute for Standards and Technology (NIST), is a set of tracks that represent different areas of text retrieval. These tracks provide a way to measure systems progress in certain fields in text retrieval such as cross-language retrieval, retrieval filtering and genomics. We participated in the question answering track.

The questions in the TREC-2004 QA track are clustered by target, which is the topic of the question. The QA track for 2004 has three types of questions: factoid questions that require only one correct response, list questions that require a non redundant list of correct responses and other questions that require a non redundant list of facts about the target that has not already been discovered by a previous answer. These questions will be answered using the AQUAINT collection, a collection of over a million newswire documents.

2 System Overview

The global architecture of our system is shown in Figure 1. Each of the modules in the system will be described below.

2.1 Topic Handler

This module is the driver for the system that handles the question file that is given for this track. This module extracts from the question file the target and the set of questions that corresponds to that topic. The target is then passed to the passage retrieval module.

The text of the question and question type (e.g. list, factoid or other) are then given to the question information extractor. For redundancy checking for the other questions, after each question is answered, the answers collected by this module and a list of all the answers for a topic is passed to the question information extractor.

2.2 Passage Retrieval

The AQUAINT collection is quite big and processing each document at runtime would be time consuming. An information retrieval system can be used to index the documents and

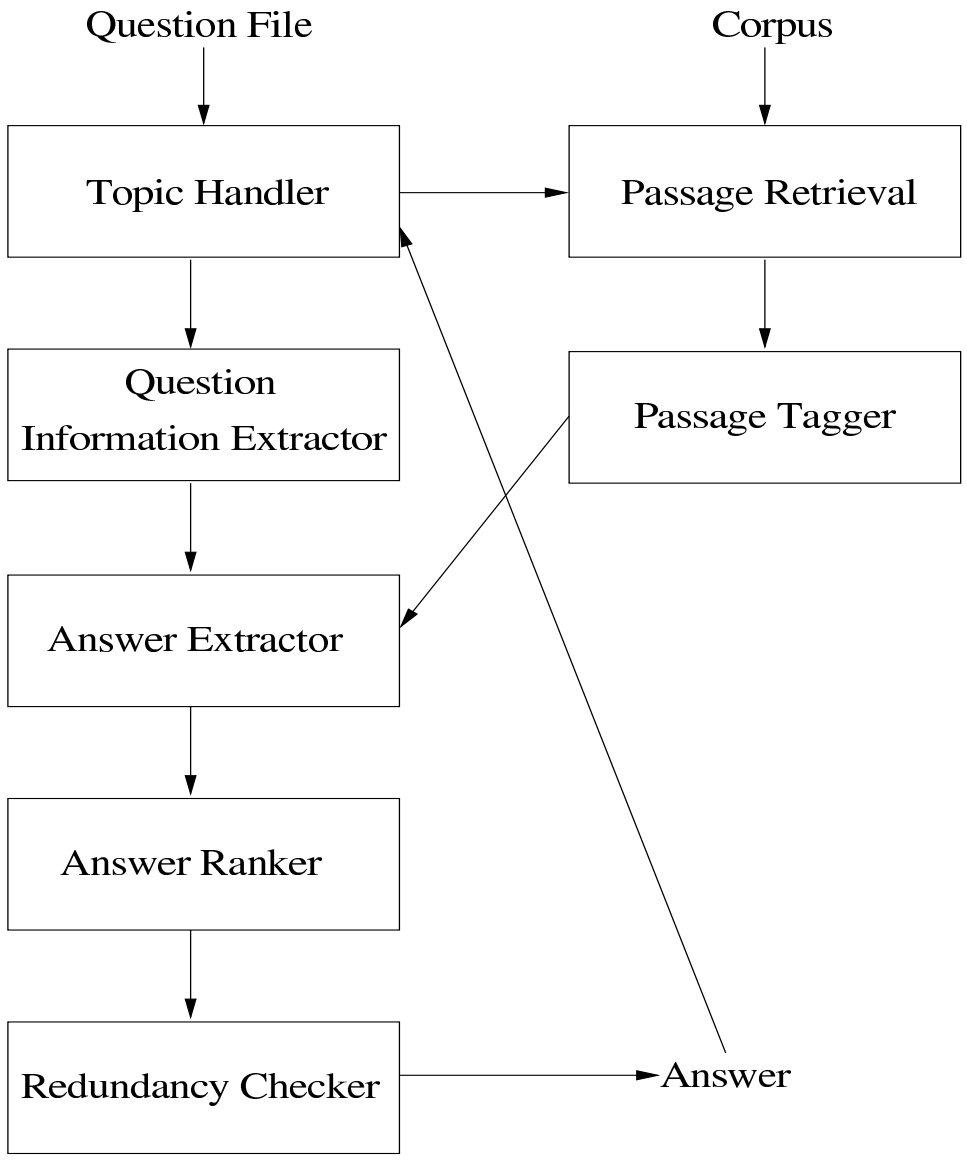

Figure 1: Model of Our Question Answering System

a system can query the documents. This will allow a system to process less documents. NIST also provides their top 1000 ranked documents for a question although, they do not guarantee that the answer will be contained in those 1000 documents. We used Managing Gigabytes (Witten, Muffat, and Bell, 1999) to index the AQUAINT collection by paragraph. Indexing by paragraph instead of by document, increases the frequency that the retrieved data is related to the target.

This module creates a query from the target. We are also using boolean queries rather than ranked queries. The query will retrieve all documents that contain the target because boolean queries retrieve all the documents that exactly match the query.

2.3 Passage Tagger

A program does not understand the meaning of words. Taggers and parsers are used to convey to the system, knowledge of what the words are and how they relate to each other. The system needs to be able to handle the information that is extracted from a document for it to be useful.

We tag the documents returned by the passage retrieval module with the Oak System tagger (Sekine, 2002) with named entities and chunked parts of speech. Adding a syntactic parser to this module would improve the knowledge gained about the passages.

2.4 Question Information Extractor

The information extracted from the question will give important clues on how to discover the answer. If no information is extracted from the question the correct answer will be impossible to find. Similar to the document tagger, the more information extracted from the question and used, the more precise the system will be at understanding the words and finding the answer.

This module and following modules have separate sections for the different question types because they handle each of them differently.

2.4.1 Factoid and List

This module determines the answer type, question type and the important words from the questions that give clues about where an answer could be found. Answer type of a question is useful because it gives the system knowledge of which named entity to look for. The question type is useful because it gives clues for which patterns the answer is likely to be found in.

The words of the question can give clues to the words that will be found around the answer. Matching words in the question to the documents retrieved is not the most intelligent way of approaching answering questions. A more efficient approach would be to put the question into a dependency form (Pasca and Harabagiu, 2001) and use it to match with the dependency form of a prospective answer. Finding word dependencies will require a syntactic parse of the question and retrieved passages. Proper nouns from the question are going to be represented in any paragraph containing a possible answer. Other words in the question might be represented in the question by a synonym which will not be found by simple pattern matching. Proper nouns will be used in answer extraction and other words will be used later in answer ranking.

2.4.2 Other

Other questions are stated as the string ”Other” with no extra information to extract.

2.5 Answer Extractor

2.5.1 Factoid and List

This module works with the tagged passages and the proper nouns from the question. A passage is sent to the answer ranking module if the passage contains the proper nouns from the question, and a named entity that matches the answer type.

2.5.2 Other

Answers for other questions are extracted using patterns for finding facts about a target. An example of one of the patterns the system uses, if the target is a person, is *"TOPIC,fact,"* if *fact* is not a name.

2.6 Answer Ranker

2.6.1 Factoid

The answer extractor may return multiple possible answers for a factoid question. This module gives a rank to each possible answer and returns the one with the highest rank. A possible answer is given weighted ranks for: how many times it appear in the possible answer list, if the passage it is found in include words from the question, and if it is found in a pattern associated with the type of question.

2.6.2 List

For list questions, every possible answer might not be a correct answer. This module will give points to each answer in a way similar to factoid questions. The answers that get high enough scores will be included in the list of answers given to the redundancy checker.

2.6.3 Other

We do not currently have a method for ranking facts for the other questions and all facts are passed to the redundancy checker.

2.7 Redundancy Checker

2.7.1 Factoid

Factoid questions require only one answer so no redundancy checking is needed.

2.7.2 List and Other

For list and other questions, a system will lose ranks for an answer list that contains redundancies. When this module returns an answer, it checks to see if the answer has already been in the list of answers given. This check is performs by checking if the answer contains a non-trivial word from answers already given. Examples of trivial words are: stop words, pronouns and prepositions.

3 Evaluation

This is the first year that University of Lethbridge has participated in TREC question answering track and we did not rank as competitively as we wanted. Table 1 outlines the range of scores for each type of question in the QA track for all participants and the results for our system (UofL1). If the questions that have no known answer in the documents were removed, our system would have received 0.043 for the factoid questions. The lower result is because our system will return NIL if the answer type of the question is undetermined.

Question Type	Median	Worst	Best	UofL1
Factoid	0.170	0.009	0.770	0.104
List	0.094	0.000	0.622	0.024
Other	0.184	0.000	0.460	0.023

Table 1: Evaluations Results for QA Track from TREC-2004

NIST changes the type of questions they ask from year to year to keep the question answering track challenging and to move the field in the direction of being able to answer any question. For instance, questions that start with "Where" or "Who" made up less then one percent of questions in the question answering track in 2003. This is significantly different then TREC-2004 where "Where" and "Who" questions made up 25 percent of the questions.

A system that ranks consistently high in the QA Track has taken into account many different domains of questions. Our system ranked low because we did not consider enough question domains. After receiving our TREC results, we were able to change our system to recognise a greater variety of question to answer type patterns. This change raised our attempts at answering questions from TREC-2004 from 24% to 70%. Without performing any other changes to our system the increase in attempts raised our overall accuracy from .104 to .217 on the questions from TREC-2004.

4 Conclusion

The TREC-2004 question answering track has been a learning experience for us. Preparation for next year will include many improvements to our system.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) research grant and the Alberta Heritage Foundation for Science and Engineering Research under the Research Excellence Envelope funding.

References

Pasca, M. and S. M. Harabagiu. 2001. Answer mining from on-line documents. In *Proceedings of 39th Annual Meeting and 10th Conference of the European Chapter, Workshop on Open-Domain Question Answering*, pages 38–45, Toulouse, France.

Sekine, S., 2002. *Proteus Project - OAK System (English Sentence Analyzer)*. http://nlp.cs.nyu.edu/oak.

Witten, I., A. Muffat, and T. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann.