Knowledge-Intensive Question Answering

Christof Monz Maarten de Rijke

Language and Inference Technology, ILLC, U. of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam
Email: {christof,mdr}@science.uva.nl

The full version of this paper appeared as C. Monz and M. de Rijke, Tequesta: The University of Amsterdam's Textual Question Answering System, in: E. Voorhees and D.K. Harman (eds), *The Tenth Text REtrieval Conference (TREC 2001)*, NIST Special Publication 500-250, pages 519–528, 2002.

With recent advances in computer and Internet technology, people have access to more information than ever before. Much of the information is available in free text with little or no metadata, and there is a tremendous need for tools to help organize, classify, and store the information, and to allow better access to the stored information. Research in information retrieval (IR) has made much progress in addressing this problem. However, current IR systems only allow us to locate documents that might contain the pertinent information; most of them leave it to the user to extract the useful information from a ranked list. This leaves the (often unwilling) user with a relatively large amount of text to consume. People have questions and they need answers, not documents.

Corpus-based question answering is designed to take a step closer to *information* retrieval rather than *document* retrieval. Briefly, the question answering (QA) task is to find, in a large collection of data, an answer to a question posed in natural language. Here's an example of a fact-based question that modern corpus-based QA systems are able to answer by returning a short text snippet (taken from a document in the collection) that is believed to contain an answer.

(1) *What river in the US is known as the Big Muddy?*

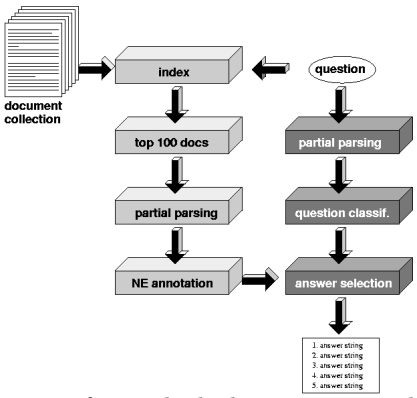
The QA system, and not the user, is responsible for analyzing the content of relevant documents and identifying text snippets with the answer.

The advantages of QA systems over document retrieval come at a price. The complexity of open-domain questions calls for an intricate mixture of natural language processing (NLP) and IR. IR is needed to identify candidate answer documents, and NLP has to provide a question analysis module to begin with, and part-of-speech tagging, shallow parsing, as well as named entity recognition and information extraction modules to identify answers within the candidate answer documents. The real challenge is that questions and candidate answer documents are often phrased in different vocabularies. To address this challenge various degrees of *bridging inference* are required, at the lexical level and at the level of argument structures. For instance, humans appear to use inference rules such as "X writes Y" imples "X is the author of Y" in answering questions, but such rules are hard to construct in a robust way.

Our Current Implementation. The annual TREC conferences have featured a track for evaluating QA systems since 1999 (TREC-8) [2]. The documents used in the task consist mostly of open domain newspaper articles. In TREC-10 (2001) participants were given 3Gb of text and 500 fact-based, short-answer questions such as those mentioned above. Not every question was guaranteed to have at least one document in the collection that explicitly answered the question.

Participating systems returned a ranked list of five strings per question, such that each string was believed to contain an answer to the question. Answer strings were limited to 50 bytes, and could either be extracted from the corresponding document or automatically generated from information contained in the document. Human assessors read each string and made binary decisions as to whether the string actually did contain an answer to the question in the context provided by the document. Given a set of judgments for the answer strings, the score computed for a submission was mean reciprocal rank (MRR): an individual question received a score of $1/n$, where n is the rank at which the first correct response was returned, or 0 if none of the five responses contained a correct answer. The score of a submission was the mean of the individual questions' reciprocal ranks.

The Language and Inference Technology group at the University of Amsterdam took part in the QA track in TREC-10 using the general knowledge-intensive strategy outlined above; see the figure to the right for a high-level overview of the system. With an MRR of 0.20, our scores were in the mid range.



Future Work. An ambitious roadmap for QA research was recently developed; it describes a program aimed at increasing the complexity of the types of questions that can be answered, the diversity of sources from which the answers can be drawn, and the means by which answers are displayed [1]. The roadmap includes a five year plan for introducing aspects of these research issues to the TREC QA track. The QA track in TREC-10 included the first steps of this roadmap, and TREC-11 will see a further implementation of these plans, e.g., by demanding *exact answers* where appropriate.

The QA track provides a very attractive setting for experimenting with mixtures of knowledge-intensive NLP and information retrieval. Our QA plans for the immediate future involve more sophisticated question classification as well as the construction and use of additional knowledge sources.

[1] S. Harabagiu et al. Issues, tasks, and program structures to roadmap research in question & answering (Q&A). URL: `http://www-nlpir.nist.gov/projects/duc/roadmapping.html`, October 2000.

[2] TREC: Text REtrieval Conference. URL: `http://trec.nist.gov`.