AskMSR: Question Answering Using the Worldwide Web

Michele Banko, Eric Brill, Susan Dumais, Jimmy Lin

Microsoft Research
1 Microsoft Way
Redmond, WA 98052
{mbanko, brill, sdumais}@microsoft.com
jimmylin@ai.mit.edu

Introduction

The design of the AskMSR question answering system is motivated by recent observations in natural language processing that for many applications, significant improvements in accuracy can be attained simply by increasing the amount of data used for learning (e.g., Banko & Brill, 2001). By taking advantage of the vast amount of online text available via the worldwide web, rather than relying on an approach that depends heavily on natural language intensive techniques, we developed a simple but effective question answering system. Many groups working on question answering use a variety of linguistic resources – part-of-speech tagging, parsing, named entiry extraction, WordNet, etc. We chose instead to focus on the tremendous resource that the web provides simply as a gigantic data repository. The web, which is home to billions of pages of electronic text, is orders of magnitude larger than the TREC QA document collection, which consists of fewer than 1 million documents.

Recently, other researchers have also looked to the web as a resource for question answering. These systems typically perform complex parsing and entity extraction for both queries and best matching web pages (Kwok et al., 2001, Buchholtz, 2001), which limits the number of web pages that they can analyze in detail. Other systems require term weighting for selecting or ranking the best-matching passages (Clarke et al., 2001, Kwok et al., 2001), and this requires auxiliary data structures. Our approach is distinguished from these in its simplicity and efficiency in the use of web resources.

An early version of the AskMSR system attained the sixth best accuracy in lenient scoring in the TREC 2001 Question Answering Track (Brill, et al. 2001; Voorhees and Harman, 2001).

Redundancy is Key

Automatic QA from a single, small information source is extremely challenging, as there is likely to be at most one answer in the source to any user's question. An analysis of the TREC 2001 query set found that only 37 of the 500 questions possessed at least 25 TREC documents containing a correct answer, and 138 of 500 have at least 10 TREC documents. So, while we could run our system directly on TREC documents, the lack of redundancy will limit its success.

The greater the number of information sources we can draw from, the easier the task becomes, since the answer is more likely to be expressed in different manners. For example, consider the difficulty of gleaning an answer to the question *"Who killed Abraham Lincoln?"* from a source which contains only the text *"John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life,"* versus one that also contains the transparent answer string, *"John Wilkes Booth killed Abraham Lincoln."*

Even when no obvious answer strings can be found in the text, redundancy (multiple, differently phrased, answer occurrences) can improve the efficacy of QA. For instance, consider the question: "How many times did Bjorn Borg win Wimbledon?" Assume the system is unable to find any obvious answer strings, but does find the following sentences containing "Bjorn Borg" and "Wimbledon", as well as a number:

(1) **Bjorn Borg** blah blah **Wimbledon** blah blah 5 blah
(2) **Wimbledon** blah blah blah **Bjorn Borg** blah 37 blah.
(3) blah **Bjorn Borg** blah blah 5 blah blah **Wimbledon**
(4) 5 blah blah **Wimbledon** blah blah **Bjorn Borg**.

By virtue of the fact that the most frequent number in these sentences is 5, we can posit that as the most likely answer.

Figure 1. AskMSR System Architecture
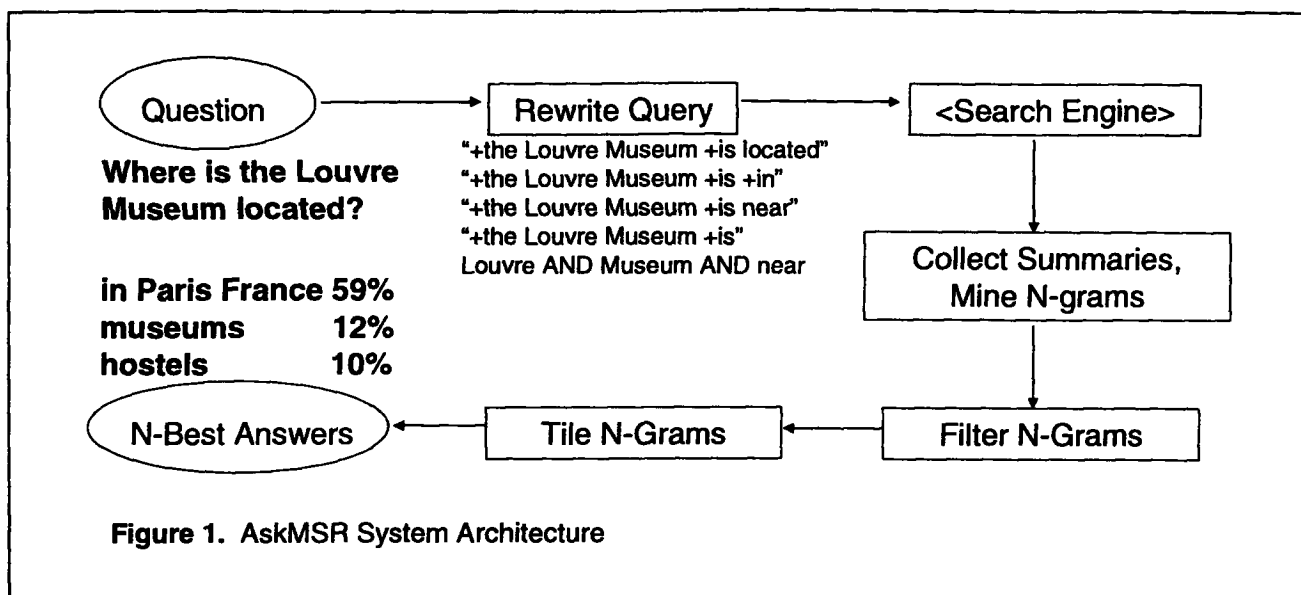
System Overview

A flow diagram of AskMSR is shown in Figure 1[1]. The system consists of four main components.

Rewrite Query. Given a question, the system generates a number of rewrite strings, which are likely substrings of declarative answers to the question. Such transformations from query to rewrite string make use of simple morphological information, but the process does not rely on part-of-speech tagging or parsing. There are fewer than 10 rewrite types, which vary from specific string matching to a simple ANDing of the query words. These strings are then formulated as search engine queries and sent to a search engine[2] from which page summaries are collected.

Mine N-Grams. From the page summaries returned by the search engine, n-grams are mined. 1-, 2-, and 3-grams are extracted from the summaries and scored according to their frequency of occurrence and the weight of the query rewrite that retrieved it. For reasons of efficiency, we use only the returned summaries and not the full-text of the corresponding web page.

Filter N-Grams. The n-grams are subsequently filtered and reweighted according to how well each candidate matches the expected answer-type, as specified by a handful of handwritten filters.

Tile N-Grams. Finally, the n-grams are tiled together where appropriate, so that we may assemble longer answers from shorter ones

---

[1] A more detailed description of the system is given in (Brill, et al., 2001)
[2] We used Google as our engine.

System Performance and Issues

The Table shows our TREC 2001 results from an early version of the AskMSR system. While we have improved the system a great deal since TREC 2001, much work remains to be done. We are now carefully studying the importance of various system components in hopes of improving things further.

AskMSR	Strict	Lenient
MRR	0.347	0.434
%Q with No Correct Resp	49.2	40

References

M. Banko and E. Brill (2001). Scaling to very very large corpora for natural language disambiguation. *ACL 2001*.

E. Brill, J. Lin, M. Banko, S. Dumais and A. Ng (2001). Data-intensive question answering. To appear in *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

S. Buchholz (2001). Using grammatical relations, answer frequencies and the World Wide Web for TREC question answering. To appear in *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

C. Clarke, G. Cormack and T. Lynam (2001). Web reinforced question answering. To appear in *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.

C. Kwok, O. Etzioni and D. Weld (2001). Scaling question answering to the Web. *WWW'10*.

E. Voorhees and D. Harmon, Eds. (2001). *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*.