Passage Retrieval vs. Document Retrieval for Factoid Question Answering

Charles L. A. Clarke Egidio L. Terra
School of Computer Science, University of Waterloo, Canada
{claclark,elterra}@plg.uwaterloo.ca

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Measurement, Performance, Experimentation

Keywords

Question answering, Passage retrieval

1. INTRODUCTION

Question answering (QA) systems often contain an information retrieval subsystem that identifies documents or passages where the answer to a question might appear [1–3, 5, 6, 10]. The QA system generates queries from the questions and submits them to the IR subsystem. The IR subsystem returns the top-ranked documents or passages, and the QA system selects the answers from them.

In many QA systems, the IR component retrieves entire documents. Then, in a post-retrieval step, the system scans the retrieved documents and locates groups of sentences that contain most or all of the question keywords [3, 10, and others]. These sentences are subjected to further analysis to select the answer. In other QA systems, a passage-retrieval technique is employed to directly identify locations within the document collection where the answer might be found, avoiding the post-retrieval step [1, 2, 5, 6, and others].

In this context, a "relevant" document or passage is one that contains an answer. We utilize this notion of relevance to evaluate an IR subsystem in isolation from the rest of its QA system by applying standard measures of IR effectiveness. By restricting our evaluation to a single subsystem we hope to gain experience that is applicable to QA systems beyond our own. An assumption inherent in this approach is that improved precision in the IR subsystem will translate to improved performance of the QA system as a whole. This assumption holds for our own system, and should (at least) hold for any system that exploits redundancy—that takes advantage of the observation that answers tend to occur in more than one retrieved passage [1, 2, 5].

In this paper we compare a successful passage-retrieval method [1, 5] with a well-known and effective document-retrieval method: Okapi BM25 [7]. Our goal is to examine the relative advantages and disadvantages of each method in the context of a question answering task. We apply both methods to two problems: the problem of finding documents that contain an answer, and the related but distinct problem of finding passages that contain an answer.

SIGIR'03, July 28–August 1, 2003, Toronto, Canada.
ACM 1-58113-646-3/03/0007.

2. THE TREC 2002 QA TASK

Every year since 1999, TREC has included a QA track to evaluate the performance of QA systems using a series of fact-finding questions [8]. At TREC 2002, the QA track used 500 questions. For each question, a system was required to return an exact answer and a document supporting the answer. The source for the supporting document was restricted to a 3GB corpus of newspaper articles supplied by the TREC organizers. After the evaluation was completed, the TREC organizers released a list of the correct answers along with the documents that support them. These questions, answers and documents form the basis for the experiments reported in this paper.

3. PASSAGE RETRIEVAL

We examine two passage-retrieval methods. The implementation of the first method ("MultiText passages") was described by Clarke et al. [1] and independently implemented by Lin et al. [5] for use at TREC 2002. The method is related to passage-retrieval methods developed by others [4, 9].

This method identifies short text fragments, or "hotspots" where query terms appear in close proximity. The score of a hotspot is computed from its length and the weights of the terms it contains. A passage returned by the method consists of a hotspot and a window of text surrounding it, where the size of this window may be determined dynamically at query time and may even consist of an entire document. Regardless of the window size, the location of the hotspot within the window is retained for use by the QA system.

While a hotspot may be of any length, and is not constrained by sentence boundaries, it is generally less than 20 words in length, and may be very short, comprised only of query terms adjacent to one another. The answer to a question often appears within a hotspot or close to it.

Figure 1 illustrates this effect. We retrieved the top 20 MultiText passages for each of the TREC 2002 questions using a window size of 200 words. In these passages, the average length of the hotspot is 9 words. The figure plots the location of answers relative to the hotspot. The x-axis of the figure indicates the word position of an answer before or after the hotspot. Answers that appear within the hotspot are plotted at word position 0. The y-axis indicates the

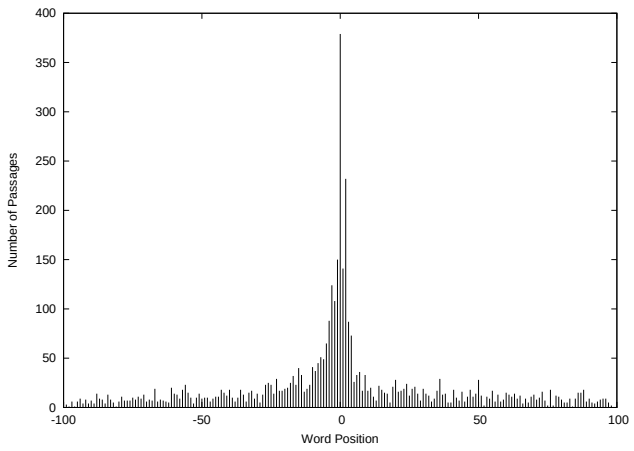

Figure 1: Answer Location Relative to the Hotspot

number of passages that have an answer starting at that word position. As seen in the figure, answers tend to cluster in and around the hotspot.

Our second passage-retrieval method is an adaptation of the Okapi BM25 document-retrieval method to the problem of passage retrieval. This method divides each document into sub-documents and applies the Okapi similarity measure to rank the collection of sub-documents. Following Prager et al. [6] we divide each document into sub-documents of three sentences each, allowing the sub-documents to overlap, with a new sub-document starting at each sentence boundary. Our implementation of Okapi BM25 is based on the description of Robertson et al. [7] with typical parameter values ($b = 0.75$, $k_1 = 1.2$, $k_2 = 0$, $k_3 = \infty$).

For the purposes of comparing the two passage retrieval methods, we restrict MultiText passages to exactly three sentences. If a hotspot is longer than three sentences, we discard it. If a hotspot is shorter than three sentence, we return a three-sentence window containing the hotspot.

For both methods we retrieve at most one passage from each document. QA systems that exploit redundancy depend on the independent occurrence of answers, and multiple passages from the same document are unlikely to exhibit this independence [1].

4. PASSAGES VS. DOCUMENTS

The MultiText passage-retrieval method may be treated as a document-retrieval method by extending the window around the hotspot to include the entire document. Thus, we can treat both MultiText and Okapi as passage- and document-retrieval methods and compare them on an equal footing.

The table in figure 2 compares the two retrieval methods on both sub-documents and full documents, reporting precision values at five levels. On sub-documents, MultiText outperforms Okapi in all cases. Using the Wilcoxen matched-pairs signed-ranks test at the 95% confidence level, the differences are significant in all cases except "precision@1". On full documents, Okapi and MultiText provide similar performance, with none of the differences being significant at the 95% level. Full-document retrieval consistently outperforms sub-document retrieval, with all differences being significant.

	Sub-documents		Full documents	
Precision	MultiText	Okapi	MultiText	Okapi
@1	0.313	0.284	0.428	0.408
@5	0.238	0.212	0.321	0.337
@10	0.202	0.179	0.276	0.290
@20	0.169	0.146	0.234	0.239
@40	0.140	0.121	0.198	0.194

Figure 2: Passage Retrieval vs. Document Retrieval

5. CONCLUSION

QA systems are usually evaluated end-to-end, on the basis of their ability to find answers to questions. Unfortunately, QA systems are complex — with many subsystems — and an end-to-end evaluation often does not allow us to understand the relative contribution that each subsystem makes to the overall performance. Here, we examine the IR subsystem in isolation.

Our results suggest that full-document retrieval may be desirable in a QA system, since higher precision may translate into better performance. However, when full documents are retrieved, post-retrieval passage selection may still be required. The MultiText method, even when used for full-document retrieval, identifies a hotspot where the answer is more likely to found, combining the advantages of both passage and document retrieval.

6. REFERENCES

[1] C. L. A. Clarke, G. V. Cormack, and T. R. Lynam. Exploiting redundancy in question answering. In *24th ACM SIGIR*, pages 358–365, September 2001.

[2] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web question answering: Is more always better? In *25th ACM SIGIR*, August 2002.

[3] U. Hermjakob, A. Echihabi, and D. Marcu. Natural language based reformulation resource and web exploitation for question answering. In *2002 Text REtrieval Conference*, 2002.

[4] M. Kaszkiel and J. Zobel. Passage retrieval revisited. In *20th ACM SIGIR*, pages 178–185, July 1997.

[5] J. Lin, A. Fernandes, B. Katz, G. Marton, and S. Tellex. Extracting answers from the Web using data annotation and data mining techniques. In *2002 Text REtrieval Conference*, 2002.

[6] J. Prager, E. Brown, A. Coden, and D. Radev. Question-answering by predictive annotation. In *23rd ACM SIGIR*, pages 184–191, August 2000.

[7] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7: Automatic ad hoc, filtering, VLC and interactive track. In *7th Text REtrieval Conference*, 1998.

[8] E. M. Voorhees. Overview of the TREC question answering track. In *2002 Text REtrieval Conference*, 2002.

[9] W. A. Woods, L. A. Bookman, A. Houston, R. J. Kuhns, P. Martin, and S. Green. Linguistic knowledge can improve information retrieval. In *Applied Natural Language Processing Conference*, May 2000.

[10] H. Yang and T.-S. Chua. The integration of lexical knowledge and external resources for question answering. In *2002 Text REtrieval Conference*, 2002.