UMass Complex Interactive Question Answering (ciQA) 2007: Human Performance as Question Answerers [Notebook Version]

Mark D. Smucker, James Allan, and Blagovest Dachev
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst

Abstract

Every day, people widely use information retrieval (IR) systems to answer their questions. We utilized the TREC 2007 complex, interactive question answering (ciQA) track to measure the performance of humans using an interactive IR system to answer questions. Using our IR system, assessors searched for relevant documents and recorded answers to their questions. We submitted the assessors' answers without modification as one of our runs. For our other submission, one of the authors used our IR system for an unlimited time and recorded answers to the questions. We found that human performance using an interactive IR system for question answering is variable but that interactive IR systems offer the potential for superior question answering performance.

1 Introduction

The complex, interactive question answering (ciQA) track looks at complex information needs and aims to investigate the performance gains attainable when a QA system has the chance to interact with users. This year, for each question the track allowed participants to provide a web address (URL) at which the participants could provide any sort of web page to interact with the assessor for 5 minutes.

Today when users have questions, one of their likely tactics for finding answers is to use an information retrieval (IR) system. The ciQA track provided us with the unique opportunity to measure the assessors' abilities to answer their own questions using an interactive IR system. Rather than attempt to use interaction to boost the performance of a QA system, we wanted to see how good users are at answering their questions using information retrieval.

A good information retrieval system should inherently be a good complex, interactive question answering system. Or as we told the assessors, "Our belief is that human searchers, such as yourself, can find answers faster and more accurately than computers." Rather than build question answering systems, our intent is to build interactive IR systems that enable people to answer questions.

To measure the assessors' performance at answering their questions using an IR system, we created an interactive, web-based IR system. Assessors were free to issue queries, view documents, and save answers. We did not place any restrictions on the type of answer the assessors could enter and save. Assessors could copy text from a displayed document, a result snippet, or type in their own answer.

While confident that the assessors would find answers, we hedged our bet by utilizing our two runs to give the assessors 10 minutes on each question. For each run, we provided the same IR system and when the assessor returned to a question, any previously saved answers were still displayed.

One of our submitted runs was the exact set of answers saved by the assessors after all 10 minutes of interaction. As such, the assessors judged their own saved answers. This allows us to not only compare an interactive IR system's performance to the other submitted interactive QA systems, but to also measure to what extent assessors agree with themselves.

A concern with interactive systems is that users have difficulty in judging recall and often quit searching too soon. For our other run, we had one of the authors use the system for an unlimited time to find as many answers for each question as possible. This run allows us to get a sense of the maximum performance of our interactive system.

2 Methods

We built a fully interactive IR system with facilities for recording answers to questions. We stress

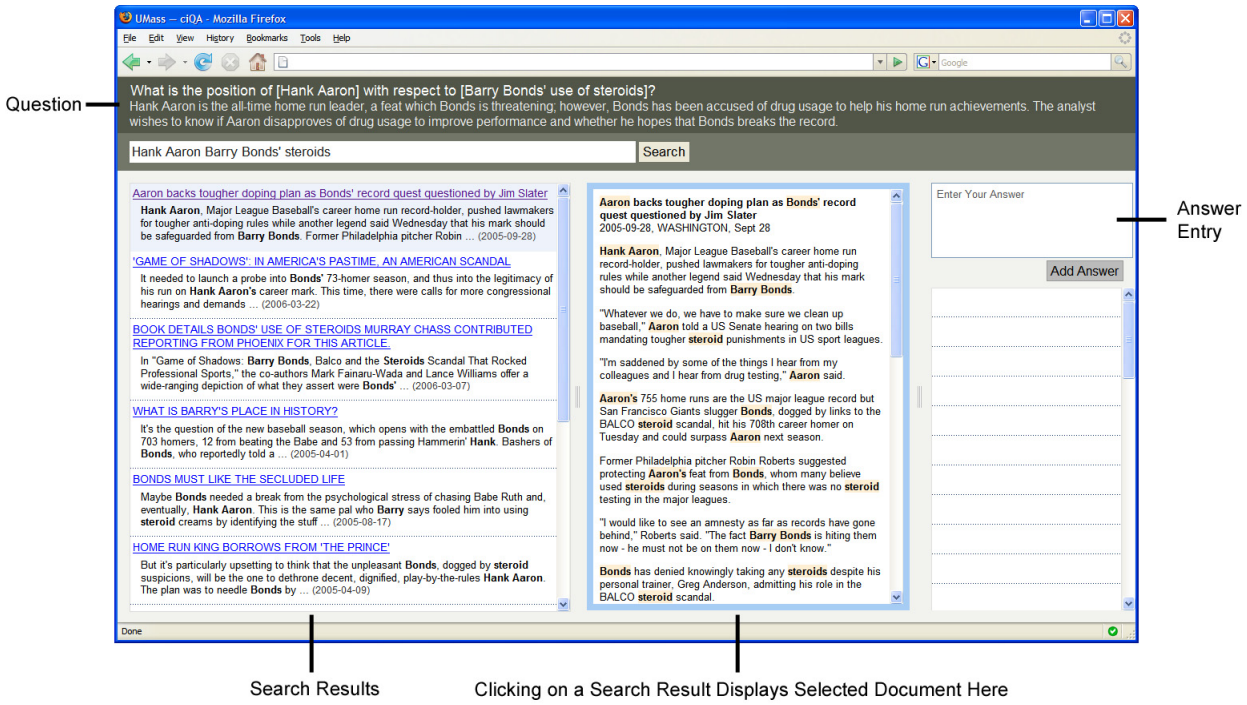

Figure 1: A screenshot of the web-based interface for our fully interactive, IR system.

that the IR system was fully interactive because in past ciQA and HARD track evaluations, participants largely supplied static HTML forms.

As Figure 1 shows, the user interface to our IR system consisted of three vertically oriented panes. The left pane showed search results with query-biased snippets. Clicking on a result showed the respective document in the middle pane and also changed the color of the link allowing users to keep track of already examined documents. Both the snippets and displayed document had query terms highlighted. The right hand pane provided a text box allowing the user to enter and save an answer to the question. A list of the user's saved answers appeared below the answer entry box. The assessors could delete saved answers.

Each assessor had the chance to read a tutorial on how to use our search engine and about our research goals. At the end of the tutorial, they could experiment with the live system and the ciQA designated throw away question.

Our IR front end client was a modern AJAX-like interface written in XHTML, CSS, and JavaScript. Submitting queries, clicking on results to view documents, and saving answers all occurred within the same web page and did not require an entire page refresh for each event. We built the back-end server using a combination of the Apache web server, PHP, mySQL, C++, and the Indri [6, 5] retrieval system.

We supported a simple query language. Users could specify phrases by enclosing a phrase with double quotes. Users could also force all results to contain a query term by preceding the term with a plus sign. The Indri query language provides support for both of these features.

When the assessor first accesses the system for a given question, the system shows the results for a default query created automatically from the templated question as shown in Figure 1. To create the query, we extracted the terms within the slots of the template and then removed stop words. The remaining terms formed a bag of words query that we implemented using Indri's #combine operator. For example, the question "What is the position of [Hank Aaron] with respect to [Barry Bond's use of steroids]?" results in the Indri query #combine(Hank Aaron Barry Bonds steroids). Our tutorial encouraged assessors to change the query to meet their needs.

We used the same system for each run and also used the same tutorial for each run. We provided a link to explain to the assessor why they were seeing the same tutorial a second time. For each question, the interface showed previously saved answers and also

kept track of viewed documents to allow the links to the documents to be properly highlighted. The system did not save any query state and thus the assessor saw for a second time the default query and results when returning to the interface.

We annotated the sentences in the AQUAINT2 collection using a locally modified version of a sentence splitter [1]. To construct the query-biased snippets for each document, we converted the user's query to a bag-of-words query and then retrieved the top two scoring sentences from the document.

We stemmed all words with the Porter stemmer built into Indri and used an in-house list of 418 stop words. For retrieval, we used Indri's default parameters, which includes setting the Dirichlet prior smoothing parameter to a value of 2500.

3 Submissions

We constructed our baseline submission, **UMassBaseAut**, to be similar to the displayed query-biased snippets for each question's default query. As described above, the default query consisted of the words from the slots of the templated question. Using the default query, we retrieved the top 10 documents, which are the same 10 documents shown initially to the assessor. For each of these documents, we returned at most the top 2 sentences as answers for a maximum of 20 answers per topic. Some documents contained a single sentence and as a result we returned less than 20 answers in some cases. Unlike our displayed snippets, we did not truncate the sentences returned as answers. Our baseline represents the state from which the assessors started their usage of the system.

Our second submission, **UMassIntA**, consists of the answers provided by the assessors during their interactions with the system. For both of our allowed runs, we had the assessors interact with the same live IR system to obtain 10 minutes of interaction on each question. As described above, the system retained its state between "runs." Assessors directly entered answers to the questions. We submitted the answers saved by the assessors with no modification. Some assessors did delete some of their submitted answers, and we did not submit deleted answers. As we used the interaction automatically to produce the submission, we marked this as an automatic run. During one day of interaction, the assessors experienced network slowdowns, which likely hurt their ability to find and record answers.

We constructed our third submission, **UMassIntM**, by having one of the authors use our interactive system without a time limit to find answers. For questions where we found over 7000 non-whitespace characters of answers, we edited the answers to fit within the limit. Most answers are in the form of snippets of text extracted directly from the source documents.

4 Results and Discussion

Table 1 summarizes the results of our three submissions. Measures are computed in the same manner as the 2006 ciQA track [2] with pyramid nugget based scoring [4]. The official results report the F measure with a $\beta = 3$, which places 3 times more importance on recall than precision. We also show the F measure with $\beta = 1$, which places equal weight on recall and precision. Unless we specify otherwise, all F score mentions will be with respect to the official F score with a $\beta = 3$.

The median F score of all ciQA final runs (both automatic and manual) was reported by NIST to be 0.361. The best and worst scores were reported to be 0.503 and 0.156, respectively.

We were somewhat surprised by our baseline's performance. The baseline, UMassBaseAut, had a F score of 0.318, which was only 12% below the median performance of all final submitted runs (0.361). Our baseline appears to be representative of many of the baselines in its strong performance relative to the submitted runs. The median performance of the submitted baselines was 0.359 as reported by NIST.

Our automatic submission, UMassIntA, performed 9% better than our baseline with an F score of 0.347 and performed near the median performance of all final runs. The improvement over the baseline came from an improved precision that rose from 0.154 for the baseline to 0.427 for the assessor's answers. If we place equal importance on recall and precision ($\beta = 1$), UMassIntA achieves a 59% improvement over the baseline with an F score of 0.333 compared to the baseline's 0.210.

Our manual submission, UMassIntM, matched the best reported performance with an average F ($\beta = 3$) score of 0.503. Even with issues of inter-annotator agreement, we see this as informative of the performance attainable by a human using an IR system to answer questions. With an equal weight given to precision and recall ($\beta = 1$), UMassIntM performs 12% worse than UMassIntA (F score of 0.293 compared to 0.333).

While we have just begun to analyze our results, we have observed that significant variation exists among the assessor's performance at answering their ques-

Run	Average over 30 Questions				F ($\beta = 3$) Summary			
	Recall	Precision	F ($\beta = 3$)	F ($\beta = 1$)	#Best	#Worst	#Zeros	#>Median
UMassBaseAut	0.374	0.154	0.318	0.210	NA	NA	4	12
UMassIntA	0.362	0.427	0.347	0.333	5	6	5	14
UMassIntM	0.658	0.210	0.503	0.293	8	0	0	23

Table 1: The average recall, precision and F scores for our three submissions and the number of questions for which F with $\beta = 3$ was the best, worst, and greater than the question median for final runs. The number of questions for which the submission scored a zero is reported in the column labeled #Zeros. Of note, only one question in common between UMassBaseAut and UMassIntA received a zero.

tions using our system. For the 4 topics assigned to assessor 5, the assessor only entered one answer into our system. The assessor later judged this single answer as not containing a single nugget. Thus, assessor 5 produced 4 of the 5 zero scoring questions for our UMassIntA submission. In contrast, assessor 8's answers to three of the assessor's four questions achieved the highest F scores of all submitted runs. It appears that some of assessor 5's interactions with our system occurred during a network slowdown, which could partly explain the lack of answers.

5 Related Work

Lin [3] proposed an evaluation framework of recall curves and using this framework compared the performance of an IR system to the submitted runs for the TREC 2004 and 2005 question answering tracks. Lin found that while the QA systems were superior for *factoid* questions, for *other* questions the performance of the IR and QA systems were similar.

Lin's experiment is similar to our baseline submission where the results being measured are the static ranked lists generated from a query. In contrast, our work looks at the performance of an IR system being used as an interactive tool for question answering.

6 Conclusion

Our results suggest that interactive IR systems are inherently powerful for question answering. We believe that bridging the gap in performance between our automatic and manual submissions will be possible with the creation of innovative interaction mechanisms that allow users to better find, understand, and organize both documents and answers.

7 Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part by NSF Nano # DMI-0531171. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

[1] Cognitive Computation Group, University of Illinois at Urbana-Champaign. Sentence segmentation tool, 2001. `http://l2r.cs.uiuc.edu/~cogcomp/atool.php?tkey=SS`.

[2] D. Kelly and J. Lin. Overview of the TREC 2006 ciQA task. *SIGIR Forum*, 41(1):107–116, 2007.

[3] J. Lin. Is Question Answering Better than Information Retrieval? Towards a Task-Based Evaluation Framework for Question Series. In *HLT/NAACL 2007*, pages 212–219. Association for Computational Linguistics, 2007.

[4] J. Lin and D. Demner-Fushman. Will pyramids built of nuggets topple over? In *HLT/NAACL 2006*, pages 383–390. Association for Computational Linguistics, 2006.

[5] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *IPM*, 40(5):735–750, 2004.

[6] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language-model based search engine for complex queries (extended version). Technical Report IR-407, CIIR, Department of Computer Science, University of Massachusetts Amherst, 2005.