Human Question Answering Performance Using an Interactive Document Retrieval System

Mark D. Smucker
Department of Management Sciences
University of Waterloo
mark.smucker@uwaterloo.ca

James Allan
Department of Computer Science
University of Massachusetts Amherst
allan@cs.umass.edu

Blagovest Dachev
TST Media
Minneapolis, MN, USA
blago@dachev.com

ABSTRACT

Every day, people answer their questions by using document retrieval systems. Compared to document retrieval systems, question answering (QA) systems aim to speed the rate at which users find answers by retrieving answers rather than documents. To better understand how document retrieval systems compare to QA systems, we measured the performance of humans using an interactive document retrieval system to answer questions. We first measured the ability of users to answer their questions using an interactive document retrieval system, and then compared the users' performance with the document retrieval system to question answering systems. We found that while users can successfully answer their questions using a document retrieval system, question answering systems have the potential to significantly increase the rate at which users find answers.

Categories and Subject Descriptors

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

General Terms

Performance, Experimentation, Human Factors

Keywords

Interactive information retrieval, human performance, question answering, ciQA, TREC

1. INTRODUCTION

In 1999, the first TREC question answering (QA) track established as a goal the retrieval of short answers rather than documents under the assumption "that users would usually prefer to be given the answer rather than find the answer themselves in a document" [17].

QA and other focused-retrieval systems [9] aim to eliminate much of a user's overhead in finding information. At the front end, the user of a QA system gets to enter a natural language question rather than a keyword query. The perfect QA system then finds, extracts, and composes a concise answer and saves users all the time they would have to spend on these tasks if they used a document retrieval system.

A user of a document retrieval system must first transform a question into a query suitable for the retrieval system. The document retrieval system then generates a ranked list of documents. Next the user must evaluate the list and decide which documents look like good candidates for answering the question. Once the user selects a document, many systems provide little to no help in finding relevant material within the document beyond query term highlighting.

Nevertheless, the dominant question answering systems today are document retrieval systems. Each and every day, millions of queries are issued and answers are found using the major web search engines. How good are users at answering their questions using document retrieval systems? The TREC 2007 complex, interactive question answering (ciQA) track [3] provided us with a unique opportunity to answer this question.

The ciQA TREC track looked at complex information needs and aimed to investigate the performance gains attainable when a QA system has the chance to interact with users. Assessors at the U.S. National Institute of Standards and Technology (NIST) generated questions, interacted with systems, and judged the quality of answers. For each question, the 2007 ciQA 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. The NIST assessors were both the users and the eventual relevance assessors. We will simply refer to the users/assessors as assessors throughout this paper.

At our URL, we provided the assessors with a fully interactive, document retrieval system. Figure 2 shows the interface, which we describe in detail in Section 3.4. We asked the assessors to use the IR system to search for answers and to save all found answers.

We submitted to NIST the exact set of answers saved by the assessors. The other participating groups submitted both baseline QA results and post-interaction QA re-

IIiX 2012, Nijmegen, The Netherlands

sults. The baseline QA results represented the best that the QA systems could do without interacting with the assessors. The assessors then judged the answers from all submitted systems. For our submission, the assessors judged their own answers as though they had been produced by a QA system. The ciQA evaluation was nugget-based. A *nugget* represents a single, atomic answer. An answer submitted by a QA system can contain zero or more nuggets.

Our experiment allowed us to measure the assessors' performance at answering their questions using a document retrieval system and compare this performance to the other participants' QA systems. We were thus able to not only measure the rate at which nuggets can be found with a document retrieval system, we also were able to measure the degree to which assessors successfully find nuggets in their own, presumably, correct answers.

Our contributions in this paper include showing that:

- If the only effort in using QA systems is to read the answers, then QA systems should be superior to document retrieval systems for the ciQA task. On the other hand, if the user has to spend time to understand the context of each answer, this overhead has the potential to eliminate the advantage of QA systems.
- While the assessors worked at different rates, over a period of 10 minutes, the NIST assessors showed a linear gain in recall over time.
- The assessors found their own answers to be less than perfect. On average, the assessors only had 0.6 nuggets per submitted answer.

This work extends our TREC report [15] with additional experiments, results, and analyses. We next review related work, our methods and materials, and finally present and discuss our results.

2. RELATED WORK

The question answering component of the ciQA track has its roots in the *definition* questions of the TREC 2003 QA track. Voorhees provides a good review of the QA track from TREC-8 through TREC 2003 [18]. Measuring human performance using IR systems has a long history and was the focus of the TREC interactive track [5].

For TREC-9 (2000), the interactive track task used a fact-finding task that required users to view multiple documents to construct an answer [7]. Many participating sites explored the effect of different interfaces and retrieval systems on searcher performance, but to our knowledge, sites did not compare human performance with the IR systems to automatic question answering systems.

In a similar fashion to our work, Wu et al. [19] tested "human versus machine" in the topic distillation task of the TREC 2003 Web interactive track. The distillation task required the gathering of key web pages that cover a topic without overlap. Wu et al. found that the automatic distillation system's performance equaled that of humans using a regular IR system for 10 minutes.

Lin [10, 11] proposed an evaluation framework of recall curves, and using this framework, he compared the performance of an IR system to the submitted runs for the TREC 2004 and 2005 question answering tracks. An important observation made by Lin that we adopt is to have an evaluation metric that reflects the rate at which the user finds information. Lin simulated the behavior of users of both the document retrieval system and the QA systems. In each case, Lin made certain reasonable assumptions about what text would be read by users. Lin then plotted recall as a function of non-whitespace characters.

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. Factoid questions are closed-class questions with a single answer given typically as a short noun phrase. Questions of type *other* in TREC aim to find many nuggets of information regarding a target. For ciQA, the evaluation framework was designed to be the same as for *other* questions except that instead of a target, templated questions were posed by each of the NIST assessors. We discuss ciQA in greater detail in the next section.

In our work, we have extended Lin's recall curves to plot recall vs. time as opposed to Lin's plots of recall vs. number of characters retrieved. We've replaced Lin's user simulation of the document retrieval system with real human subjects (the NIST assessors). We simulate usage of the QA systems to obtain estimates of the rate at which users could discover information using a QA system.

Erbach has looked at human performance at question answering with a document retrieval system [6]. Erbach established a question answering baseline for human performance with a document retrieval system. Erbach found that QA systems had the same accuracy as humans if humans were limited to only 34 seconds per question. Erbach also found that QA systems achieve higher recall than humans. Unlike our experiments, Erbach's users were not the same as the assessors that judged the QA systems.

Xu and Mease [20] have investigated the use of *task completion time* to evaluate retrieval systems. Using a set of informational search tasks collected from users, they then ask their study participants to complete the search tasks and measured the amount of time till the participant found an answer to the task. Xu and Mease found that task completion time could be used to distinguish between different quality IR systems. In contrast to the tasks used by Xu and Mease, the ciQA questions require finding multiple answers.

3. MATERIALS AND METHODS

We conducted our experiments within the framework provided by the 2007 TREC complex, interactive question answering (ciQA) track [3]. The ciQA track's goals were to address questions that are more complex than closed-class questions such as "Where is the Taj Mahal?" and to look at how interacting with the user can improve the performance of QA systems. Our experiments utilized the track to measure the performance of humans using an interactive document retrieval system to answer questions.

The ciQA track followed the same three step process of its predecessor, the HARD track [1]: submit baselines, interact with assessors, submit final runs. Participating sites/groups created and submitted a baseline using only the NIST assessors' questions as input. The baseline captures performance levels before any user interaction. After submitting a baseline, each site had the opportunity to have two sets of interactions with the assessors. For each set, the site had the chance to interact with an assessor for each question for a maximum of 5 minutes. Using these sets of interaction, sites then prepared their final submissions. In 2007, sites were al-

Template 1, Question 56: What evidence is there for transport of [illegal immigrants] from [Croatia] to [the European Union]? Narrative: The analyst desires to know the nationality of both the smugglers and the illegal immigrants, as well as the routes and methods used for the transport.
Template 2, Question 67: What [common interests] exist between [Yo Yo Ma] and [Itzhak Perlman]? Narrative: The analyst would like to know of joint performances in which the two great musicians participated, as well as facts about their lives and education and other things that the two men have in common.
Template 3, Question 73: What effect does [lycopene] have on [reducing the risk of cancer]? Narrative: The analyst would like to know of any evidence in which lycopene, an antioxidant found in red pigments like tomatoes, prevents or reduces the risk of cancer in humans.
Template 4, Question 76: What is the position of [China] with respect to [Taiwanese independence]? Narrative: The analyst is interested in the intention of China toward Taiwan. Specifically, how does China view the Taiwanese movement toward independence.
Template 5, Question 85: Is there evidence to support the involvement of [Hezbollah] in [Argentina]? Narrative: The analyst desires to know what evidence exists for or against activities by the middle east terrorist organization, Hezbollah, inside the country of Argentina.

Table 1: Example questions.

lowed to submit two baselines and two post-interaction runs, which typically correspond to the two interaction sets.

3.1 Questions, Assessors, Collection

The ciQA 2007 TREC track used 30 questions. Questions consisted of two parts: a templated question and a longer narrative. There were 5 template types. Table 1 shows examples of the questions. The track divided the 30 questions among 8 assessors. Most assessors were responsible for 4 questions and two assessors did 3 questions. The ciQA track used the AQUAINT2 document collection. This collection consists of 906,777 documents from newswire sources.

3.2 Interaction

In 2007, ciQA had an additional goal of going beyond the one-shot interactions allowed in previous years. In previous years, the ciQA and HARD tracks allowed participants to submit an HTML form that the NIST assessors would fill out. For 2007, participating sites provided a web address (URL) for each question to NIST. At the URL the site could build any web-based system to have nearly unlimited interaction with the assessors. In addition to a URL for each of the 30 questions, sites provided a URL at which they could offer instructions or a tutorial on usage of their system. Before interacting with a site's system, the assessors first went to this "tutorial" URL.

NIST conducted an exit questionnaire following the assessors' interactions with all the systems. Questions ranged from ease of interaction to open ended feedback.

3.3 Evaluation

The ciQA track used a nugget-based evaluation. Each run could return as many answers to each question as desired up to a 7000 non-whitespace character limit. There was no requirement that systems break their responses up into separate answers, but most systems returned sentences as answers. Assessors created a list of *nuggets*. A nugget represents a single, atomic answer.

ID	Vital	Nugget
1	1.000	Both musicians went to Juilliard.
2	0.625	Both men have won at least 15 Grammy awards.
3	0.750	Both men performed solos in the movie "Memoirs of a Geisha."
4	0.750	Isaac Stern cultivated the careers of/discovered both men.
5	0.875	Both musicians performed on the telecast "Thirty Years of Live at Lincoln Center."
6	0.875	Both musicians performed at the Kennedy Center for the 75th anniversary of the National Symphony Orchestra.
7	0.625	Both men are musicians.

Table 2: The 7 nuggets for question 67 and their pyramid vital score. Question 67 asks "What common interests exist between Yo Yo Ma and Itzhak Perlman?" See Table 1 for the full narrative of question 67.

For example, question 67, which asks "What common interests exist between Yo Yo Ma and Itzhak Perlman?" was determined by the NIST assessor who asked the question to have in total 7 nuggets. Table 2 shows the nuggets created for question 67. The full narrative for question 67 is shown in Table 1.

For each submitted answer, the assessors determined which nuggets, if any, exist in the answer. An answer may contain zero or more nuggets. Nuggets are only counted once, i.e. duplicate nugget mentions count as returning the nugget once.

As example of this process, Table 3 shows the answers submitted by Assessor 8 using our retrieval system for question 67. In the first answer, the assessor found it to contain nugget 3: Both men performed solos in the movie "Memoirs of a Geisha." The fourth answer submitted by assessor 8 to our system, was later judged by the assessor to be a duplicate of the first, and thus this answer is counted as having no nuggets in it.

Some nuggets are considered more important than other nuggets. To place a value on each nugget, NIST constructs a *nugget pyramid* [12]. All 8 of the assessors judge each nugget as either being a *vital* or an *okay* nugget. The assessor in charge of the question, then judges nuggets one more time. The *vital score* of a nugget is the fraction of judgments that were vital. For example, if a nugget receives 1 judgment as vital and 8 okay judgments, then its vital score is 1/9. Nugget vital scores are normalized on a per question basis such that the nugget with the most vital votes has a score of 1. For example, Table 2 shows the vital scores assigned to the nuggets for question 67.

For each question, recall is computed as the sum of the vital scores of the returned nuggets divided by the sum of the vital scores for all known nuggets.

The official measure of the 2007 ciQA track was the F measure with $\beta = 3$, which weights recall as being three times as important as precision. The F-measure is a set-based measure. While our assessors produced a set of an-

No.	Time	Nuggets	Answer
1	51.6	3	Yo-Yo Ma reunites with John Williams and Itzhak Perlman for 'Geisha' score
2	97.4	5	Besides clips of performances by Ma, "Thirty Years of Live from Lincoln Center" will feature the likes of Itzhak Perlman,
3	121	7, 1	Julliard the prestigious Manhattan conservatory renowned for minting musicians such as Yo-Yo Ma and Itzhak Perlman
4	224		"Memoirs of a Geisha," features a romantic John Williams score _ with cello solos from Yo-Yo Ma and violin solos from Itzhak Perlman _
5	281	4	Stern was among the most recorded classical musicians in history, and played a major role in cultivating the careers of such musicians as Itzhak Perlman, Pinchas Zukerman and Yo-Yo Ma.

Table 3: The answers entered and their time of entry in seconds by Assessor 8 for question 67. Note that for answer 4, it is not assigned a nugget because the nugget that it contains (nugget 3, see Table 2 for all nuggets), has already been found in answer 1. Figure 1 shows the corresponding recall curve for these answers.

swers for each question, their set sizes were limited by time (10 minutes), while the QA systems' sets were limited by text length (7000 characters). To be able to fairly compare human performance with an interactive document retrieval system with ranked answers from a QA system, we replace the F-measure with a modified version of the the recall curves of Lin [10, 11]. Lin proposed the plotting of recall versus response length with the response length being a surrogate for time. Better systems will have curves that rise faster and higher (greater recall) than the curves of the worse systems.

For our recall curves, we explicitly measure recall versus time. For our interactive document retrieval system, we simply recorded the time the assessors saved answers. As an example, Table 3 shows the times at which Assessor 8 recorded answers to question 67.

To compare a QA system to the document retrieval system, we estimate the rate at which users find nuggets based on the rate at which they would read a QA system's answers.

We picked a reading speed of 225 words per minute based on existing studies. People read at different speeds for different tasks such as reading for comprehension vs. scanning or skimming [2]. Muter and Maurutto [14] conducted two experiments and found reading speeds on a CRT monitor ranging from 199 in the first experiments to 251 words per minute in the second. In each experiment, the test subjects knew that they had to answer questions about the material after reading it and thus needed to read for comprehension. Dillon et al. [4] studied the effect of Microsoft's Cleartype font enhancement technology. Dillon et al. had people read approximately 2000 words and found reading speeds of 207 words per minute for "regular" font display and 219 words per minute for Cleartype. Hewitt et al. [8] found that people read at a rate of 238 words per minute with a range of 1.79 and 6.39 words per second (107-383 wpm). It seems clear that the average reading speed on screen falls somewhere between 200 and 250 words per minute. The average rate of the above reported speeds (199, 251, 207, 219, 238) comes to 223 words per minute. We rounded 223 up to 225 as a reasonable rate given the existing literature.

To compute the number of words in an answer, we carefully processed the text answers and tried to avoid inflating the word count. We first converted numbers into a single word to avoid bad breaks on punctuation. We next compressed uppercase acronyms and abbreviations such as U.S.A. to USA. We then converted parentheses, dashes, periods, semicolons, commas, and question and exclamation marks into spaces and deleted double quotes and back ticks and apostrophes. Finally, we broke the answer into whitespace separated words.

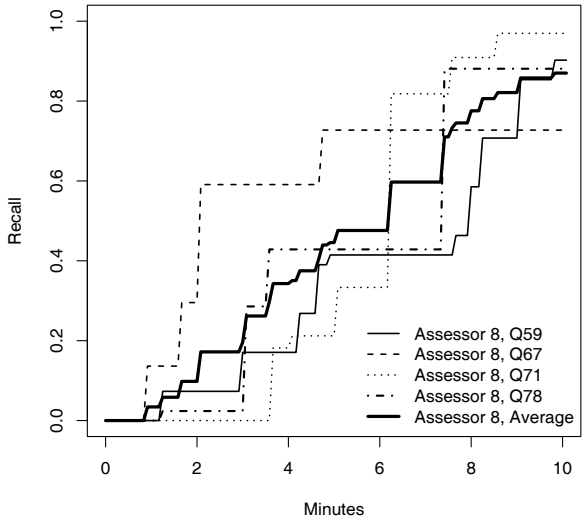

Figure 1: Assessor 8's recall curves for 4 questions, and the average of these 4 curves. Table 3 shows the corresponding answers for question 67, and Table 2 shows the nuggets.

We only compute recall at answer boundaries. Because most answers are short, we believe this is an okay, conservative approximation to when the user finds the nuggets in an answer. The ciQA track defined a measure of precision based on a text allowance of 100 non-whitespace characters per nugget returned. We do not report precision. The recall curve demonstrates the rate at which nuggets are discovered by the user, which is the equivalent of precision.

Recall curves are relatively simple to compute. Each answer takes a certain amount of time to either be entered by the user of the document retrieval system or to be read by the user of a QA system. Each answer has either zero or more nuggets and thus recall increases monotonically with answers. To produce an average curve, one averages the recall of all curves at a given point in time. We have computed the curves at 5 second intervals.

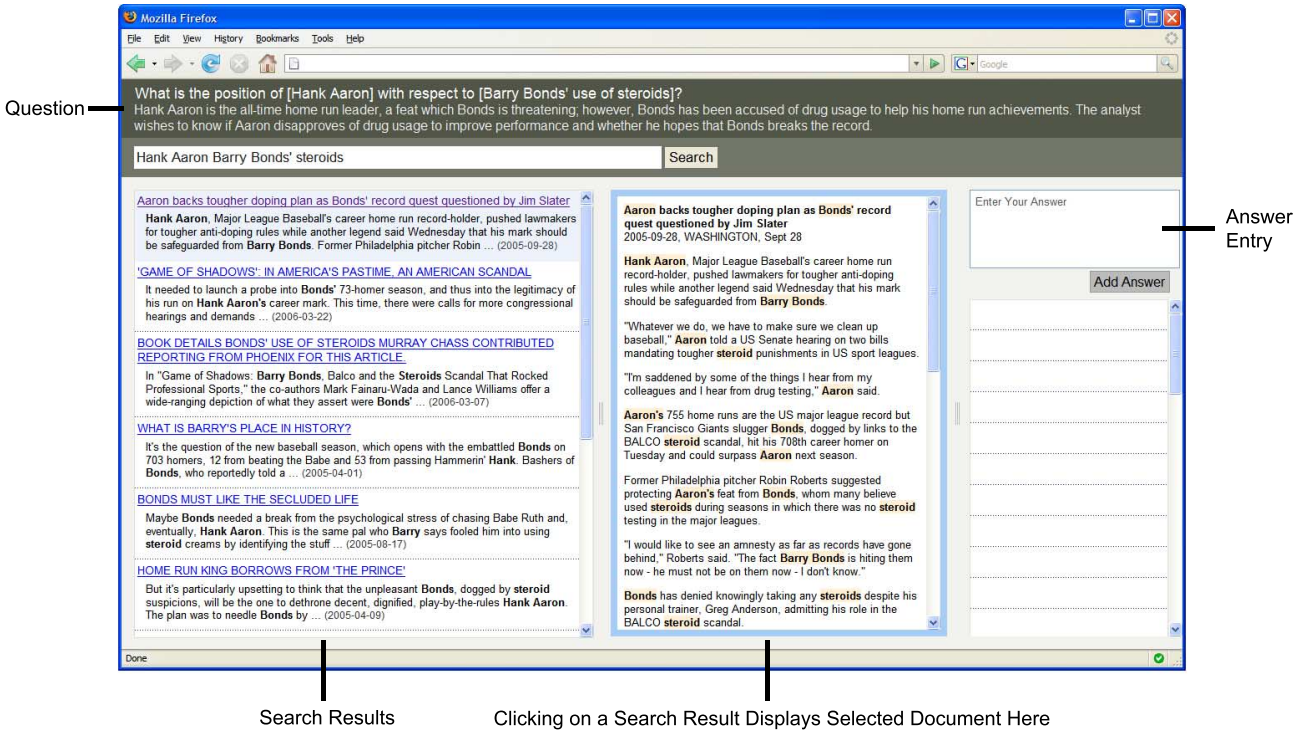

Figure 2: A screenshot of the web-based interface for our fully interactive, IR system.

As an example, Figure 1 shows the recall curves for Assessor 8. In particular, we will look at question 67 in this example. The curve for question 67 increases at the points in time when Assessor 8 saved answers containing novel nuggets, which are given in Table 3. At 51.6 seconds, nugget 3 is saved as part of answer 1. Nugget 3 has a vital score of 0.75, and the vital sum for question 67 is 5.5. Thus, at 51.6 seconds, Assessor 8's recall is $0.75/5.5 = 0.14$. The assessor saves answer 2 at 97.4 seconds. Answer 2 contains nugget 5, which has a vital score of 0.875, and thus recall increases to $(0.875+0.75)/5.5 = 0.30$. When Assessor 8 records answer 3 at 121 seconds, the Assessor's recall takes a large jump that reflects the simultaneous discovery of two nuggets. At 224 seconds (3.7 minutes), Assessor 8 records answer 4, but the recall curve does not rise at that time because answer 4 has no novel nuggets. The growth in the recall curve for question 67 ends with a recall of 0.73 with the entry of answer 5 at 281 seconds (4.7 minutes).

3.4 Our Experimental System

We built a fully interactive IR system with facilities for recording answers to questions. Figure 2 shows the interface to our IR system.

At the top of the interface, we presented the question and a search textbox. For the question, we presented both the templated version and the expanded narrative. To the far right of the search box, we provided a timer (not shown in Figure 2) that counted down from 5 minutes in minute increments for the first 4 minutes and then showed remaining time in seconds for the last minute.

The area below the question and search box consisted of three vertically oriented panes. The left pane showed search results. Each result displayed the document's title, a query-biased snippet with term highlighting, and the date of publication. The user could click on a link at the end of the results to have the next 10 results added to the list. 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. The document display highlighted query terms and showed each document cleanly divided into paragraphs. The right hand pane provided a textbox 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. Users could go back to a source document by clicking on a saved answer and could also delete saved answers. Users could adjust the size of the three panes by clicking and dragging a "grippie" widget located between adjoining panes.

Our web-based, front-end client was a modern AJAX-like interface. 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. This behavior is in contrast to the majority of web search engines that require users to transition between a page of results and web pages.

When the assessor first accessed the system for a given question, the system showed 10 results for a default query created automatically from the templated question as shown in Figure 2. 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. For example, the question "What is the position of [Hank Aaron] with respect to [Barry Bonds' use of steroids]?" resulted in the query "Hank Aaron Barry Bonds' steroids."

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. For retrieval, we used Indri [16]. The Indri query language provides support for both of these query language features. We automatically transformed users' queries into well formed Indri queries.

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 or after hitting the refresh button on their web browser.

3.4.1 Implementation Details

We wrote the front-end client using XHTML, CSS, and JavaScript. We built the back-end server using a combination of the Apache web server, PHP, mySQL, Perl, C++, and the Indri [16] retrieval system.

We annotated the sentences in the AQUAINT2 collection using a locally modified version of a sentence splitter [13]. We stemmed all words with the Porter stemmer built into Indri and used an in-house list of 418 stop words. We used Indri's default parameters, which includes setting the Dirichlet prior smoothing parameter to a value of 2500.

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 trimmed the snippet to have a maximum length of 35 words.

3.5 Experimental Setup

We utilized the 8 NIST assessors to search for and save answers to their questions. As already described, we supplied a fully interactive IR system for the assessors to use to find answers. We submitted the answers saved by the assessors with no modification as one of our ciQA runs.

We provided a detailed tutorial that each assessor was to read before using our system. In the tutorial, we motivated and explained our system to the assessors as follows:

> We have a novel approach to the ciQA task. Our belief is that human searchers, such as yourself, can find answers faster and more accurately than computers. Given our search system, our hope is that you can quickly find answers to the questions.
>
> We have constructed a system that allows you to search for documents that will help you answer the question. When you find an answer, you will enter and save the answer using the system.

We then explained how the system worked including the basic parts of the interface, the auto-generated default query, and the query language. We provided example usage and ended with a chance for the assessors to practice using the system with the throwaway question provided by NIST. We did not explain to the assessors that they could delete answers, but left that as a discoverable feature.

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. We did not worry about the assessors entering memorized answers because of the complex nature of the questions, which would be a concern for single answer, factoid questions.

The ciQA track allocates two sets of interactions with the assessors. Each interaction set gets at most 5 minutes of interaction for each question. The assessor who generated the question both does the interaction and the judging of the question.

We used both of our allotted 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. While this strategy was suboptimal, we wanted to see how user performance improved over a time period greater than 5 minutes. We submitted to NIST the full 10 minutes of interaction as one run.

Because assessors could start with either run, we provided the same tutorial for each run. At the top of the tutorial, we explained why the users would view the instructions a second time.

The assessors interacted with the system during a 2 day period. On the first day, some assessors experienced network slowdowns, which likely hurt their ability to find and record answers.

3.6 Automatic QA Systems

We compare usage of our document retrieval system to the baseline QA runs, i.e. the QA runs that do not utilize any interaction with the assessors. If we used the post-interaction runs, we would need information regarding the amount of time the assessors interacted with the systems. We do not have this timing information and compare only to the baselines. Of note, the post-interaction runs showed little gain over the baselines [3].

The same assessors who used our interactive document retrieval system to answer their questions are the same assessors that judged the answers returned by the automatic QA systems.

Seven different groups including the track organizers submitted a total of 12 automatic baseline runs to ciQA 2007. The majority of the systems could be described as following a sentence retrieval procedure whereby they first retrieved the top 20-100 documents and then reranked the sentences within these documents. While ciQA was a question answering track, the retrieval techniques seemed more akin to focused retrieval than to traditional question answering.

The track organizers created a sentence retrieval baseline, BaseA [3]. Of these baseline runs, we restrict our comparisons to one run from each of the top two performing groups(RUN-24-3, RUN-26-4), and BaseA. RUN-24-3 and RUN-26-4 are the anonymous names assigned by NIST. We will refer to RUN-24-3 and Run-26-4 by the simpler names Run-1 and Run-2 respectively.

3.7 Data Cleaning

For the results in this paper, we have excluded the four questions completed by Assessor 5, for we do not believe Assessor 5 understood the assigned task. On the exit questionnaire, Assessor 5 wrote of our IR system: "I guess I never really understood how this one was supposed to work." In addition, Assessor 5's performance was far from the norm of the other 7 assessors. Assessor 5 only entered one an-

swer on one question and the assessor judged that answer to contain no nuggets. The other assessors entered an average of 8.0 answers per question. We believe that Assessor's 5 performance is more a reflection of the assessor's lack of understanding the task rather than a reflection on the assessor's ability to use an interactive IR-system. One possible cause of Assessor 5's performance is that during Assessor 5's initial usage of the system, we experienced a serious network slowdown that likely affected the assessor's perception of what was possible with our system. Overall, we feel that the results are more accurate with Assessor 5's questions removed.

For one question, we recorded an answer from an assessor at a time of 603 seconds — 3 seconds over the 10 minutes of allowed interaction time. We have included this "extra" answer in our results.

4. RESULTS AND DISCUSSION

In this section, we first describe the question answering performance of the assessors using the document retrieval system, and then we compare their performance to the automatic QA systems.

4.1 Human Question Answering Performance

Figure 3 shows the average recall curves for the assessors on the left and shows the overall average recall curve on the right. Table 4 reports various statistics for the assessors.

As we can see in both Figure 3 and Table 4, there is a wide variation in performance between assessors. The assessors group roughly into 3 groups based on performance. Assessor 8 by far found nuggets at a much faster rate than the other assessors and achieved an average recall of 0.870 in 10 minutes. The next best performing group consists of assessors 4 and 6, who averaged around 0.5 recall in 10 minutes. The last group of assessors consists of the remaining 4 assessors who have recall averaging around 0.2 recall in 10 minutes.

Interestingly, while the assessors differ in performance, they all appear to roughly have recall curves that are linear. The linear behavior is most visible in the average curve shown in the right plot of Figure 3. On average, the assessors reached 0.4 recall in 10 minutes. While the rate of finding nuggets is unlikely to stay linear with time as nuggets become rarer and more difficult to find, at this rate, the average assessor would find all nuggets in 25 minutes. The slower assessors working at a rate of about 0.2 recall per 10 minutes would likewise require 50 minutes to find all nuggets for their questions. The assessors are able to find answers to their questions with a document retrieval system. If the average recall curves showed significant leveling off within 10 minutes, this would be a sign that the assessors were having difficulty finding answers to their questions.

While the overall recall curves are linear, the assessors do take longer to enter their first answers. The assessors took 118 seconds on average to enter their first answer and spent on average 79.3 seconds between entering answers. Assessor 8 worked quickly and spent only 27.3 seconds between each entered answer. Examining the logs shows that Assessor 8 often extracted several answers from a single document.

We would expect that the assessors would provide answers that had at least one nugget per answer, but the average number of nuggets per answer was 0.6. One possible reason the number of nuggets per answer is less than 1 is that some answers contained duplicate nuggets, and the assessors did not carefully delete duplicate answers. Another possible reason is that assessors have difficulty judging answers for nuggets. For example, on question 76, the assessor typed in 4 answers, but the assessor found no nuggets in any of the answers even though 3 of the 4 answers appear to us to be good and contain nuggets.

As mentioned in Section 3, the ciQA track had a definition of precision that allotted 100 non-whitespace characters per nugget returned. While we did not use this definition of precision, the definition is of interest because it establishes a sense of what researchers have thought would be a reasonable amount of text per nugget.

As Table 4 shows, 5 of the 7 assessors all entered answers that on average were greater than 100 characters long. Three assessors had average answer lengths of greater than 200 characters. The assessors with short answers performed worse on average than those with longer answers. For assessors 3 and 7, short answers were the result of the assessor typing in a summarized answer rather than copying text directly from a source document.

When answers are text excerpts, the 100 character allowance is likely too small. For example, on question 84, the assessor covered 11 of 14 possible nuggets in only 6 answers. The assessor's found nuggets had a total vital score of 8 out of 9.9 possible. One longer answer contained 5 nuggets. On this question, the assessor achieved a recall of 0.809 with a total response length of 1891 non-whitespace characters. Given an allowance of 100 non-whitespace characters per nugget, the assessor was 72% over the allowance of 1100 characters.

When we look at the average number of nuggets found per answer, the 100 character allowance also looks too small. The assessors entered 255 characters per nugget (the assessors' average of 152.7 characters per answer divided by the average 0.6 nuggets per answer). Even if we assume the assessors meant to have at least one nugget per answer, 152.7 characters is over 50% more than the presumed reasonable allowance of 100 characters.

If we expect QA systems to return carefully summarized answers, then Assessor 7 provides a good example of summarized answers on question 73. Here the assessor typed by hand 4 answers totaling 196 non-whitespace characters. When the assessor judged these answers, the assessor found 3 nuggets for an average of 65 characters per nugget.

For each question, we recorded the maximum rank document that the assessor viewed. The average maximum rank viewed for each assessor is shown in Table 4. Most of the assessors did not go very deep in the ranked results. Both the better and poorer performing assessors had explorations with a maximum rank viewed of about 8.

The assessors' shallow explorations were not a result of issuing many queries. Only two assessors on a total of three questions did any query reformulation. Even though our tutorial encouraged assessors to modify the default query, they may have been confused about their ability to query the system given the default results and an interface flaw that disabled the search button unless the query was changed.

Assessor 6 reformulated two queries and each time did so by adding additional words. Assessor 4 deleted the misleading query terms "financial relationships" for question 68, which looked for a connection between DARPA and BBN. Assessor 4 attempted to force the appearance of both DARPA and BBN by joining them with an ampersand. To

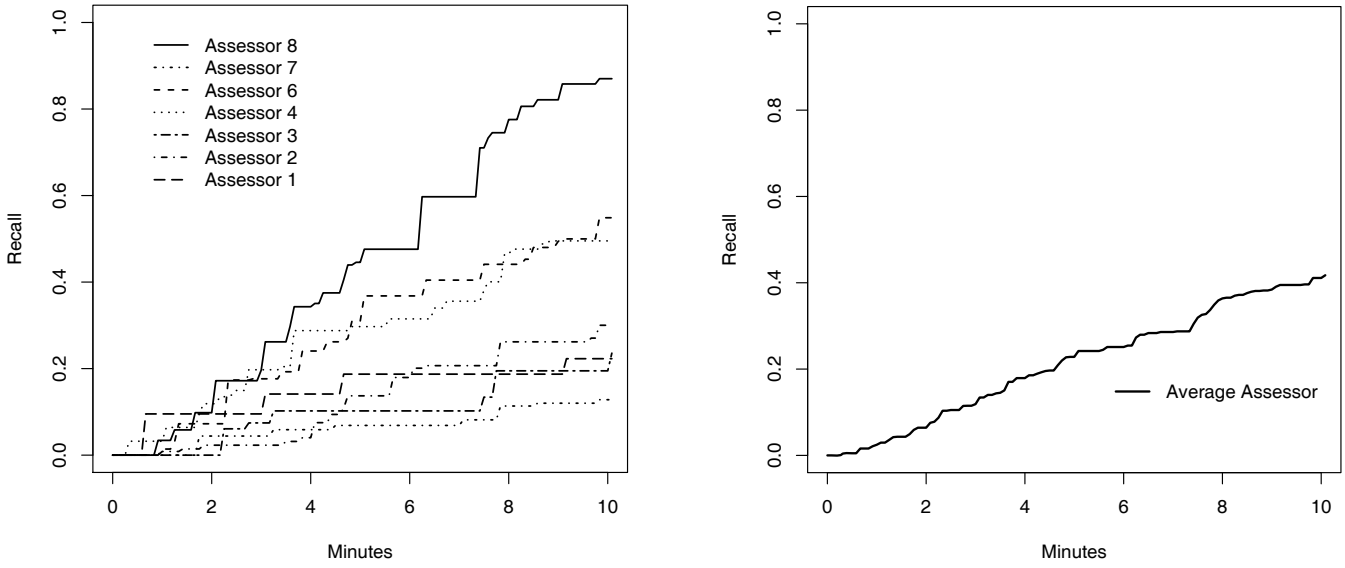

Figure 3: The plot on the left shows the assessors' average recall versus time using the document retrieval system. The plot on the right shows the overall average assessor recall curve.

	Assessors Ordered by Recall							Assessors
Mean Statistic	A7	A1	A3	A2	A4	A6	A8	Overall
Recall at 10 minutes	0.128	0.223	0.236	0.300	0.495	0.549	0.870	0.400
Answer Length (non-whitespace chars.)	51.2	238.8	59.8	202.6	115.8	237.7	162.7	152.7
Answers per Question	7.0	3.3	2.5	8.5	12.5	6.5	15.5	8.0
Nuggets per Answer	0.3	0.5	0.6	0.6	0.5	1.2	0.5	0.6
Time to First Answer (seconds)	84.7	149.8	155.9	121.0	100.7	123.8	90.3	118.0
Time between Answers (seconds)	67.1	84.1	201.8	55.4	39.2	79.9	27.3	79.3
Max. Document Rank Viewed	8.0	4.3	17.8	30.8	18.5	8.8	7.5	13.7

Table 4: Per assessor performance and statistics.

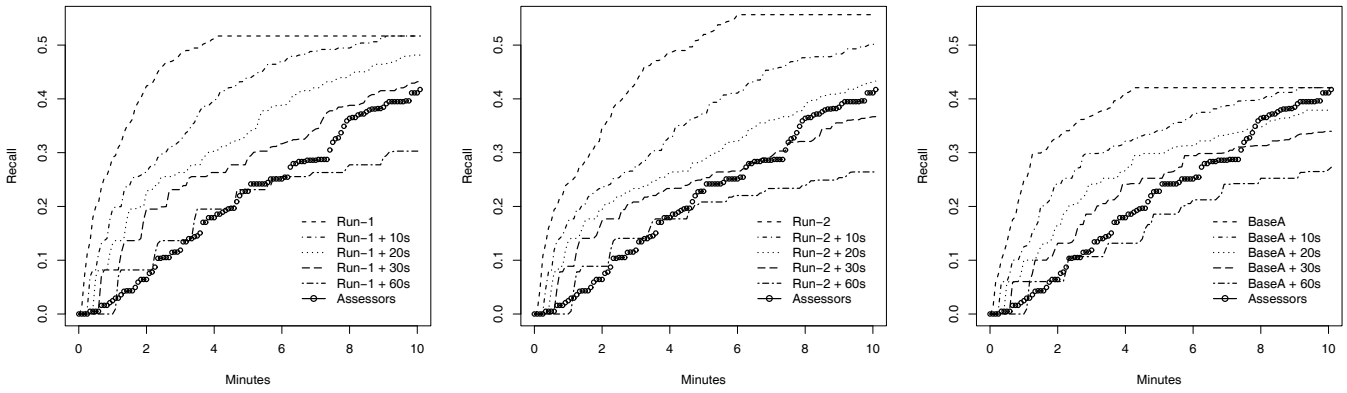

Figure 4: Average recall versus time for three QA systems versus the assessors' average performance with the document retrieval system. The far left plot shows Run-1, the middle plot shows Run-2, and the far right plot shows BaseA. For each plot, the QA system's recall vs. time is plotted with a reading rate of 225 wpm and also at rates that include 10, 20, 30, and 60 seconds additional time spent per returned answer.

achieve the desired Boolean AND, the assessor needed to prefix each query term with a plus sign.

4.2 Comparison to Automatic QA Systems

Figure 4 shows the recall curves for three QA systems compared to the assessors' performance with the document retrieval system. For each QA system, we have simulated its recall versus time by assuming answers are read at a rate of 225 words per minute (wpm) as described in Section 3. In addition to showing the recall curves obtained from merely reading answers, we also show the recall curves for the QA systems if the simulated user has to spend an additional 10, 20, 30, or 60 seconds per answer. This additional time would possibly be required by the user to understand the context of the answer in the larger document from which it comes. (In ciQA, each answer has to be associated with a single document.) Other causes of additional time could be related to various interactions with a QA system to record which returned answers are good and should be saved.

The recall curves show that while Run-1 (far left) initially has a faster rate of recall than Run-2 (middle), Run-2 achieves a greater recall. The sentence retrieval baseline, BaseA (far right), ends up with a recall the same as achieved in 10 minutes by the assessors with the document retrieval system.

In terms of performance at 10 minutes, Run-1 can better the document retrieval system even with 30 seconds spent per answer in addition to the time required to read the answer. Run-2's performance though only provides the user with about 20 seconds in "spare" time per answer. BaseA can only spare about 10 seconds per answer. If we look at performance at times less than 10 minutes, we see that the QA systems have more spare time per answer.

The QA systems' precision decreases with increasing rank of their answers. The more answers that a user needs, the less spare time is available per answer when using a QA system. For users requiring high recall, there may not be a significant advantage to using QA systems, but at lower recall, and correspondingly shorter sessions times, these QA systems appear to be significantly superior to the document retrieval system.

While it can take a long time for humans to extract answers from documents, humans can do so with human levels of performance. For example, question 68 was one of the questions were human involvement made a huge difference in performance. For this question, all of the answers that the assessor found lacked the term "DARPA" but instead made mentions to "the agency" or "the pentagon agency."

4.3 Additional Observations

Beyond being an excellent searcher, Assessor 8 was clearly enthusiastic about our task. In the exit questionnaire, Assessor 8 wrote that our system "was my favorite exercise - it was sort of like doing research on a subject and then trying to put the information in the proper order." Assessor 8 saw our task as building "his/her own article from the texts given..."

Not all assessors agreed with Assessor 8. Assessor 6, who did very well, wrote "It took a while to understand what this was all about. I felt that I was doing the exact same procedure I used to pose the original topic query! I originally used search terms, looked at documents, and copied/pasted some juicy answers. Now with this form I have successfully redone what I did before!!!!!"

Assessor 6's feedback raises the point that the assessors have already researched their questions as part of the question development process. This familiarity should only boost the assessors' performance with the document retrieval system and strengthens our results showing that today's QA systems have the potential to help users find answers faster than document retrieval systems.

A couple assessors wrote that they were either confused or upset that they came back to our same system twice. Our explanation in our tutorial either did not make sense or was not noticed.

We failed to provide a link to our tutorial instructions from within our system. While simple, and something we considered, we left it out assuming assessors would devote adequate time to the tutorial to learn what was required of them. It certainly appears that our tutorial did not convey to all the users what was expected. In the questionnaire, several assessors asked for interface instructions to always be available.

5. CONCLUSION

We measured the performance of users using a document retrieval system to answer complex questions and compared their performance to simulated users of question answering (QA) systems. The users of the document retrieval system were NIST assessors. While the assessors were able to successfully use the document retrieval system to find answers to their questions, if the assessors did not have high recall needs, they would be able to find answers much faster using a QA system. As the recall needs of the assessors increases, the performance advantage of the QA systems decreases for with increasing rank of answer, the QA systems' precision drops and increasing amounts of time is wasted skipping answers that do not contain nuggets of information.

6. ACKNOWLEDGMENTS

We thank Hoa Trang Dang, Diane Kelly, Jimmy Lin, and the NIST assessors for making the TREC ciQA track possible. We especially thank Hoa for helpfully answering our many questions regarding ciQA details.

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, in part by Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the University of Waterloo. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsors.

7. REFERENCES

[1] J. Allan. HARD track overview in TREC 2005: High accuracy retrieval from documents. In *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings.* Department of Commerce, National Institute of Standards and Technology, 2005.

[2] A. Chapman. *Making Sense: Teaching Critical Reading across the Curriculum.* College Board Publications, 1993.

[3] H. T. Dang, D. Kelly, and J. Lin. Overview of the TREC 2007 question answering track. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. Department of Commerce, National Institute of Standards and Technology, 2007.

[4] A. Dillon, L. Kleinman, G. O. Choi, and R. Bias. Visual search and reading tasks using cleartype and regular displays: two experiments. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 503–511. ACM, 2006.

[5] S. T. Dumais and N. J. Belkin. The TREC interactive tracks: Putting the user into search. In *TREC*, chapter 6, pages 123–152. MIT Press, 2005.

[6] G. Erbach. Evaluating human question answering performance under time constraints. QA@CLEF-2004, `http://www.mcgreg.net/pub/human-qa/`, August 2004.

[7] W. Hersh and P. Over. The TREC-9 interactive track report. In *The Ninth Text REtrieval Conference (TREC-9)*, pages 41–50. Department of Commerce, National Institute of Standards and Technology, 2000.

[8] J. Hewitt, C. Brett, and V. Peters. Scan rate: A new metric for the analysis of reading behaviors in asynchronous computer conferencing environments. *American Journal of Distance Education*, 21(4):215–231, 2007.

[9] J. Kamps, S. Geva, and A. Trotman. Report on the SIGIR 2008 workshop on focused retrieval. *SIGIR Forum*, 42(2):59–65, 2008.

[10] 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, 2007.

[11] J. Lin. User simulations for evaluating answers to question series. *Information Processing and Management*, 43(3):717–729, 2007.

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

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

[14] P. Muter and P. Maurutto. Reading and skimming from computer screens and books: the paperless office revisited? *Behaviour & Information Technology*, 10(4):257–266, 1991.

[15] M. D. Smucker, J. Allan, and B. Dachev. UMass Complex Interactive Question Answering (ciQA) 2007: Human Performance as Question Answerers. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*. Department of Commerce, National Institute of Standards and Technology, 2007.

[16] 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, CS Dept., U. of Mass. Amherst, 2005.

[17] E. Voorhees. The TREC-8 question answering track evaluation. In *The Eighth Text REtrieval Conference (TREC-8)*, pages 83–106. Department of Commerce, National Institute of Standards and Technology, 1999.

[18] E. M. Voorhees. Question answering in TREC. In *TREC*, chapter 10, pages 233–257. MIT Press, 2005.

[19] M. Wu, G. Muresan, A. McLean, M.-C. M. Tang, R. Wilkinson, Y. Li, H.-J. Lee, and N. J. Belkin. Human versus machine in the topic distillation task. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, pages 385–392. ACM, 2004.

[20] Y. Xu and D. Mease. Evaluating web search using task completion time. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 676–677, New York, NY, USA, 2009. ACM.