Human Question Answering Performance using an Interactive Information Retrieval System

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 find documents that answer their questions. Compared to these IR 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 IR systems compare to QA systems, we measured the performance of humans using an interactive IR system to answer questions. We conducted our experiments within the framework of the TREC 2007 complex, interactive question answering (ciQA) track. We found that the average QA system was comparable to humans using an IR system. Our results also show that for some users IR systems can be powerful question answering systems. After only 5 minutes of usage per question, one user of the IR system obtained an average F ($\beta = 3$) score of 0.800, which outperformed the best QA system by 27% and the average QA system by 40%. After 10 minutes of usage, 5 of 8 users of the IR system obtained a higher performance than the average QA system. To achieve superior performance, future QA systems should combine the flexibility and precision of IR systems with the ease-of-use and recall advantages of QA systems.

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

Today when users have questions, one of their likely tactics for finding answers is to use an information retrieval (IR) system. In 2005, an estimated 60 million U.S. adults used a web search engine on a typical day [10].

In many respects, traditional IR systems represent the most basic of question answering (QA) systems. A user must first transform a question into a query suitable for the IR system. The IR 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.

Question answering systems aim to automate these search tasks. Users are encouraged to enter their questions as questions. The QA system handles all searching. The user only has to evaluate the quality of the produced answers.

A good IR system should inherently be a good question answering system, and popular usage of search engines implies that IR systems are good QA systems. Nevertheless, 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 looks at complex information needs and aims 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) generate questions, interact with systems, and judge 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.

At our URL, we provided the assessors with a fully interactive, document retrieval system. Figure 1 shows the interface, which we describe in detail in Section 2.3. 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 assessors then judged the answers from all submitted systems. As such, the assessors judged their own answers. Our experiment allowed us to measure the assessors' performance at answering their questions using an IR system and compare this performance to the other participants' interactive QA systems. We found that:

- Interactive IR systems are competitive with automatic QA systems for users with complex information needs. The QA systems had a slight advantage in recall of answers, and as expected, the assessors had better precision using the IR system (Section 5).
- The performance of the assessors was variable. Some assessors using the IR system outperformed the QA systems. Other assessors would be better served by an automated QA system (Section 5.2).

- Although better than the QA systems, the assessors averaged a surprisingly low precision of 0.427. A possible reason for this low precision is that assessors entered answers longer than the allowance of 100 non-whitespace characters per nugget (Section 5.3).

This work extends our 2007 TREC paper [Blinded for Review] with additional experiments, results, and analyses. We next describe our methods and materials, the details of our experiments, and finally present and discuss our results.

2. METHODS AND MATERIALS

We conducted our experiments within the framework provided by the 2007 TREC complex, interactive question answering (ciQA) track [3]. The ciQA track's goals are 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. We did not directly attempt to address how to build an interactive question answering system. Rather than build question answering systems, our goal is to build interactive IR systems that better enable people to answer questions.

2.1 ciQA Track Details

The ciQA track follows the same three step process of its predecessor, the HARD track [1]: submit baselines, interact with assessors, submit final runs. Sites create and submit 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 has the opportunity to have two sets of interactions with the assessors. For each set, the site has the chance to interact with an assessor for each question for a maximum of 5 minutes. Using these sets of interaction, sites then prepare their final submissions. In 2007, sites were allowed to submit two baselines and two post-interaction runs, which typically correspond to the two interaction sets. Sites can submit both manual and automatic runs. Manual runs involve some form of human intervention by the site. An example of a manual run would be for a site to hand craft queries.

2.1.1 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, which we ignored and did not utilize. 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.

2.1.2 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. For 2007, 4 of the 14 participating systems appeared to be more interactive than the static HTML forms used in previous years. 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.

Template 2, Question 64: What [common interests] exist between [President Bush] and [Bono, the U2 Rock Star]? Narrative: The analyst is interested in knowing the subject or subjects in which two such disparate people could find common cause, and primarily what effect/actions they singly or mutually had accomplished in their field or fields of interest.
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 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.

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

2.1.3 Evaluation

The ciQA track uses a nugget-based evaluation. Each run may return as many answers to each question as desired up to a 7000 non-whitespace character limit. Assessors maintain a list of *nuggets*. A nugget represents a single, atomic answer. The assessors read each submitted answer and for each answer determine which nuggets, if any, exist in the answer. An answer may contain more than one nugget. Nuggets are only counted once, i.e. duplicate nugget mentions count as returning the nugget once.

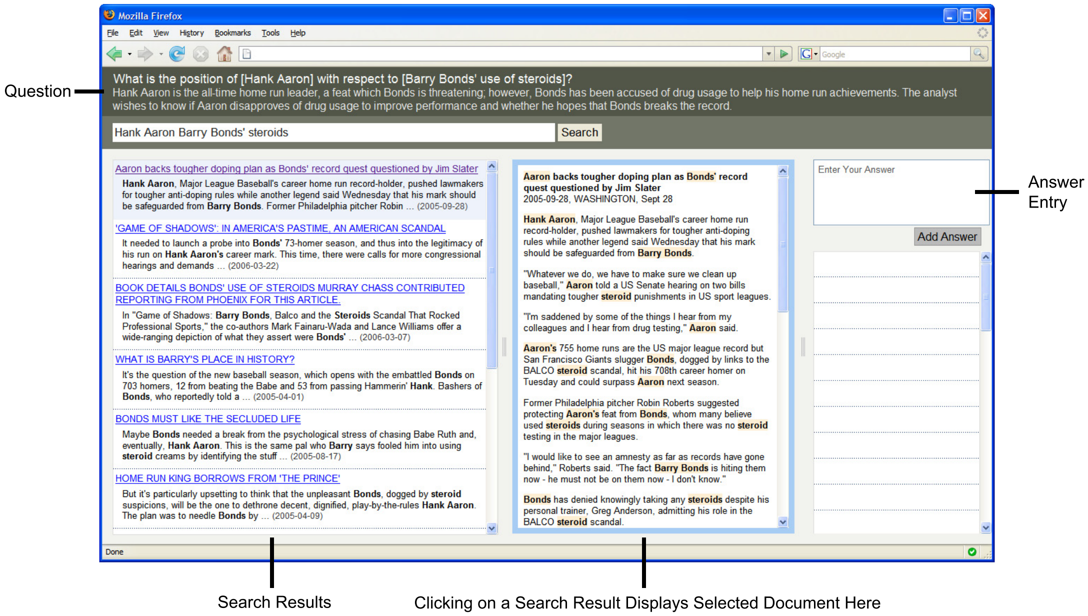
From the list of nuggets, NIST constructs a *nugget pyramid* [8]. The assessors judge each nugget as either being a *vital* or an *okay* nugget. The assessor in charge of the question, then judges nuggets one more time. The *vital score* of a nugget is the fraction of judgments that were vital. For example, if a nugget receives 1 judgment as vital and 8 okay judgments, then its vital score is 1/9.

For each question, recall and precision are calculated as follows. Recall is the sum of the vital scores of the returned nuggets divided by the sum of the vital scores for all known nuggets. For each nugget returned, a text allowance of 100 non-whitespace characters is granted. If the response text is shorter than the allowance, precision is 1. Otherwise precision is the allowance divided by the response length.

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. We also report the F measure with $\beta = 1$, which places equal importance on precision and recall. The F measure is:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

where P is precision and R is recall.


Mozilla Firefox
File Edit View History Bookmarks Tools Help
Google
Question
Search
Enter Your Answer
Add Answer
Answer Entry
Done
Search Results
Clicking on a Search Result Displays Selected Document Here

What is the position of [Hank Aaron] with respect to [Barry Bonds' use of steroids]?
Hank Aaron is the all-time home run leader, a feat which Bonds is threatening; however, Bonds has been accused of drug usage to help his home run achievements. The analyst wishes to know if Aaron disapproves of drug usage to improve performance and whether he hopes that Bonds breaks the record.

Hank Aaron Barry Bonds' steroids

Aaron backs tougher doping plan as Bonds' record quest questioned by Jim Slater
Hank Aaron, Major League Baseball's career home run record-holder, pushed lawmakers for tougher anti-doping rules while another legend said Wednesday that his mark should be safeguarded from **Barry Bonds**. Former Philadelphia pitcher Robin ... (2005-09-28)

'GAME OF SHADOWS': IN AMERICA'S PASTIME, AN AMERICAN SCANDAL
It needed to launch a probe into **Bonds'** 73-homer season, and thus into the legitimacy of his run on **Hank Aaron's** career mark. This time, there were calls for more congressional hearings and demands ... (2006-03-22)

BOOK DETAILS BONDS' USE OF STEROIDS MURRAY CHASS CONTRIBUTED REPORTING FROM PHOENIX FOR THIS ARTICLE.
In "Game of Shadows: **Barry Bonds**, Balco and the **Steroids** Scandal That Rocked Professional Sports," the co-authors Mark Fainaru-Wada and Lance Williams offer a wide-ranging depiction of what they assert were **Bonds'** ... (2006-03-07)

WHAT IS BARRY'S PLACE IN HISTORY?
It's the question of the new baseball season, which opens with the embattled **Bonds** on 703 homers, 12 from beating the Babe and 53 from passing Hammerin' **Hank**. Bashers of **Bonds**, who reportedly told a ... (2005-04-01)

BONDS MUST LIKE THE SECLUDED LIFE
Maybe **Bonds** needed a break from the psychological stress of chasing Babe Ruth and, eventually, **Hank Aaron**. This is the same pal who **Barry** says fooled him into using **steroid** creams by identifying the stuff ... (2005-08-17)

HOME RUN KING BORROWS FROM 'THE PRINCE'
But it's particularly upsetting to think that the unpleasant **Bonds**, dogged by **steroid** suspicions, will be the one to dethrone decent, dignified, play-by-the-rules **Hank Aaron**. The plan was to needle **Bonds** by ... (2005-04-09)

Aaron backs tougher doping plan as Bonds' record quest questioned by Jim Slater
2005-09-28, WASHINGTON, Sept 28

Hank Aaron, Major League Baseball's career home run record-holder, pushed lawmakers for tougher anti-doping rules while another legend said Wednesday that his mark should be safeguarded from **Barry Bonds**.

"Whatever we do, we have to make sure we clean up baseball," **Aaron** told a US Senate hearing on two bills mandating tougher **steroid** punishments in US sport leagues.

"I'm saddened by some of the things I hear from my colleagues and I hear from drug testing," **Aaron** said.

Aaron's 755 home runs are the US major league record but San Francisco Giants slugger **Bonds**, dogged by links to the BALCO **steroid** scandal, hit his 708th career homer on Tuesday and could surpass **Aaron** next season.

Former Philadelphia pitcher Robin Roberts suggested protecting **Aaron's** feat from **Bonds**, whom many believe used **steroids** during seasons in which there was no **steroid** testing in the major leagues.

"I would like to see an amnesty as far as records have gone behind," Roberts said. "The fact **Barry Bonds** is hiting them now - he must not be on them now - I don't know."

Bonds has denied knowingly taking any **steroids** despite his personal trainer, Greg Anderson, admitting his role in the BALCO **steroid** scandal.

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

2.2 Automatic Evaluation via Nuggeteer

Unlike document retrieval where a set of relevance judgments is easily reused, judging whether or not an answer contains a nugget has traditionally been a human task. Nuggeteer [9] is a program that provides automatic judging of answers. We used version 0.8 of Nuggeteer to judge experiments that we conducted post-ciQA.

Nuggeteer works by building a model of each nugget given the assessor's description of the nugget and all answers judged to contain the nugget. Once trained, Nuggeteer makes binary decisions as to whether or not an answer contains a nugget. The advantage of Nuggeteer over other automatic approaches is that it allows scoring that is comparable to the human judged scores.

The authors of Nuggeteer trained on the ciQA 2007 track data and released settings that they believe offer the best performance. The ciQA 2007 settings are: term_weighting = count, stopwords = remove, ngrams = 2, stem = 1, beta = 3, max_judgement_rank = 0, lm_from_judgements = 0, and the decision method is a fixed threshold = 0.07.

2.3 Our Experimental System

We built a fully interactive IR system with facilities for recording answers to questions. Figure 1 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 1) 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.

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 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. 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 [11]. The Indri query language provides support for both of these query language features. We automatically transformed users' queries into well formed Indri queries.

2.3.1 Implementation Details

Our web-based, front-end client was a modern AJAX-like interface written in 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 [11] retrieval system.

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.

We annotated the sentences in the AQUAINT2 collection using a locally modified version of a sentence splitter [2]. 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. EXPERIMENTS

We ran four experiments that differed in the choice of human subjects and the time constraints the subjects faced to complete the task. Our first experiment was a baseline involving no interaction. For our other experiments, our human subjects were either the NIST assessors, an expert searcher, or non-expert users. The NIST assessors had a tight limit on their task time while the other users did not.

3.1 Baseline

We constructed our baseline to be similar to the displayed query-biased snippets for each question's default query. As described in Section 2.3, 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 assessors. 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.

3.2 Assessors

This experiment utilized the 8 NIST assessors to search for and save answers to their questions. As described in the previous sections, 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:

> 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.

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.

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. 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.

3.3 Search Expert

A concern with interactive systems is that users have difficulty in judging recall and often quit searching too soon. We also felt that the assessors would not have enough interaction time to obtain a sense of the maximum performance. To address both of these concerns, we had an expert user (one of the authors) use the system for an unlimited time to find as many answers for each question as possible. We submitted these answers as another run to NIST for judging. We refer to this user as "User X."

The expert user worked on the questions in numeric order and stopped working on each question once he felt that he could not find any further answers. The longest the user worked on a question was 22 minutes and 10 seconds. For a few questions where the user found over 7000 characters of answers, the user edited the answers to fit within the limit. For our automatic analyses using Nuggeteer, we analyzed the original set of unedited answers to better reflect the actual behavior of the expert. Without the careful editing, the expert user's performance was slightly worse when judged by

Nuggeteer. Little care was taken to prevent entry of duplicate answers. Most answers are in the form of snippets of text extracted directly from the source documents.

3.4 Non-Expert Users

To further investigate human performance, we had 3 non-expert users answer the questions using the same IR system. These users are undergraduates who are paid to perform various annotation tasks for our research group. We presented this task to the users as though it was another annotation task for which they would be paid their usual hourly rate. To provide some additional motivation, we informed them that we would give a prize worth approximately $20 to user with the highest score. We refer to these users as "User A", "User B", and "User C."

The users were shown a slightly modified version of the same tutorial that the NIST assessors used. The users then practiced using the system with questions from the 2006 ciQA track. For one of the questions, the users worked alone. When done with the question, the users reviewed the answers with the researcher supervising the task. Next the users practiced on another question while the researcher observed their usage and discussed answer choices being made. This is our standard practice with annotators to make sure they understand the task.

The users could spend up to 30 minutes on each question. If the users felt that all answers had been found, they could stop working on a question early. Each user did the questions in a different random order.

We gave the users a worksheet to record their start and end times as well as to note any breaks taken. We encouraged them to not take breaks, but preferred for them to take a break and note it rather than to "let the clock run" on the task. Only one user took breaks. We removed the breaks in time from the logs. In some cases the users snuck a look at the next question and then clearly decided to work on it later. In these cases, we set the question's start time at the start of the actual session.

4. AUTOMATIC QA SYSTEMS

We will compare our experimental results to the performance of 8 automatic runs submitted by 5 different sites and NIST. These 8 runs are the final submissions that had been marked by participants as automatic runs. We dropped the worst performing automatic run for it had an F measure that was more than 50% less than the other systems. We included NIST's post-interaction "baseline" run as one of the 8 runs. This run represents the performance of an IR system that returns top ranked sentences but which uses the assessors to throw out sentences that don't contain answers. We excluded our own runs and all manual runs from this set of automatic QA runs.

Some of the runs performed worse than their corresponding baselines as measured by F ($\beta = 3$) despite being post-interaction runs, i.e. they are produced following the chance to interact with the assessors for 5 minutes. We chose to use the final runs because in a likewise fashion, for our experiments several of the assessors performed worse than our baseline. In fact, the average performance of the final automatic runs differed little from the average performance of the corresponding baselines. Using the assessors' judgments, the baselines had an average F ($\beta = 3$) of 0.354 and the final runs had an average score of 0.353. We excluded one baseline from this calculation because it was reported as a mistake [6].

	F ($\beta = 3$)		F ($\beta = 1$)	
	Human	Nugtr.	Human	Nugtr.
Avg. Auto-QA	0.353	0.368	0.167	0.175
Baseline	0.318	0.335	0.210	0.223
Assessors	0.347	0.360	0.333	0.343
User X	0.503	0.492	0.293	0.285
User A		0.337		0.264
User B		0.447		0.370
User C		0.288		0.320
Avg. Users A-C		0.357		0.318

Table 2: Overall results. Human scores are calculated using the NIST assessors' judgments, and Nugtr. scores are automatic scores calculated by Nuggeteer.

5. RESULTS AND DISCUSSION

In this section we report the results of our experiments and compare the results to the 8 automatic QA runs described in Section 4.

Table 2 shows the overall results for our experiments. While we report the F measure with β equal to both 3 and 1, submitted systems are likely to have been tuned for $\beta = 3$, which was the official metric of the track.

As discussed in Sections 2.1.3 and 2.2, we had some of our experiments judged by the NIST assessors and for some we used Nuggeteer to automatically judge results. Systems that were submitted to NIST, including the 8 automatic QA systems, receive the same or higher scores when judged by Nuggeteer. This is because Nuggeteer memorizes known answers but also may find new occurrences of nuggets in the submitted responses. As explained in Section 3.3, User X's human and Nuggeteer judged runs are slightly different causing the Nuggeteer score to be lower than the human score. Our experiments with Users A-C occurred post-ciQA and thus can only be judged automatically. To be able to compare across all experiments, we will discuss the Nuggeteer scores except where noted.

Our baseline had an F3 (F measure with $\beta = 3$) of 0.335 and an F1 of 0.223. We were somewhat surprised by how well our baseline scored considering its simplistic construction. The baseline's good performance likely shows that the assessors' questions were relatively easy from an IR perspective. In other words, the top ranked documents tended to be relevant.

For the 8 automatic QA systems, the average F3 was 0.368 with a minimum of 0.335 and a maximum of 0.402. For F1, the automatic QA systems scored an average of 0.175 with a minimum of 0.149 and a maximum of 0.214. Our baseline likely scored a higher F1 than the QA systems because we returned relatively shorter responses in comparison.

After using our IR system for 10 minutes, the assessors did quite well with an F3 of 0.360. This performance is only 2% less than the average automatic QA score of 0.368, but in comparison to our baseline, the assessors only achieved a 7% improvement. When we look at F1, we see significant differences in performance. The assessors scored an F1 of 0.343, which is a 54% increase over the baseline and a 96%

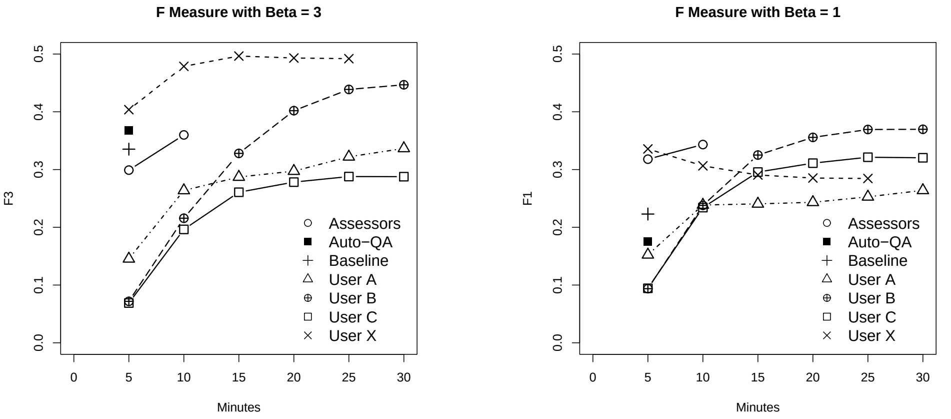


Figure 2: F ($\beta = 3$) and F ($\beta = 1$) performance over time as measured using Nuggeteer.

increase over the QA systems.

For complex information needs, human question answering performance using an interactive IR system appears to be competitive with automatic QA system approaches as measured by F3. Humans using an IR system perform considerably better than automatic QA systems when precision and recall are given equal weighting (F1).

We found similar levels of human performance for our other two experiments. Our expert user obtained the best performance of all submitted ciQA 2007 runs, but did not do as well as the assessors on F1. While the expert aimed for recall, it seems clear that the assessors know best what they want for answers. Users A, B, and C did well considering they are not expert searchers. On average they did effectively the same as the assessors for F3. For F1, these non-expert users did worse than the assessors but better than the automatic systems.

These results ignore the varying lengths of time the users had to find answers, which we examine in the next section.

5.1 Performance Over Time

Figure 2 shows the performance of the assessors, automatic QA systems, and users over time. We've placed the QA systems and our baseline at 5 minutes to reflect that the assessors required time to interact with (5 minutes) or at least read the output of these systems. Some QA systems may have taken much longer than 5 minutes per question to produce their answers, but here we give them the benefit of the doubt.

When the performance measure is F3, it takes the assessors the full 10 minutes to achieve a performance comparable to the average QA system. Our expert user, User X, does significantly better than the QA systems with only 5 minutes of interaction. User X's curve only goes out to 25 minutes because by this time he had stopped working. Users A-C all worked on at least one question past the 25 minute mark.

At 5 minutes, User X's F3 score is 0.404 and the best performing QA system's F3 is 0.402. Only two assessors did better than the best QA system at the 5 minute mark. Assessor 3 had an F3 of 0.174 while the best QA system obtained an F3 of 0.160 for Assessor 3's questions. Other QA systems did much better on Assessor 3's questions. Neither User X nor any of the QA systems could match Assessor 8's 5-minute F3 of 0.800. (Assessor 8's F3 performance declined to 0.751 at 10 minutes.) Assessor 8 performed 27% better than the best QA system's F3 of 0.630 and 40% better than the average QA system's F3 of 0.573. In the hands of an expert user, an IR system can be a very powerful tool for question answering.

We see that high levels of performance take time for non-expert users. The non-expert users on average took 388 seconds to enter their first answer compared to 84 seconds for User X and 135 seconds for the assessors. While the assessors only spent on average 79 seconds between entering answers, the non-expert users took 163 seconds. User X let only 49 seconds pass in between answers. Table 3 shows individual times for the 8 assessors.

A couple of reasons may account for the slower performance of the non-expert users. One reason may be the lack of a tight time constraint. Another is that these users may have required more time to become familiar with the subject matter of the question.

Of interest, User X's F1 performance decreases with time while all other users show an increase in performance over time. User X focused on recall and as a result decreased precision at a faster rate than he increased recall. The assessors perform well at both F3 and F1. As an assessor finds an answer, it is precise and it does increase recall.

Hidden in these averages is the variable performance of the assessors and the variation in questions, which we discuss in the next section.

5.2 Variation in Performance

In both Figure 3 and Table 3 we see that some assessors performed much better than others. Figure 3 shows the performance of the assessors and the automatic QA systems on a per question basis. Table 3 shows performance and other statistics for each assessor.

Figure 3 shows that for F3 the assessors did as well or better than the average automatic QA system on only half of the questions. Meanwhile, Table 3 shows that 5 of the 8 assessors on average obtained a higher F3 using the IR system than via the automatic QA systems. For F1, the assessors did as well or better on 23 questions and all but one assessor, Assessor 5, did better with the IR system than with the QA systems.

Assessor 5 scored zeros on 4 of 4 questions. Assessor 5 only entered one answer on one question and the assessor

	Assessors Ordered by F ($\beta = 3$)							
Mean Statistic	A5	A7	A1	A3	A2	A4	A6	A8
F ($\beta = 3$)	0.000	0.139	0.223	0.251	0.297	0.494	0.535	0.751
Avg. Auto-QA Sys. F ($\beta = 3$)	0.444	0.228	0.304	0.191	0.278	0.342	0.417	0.573
F ($\beta = 1$)	0.000	0.207	0.221	0.342	0.298	0.494	0.504	0.536
Avg. Auto-QA Sys. F ($\beta = 1$)	0.171	0.124	0.131	0.092	0.154	0.180	0.221	0.243
Answer Length (non-whitespace chars.)	47.0	51.2	238.8	59.8	202.6	115.8	237.7	162.7
Answers per Question	0.3	7.0	3.3	2.5	8.5	12.5	6.5	15.5
Nuggets per Answer	0.0	0.3	0.5	0.6	0.6	0.5	1.2	0.5
Precision	0.000	0.652	0.220	0.750	0.326	0.513	0.503	0.454
Recall	0.000	0.128	0.223	0.236	0.300	0.495	0.549	0.870
Time to First Answer (seconds)	251.9	84.7	149.8	155.9	121.0	100.7	123.8	90.3
Time between Answers (seconds)	n/a	67.1	84.1	201.8	55.4	39.2	79.9	27.3
Max. Document Rank Viewed	8.8	8.0	4.3	17.8	30.8	18.5	8.8	7.5

Table 3: Per assessor performance and statistics. Also shown is the average performance for the automatic QA systems on a per assessor basis. All scores calculated from the assessors' judgments.

judged that answer to contain no nuggets. Assessor 5's performance is partly explained by the assessor's comment on the exit-questionnaire regarding our IR system: "I guess I never really understood how this one was supposed to work." Examination of the log files also shows 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.

At the other end of assessor performance is Assessor 8 whose answers to three of the assessor's four questions achieved the highest F3 scores of all runs submitted to ciQA. Table 3 shows that Assessor 8 averaged an incredible recall of 0.870. Assessor 8 spent only 27.3 seconds between each entered answer. Examining the logs shows that Assessor 8 often extracted several answers from a single document.

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 3. Surprisingly, 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.

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."

5.3 Assessors' Precision

The precision of the assessors' answers seems low considering the assessors entered the answers themselves. A possible reason for the lower than expected precision is that the ciQA track's allowance for each returned nugget is 100 non-whitespace characters. As Table 3 shows, 5 of the 8 assessors all entered answers that on average were greater than 100 characters long. Three assessors had average answers 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 but a precision of only 0.582. Why? The assessor's total response was 1891 non-whitespace characters, but the assessor's allowance was only $11 \times 100 = 1100$ non-whitespace characters.

When we look at the average number of nuggets found per answer, the 100 character allowance also looks too small. The assessors entered 278 characters per nugget (the assessors' average of 139 characters per answer divided by the average 0.5 nuggets per answer).

On the other hand, if we expect QA systems to return carefully summarized answers, then Assessor 7 showed how to get perfect precision 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 giving an allowance of 300 characters and earning a precision of 1.

Finally, low precision may be an artifact of the complexity of judging answers. 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.

5.4 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."

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

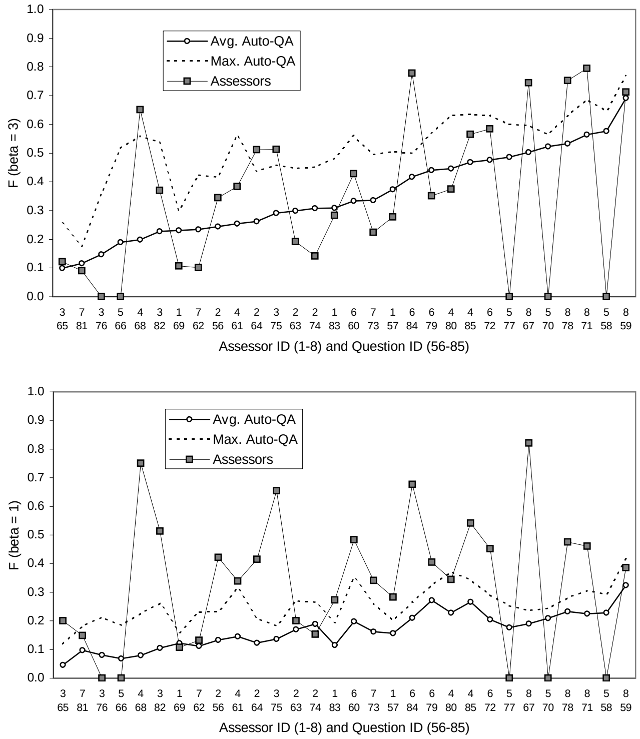


Figure 3: The per question performance of the assessors and the automatic QA systems. F3 is shown in the top chart and F1 in the bottom chart. These scores are calculated using the NIST assessors' judgments.

have already researched their questions as part of the question development process. While this familiarity may have boosted the assessors' performance with our system, it should have also boosted the QA systems. User X and the non-expert users did not research the questions in advance but showed good performance using the IR system to answer questions.

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.

6. 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 [12]. Measuring human performance using IR systems has a long history and was the focus of the TREC interactive track [4].

For TREC-9 (2000), the interactive track task used a fact-finding task that required users to view multiple documents to construct an answer [5]. 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 questing answering systems.

Recently, Lin [7] has shown with batch, non-interactive experiments that IR systems may be competitive with automatic QA systems when the users have complex information needs. Lin compared the performance of an IR system to the submitted runs for the TREC 2004 and 2005 question answering tracks. The IR system returned top ranked sentences and this static list was compared to a likewise constructed list for the QA systems. Lin's experiment is similar to our baseline submission where the results being measured are static ranked lists of sentences generated from a query. In contrast, our work looks at the performance of an IR system being used by humans as an interactive tool for question answering.

7. CONCLUSION

We measured the performance of humans using an IR system to answer complex questions. We found that human performance with an interactive document retrieval system is comparable to the average automatic QA system submitted to the TREC 2007 ciQA track. The results of some users suggest that interactive IR systems are inherently powerful for question answering, but we also found that human performance is quite variable. Some users would have more success with an automatic QA system. For question answering, automatic QA systems have ease-of-use and recall advantages over IR systems, which have flexibility and precision advantages. Future IR/QA systems should aim to integrate the best of both types of systems to meet the needs of both expert and novice searchers.

8. ACKNOWLEDGMENTS

We thank Hoa Trang Dang, Diane Kelly, Jimmy Lin, and the NIST assessors for making the TREC ciQA track possible. We also thank Gregory Marton very much for his great assistance with and support of Nuggeteer.

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

9. REFERENCES

[1] J. Allan. HARD track overview in TREC 2005: High accuracy retrieval from documents. In *TREC 2005*, 2006.

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

[3] H. T. Dang, D. Kelly, and J. Lin. Overview of the TREC 2007 question answering track. In *TREC 2007 Notebook*, 2007.

[4] 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.

[5] W. Hersh and P. Over. The TREC-9 interactive track report. In *TREC-9*, pages 41–50. NIST.

[6] D. Kelly, V. Murdock, and X. Fu. Using interactions to improve translation dictionaries: UNC, Yahoo! and ciQA. In *TREC 2007 Notebook*, 2007.

[7] 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.

[8] J. Lin and D. Demner-Fushman. Will pyramids built of nuggets topple over? In *HLT/NAACL 2006*.

[9] G. Marton and A. Radul. Nuggeteer: automatic nugget-based evaluation using descriptions and judgements. In *Proceedings of HLT/NAACL 2006*.

[10] L. Rainie and J. Shermak. Search engine use shoots up in the past year and edges towards email as the primary internet application. Pew Internet & American Life Project, Report 167, November 2005.

[11] 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.

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