User Simulations for Evaluating Answers to Question Series[*]

Jimmy Lin
College of Information Studies
Department of Computer Science
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742, USA
jimmylin@umd.edu

Abstract

Recently, question series have become one focus of research in question answering. These series are comprised of individual factoid, list, and "other" questions organized around a central topic, and represent abstractions of user–system dialogs. Existing evaluation methodologies have yet to catch up with this richer task model, as they fail to take into account contextual dependencies and different user behaviors. This paper presents a novel simulation-based methodology for evaluating answers to question series that addresses some of these shortcomings. Using this methodology, we examine two different behavior models: a "QA-styled" user and an "IR-styled" user. Results suggest that an off-the-shelf document retrieval system is competitive with state-of-the-art QA systems in this task. Advantages and limitations of evaluations based on user simulations are also discussed.

1 Introduction

Community-wide evaluations such as TREC, NTCIR, and CLEF have been a major driving force in the development of question answering technology over the past several years. While such events bring together the research community and provide a neutral forum where results can be meaningfully compared, formal evaluations have the downside of focusing attention on what can be easily measured, which may differ from what's actually important, realistic, or useful. In this paper, we describe shortcomings in the present TREC methodology for evaluating question series, propose an alternative framework based on user simulations, and use this new tool to examine interesting information-seeking behaviors brought to light by a previous user study.

Recent implementations of the question answering task have focused on contextualized information needs, which stands in contrast to earlier work on isolated "factoid" questions such as "What membrane controls the amount of light entering the eye?" Since 2004, the main task at NIST-sponsored TREC QA tracks have consisted of question series organized around topics (called "targets")—which can be people, organizations, events, or entities (Voorhees, 2004; Voorhees, 2005); cf. (Kato et al., 2004). Questions in a series inquire about different facets of a target, but are themselves either factoid or list questions. In addition, each series contains an explicit "other" question (always the last one), which can be paraphrased as "Tell me other interesting things about this target that I don't know enough to ask directly." Table 1 shows a few sample question series.

[*] Please cite as: Jimmy Lin. User Simulations for Evaluating Answers to Question Series. *Information Processing and Management*, 43(3):717-729, 2007. This is the pre-print version of a published article. Citations to and quotations from this work should reference that publication. If you cite this work, please check that the published form contains precisely the material to which you intend to refer. (submitted: April 20, 2006. accepted: June 28, 2006)

11. the band Nirvana		
1	factoid	Who is the lead singer/musician in Nirvana?
2	list	Who are the band members?
3	factoid	When was the band formed?
4	factoid	What is their biggest hit?
5	list	What are their albums?
6	factoid	What style of music do they play?
7	other	

38. quarks		
1	factoid	What kind of a particle is a quark?
2	factoid	Who discovered quarks?
3	factoid	When were they discovered?
4	list	What are the different types of quarks?
5	other	

69. France wins World Cup in soccer		
1	factoid	When did France win the World Cup?
2	factoid	Who did France beat for the World Cup?
3	factoid	What was the final score?
4	factoid	What was the nickname for the French team?
5	factoid	At what stadium was the game played?
6	factoid	Who was the coach of the French team?
7	list	Name players on the French team.
8	other	

Table 1: Sample question series.

Question series represent an attempt to incorporate context-processing as a component of the evaluation: anaphors are liberally used to generate natural-sounding questions, and earlier questions may provide context for later ones. Systems are required to process questions within a series sequentially (with no look-ahead) and are allowed to preserve state (not the case with previous TREC QA evaluations). For "other" questions, credit is not awarded for returning information already explicitly asked for; this setup forces systems to keep track of the current state of knowledge.

Question series can be viewed as abstractions of information-seeking dialogs, where a user interacts with a system to accumulate a body of knowledge (i.e., facts) about a topic. This development occurred in response to the realization that factoid questions, the focus of much previous research, do not usually occur in isolation, but are typically components of broader information needs that can only be fulfilled through multiple user–system iterations. Since fully-interactive user studies are difficult to organize within the TREC setting, question series were seen as an acceptable compromise.

Despite a focus on question series, the evaluation methodology at TREC has yet to incorporate notions of context or models of the user. Individual questions in a series are still evaluated as if they occurred in isolation, and then aggregated by a weighted sum (Voorhees, 2004; Voorhees, 2005). This implicitly assumes a hypothetical user who methodically types in each natural language question and assesses the response. Needless to say, such a model vastly oversimplifies the information seeking process, which is a complex dance of broad and directed searching, browsing, serendipitous knowledge discovery, etc. Evaluations aimed at studying information-seeking dialogs should attempt to model some of these interactions.

In many cases, user studies represent the most "natural" method for studying the information-seeking behavior of humans. However, the high-cost and time-consuming nature of such experiments limit the range of hypotheses that can be considered, the speed at which variables can be explored, and the statistical significance of results. The dominant paradigm of TREC-style batch evaluations (i.e., the Cranfield methodology) is plagued with the opposite sets of problems—while reusable test collections allow for rapid experimentation, removing the user from the loop eliminates arguably the single most important variable in the information-seeking process, thus affecting the conclusions that can be drawn. Different (and sometimes conflicting) results from system- and user-oriented evaluations have been noted by a variety of researchers, e.g., (Hersh et al., 2000; Allan et al., 2005). As an attempt to retain the best of both worlds, i.e., conduct interactive system evaluations in a rapid, affordable, and repeatable manner, user simulations have gained traction as an alternative experimental methodology (Harman, 1988; Magennis and van Rijsbergen, 1998; Chi et al., 2001; Mostafa et al., 2003; White et al., 2005).

Taking inspiration from previous work in IR research, this paper develops a framework for evaluating question answering systems with user simulations. In our setup, systems are plugged into user models that simulate the actions that a real user would take given certain observations (system output). To assess the effectiveness of different user–system combinations, we introduce a novel measure that quantifies the number of facts acquired as a function of time. Using this simulation framework, we explore two different types of user behaviors: one focused on question asking, and one focused on reading retrieved results. These two behaviors essentially boil down to using a QA system versus using an IR system to answer question series. Surprisingly, our user simulations show that a baseline document retrieval system beats all but the top-ranking question answering systems.

This paper is organized as follows: Section 2 discusses the motivation for this work. Section 3 outlines the simulation-based evaluation methodology adopted in our study. Detailed evaluations of factoid, list, and "other" questions are discussed in Sections 4, 5, and 6, respectively. An attempt at aggregating performance across these three disparate question types is described in Section 7. The limitations of this evaluation framework and related issues are discussed in Section 8, and the paper concludes in Section 9.

2 Motivation

The user study on context in question answering conducted by Lin et al. (2003) serves as a starting point for this work. Using a QA system, subjects in their study were asked to find answers to groups of related factoid questions centered around a topic, much like question series. The independent variable in the study was the amount of answer context presented to the user. In one condition, only the exact noun phrase answer was given; in three other conditions, the system presented the entire sentence, paragraph, and document in which the answer was found, respectively (with the answer highlighted). When given only the exact answers (and to a large extent, sentence-based answers), users were forced to ask each question in sequence (as there was nothing else the user could possibly do). This type of user behavior implicitly underlies the current setup of the TREC QA task: the user types in a question, reads the response, and proceeds to the next question, repeating until all questions have been answered.

Richer behaviors emerged when users were given the answer to a question embedded in a paragraph or the entire source document. Instead of proceeding to ask the next question in the series, users would read the context surrounding the given answer, which often incidentally contained answers to other questions in the series. Hence, users were able to satisfy their information needs "serendipitously". If reading system output did not yield any interesting information (e.g., answers to more questions), users would continue with the next question in the series (by typing in the query box).

In the document-length response condition, a few users appeared to not be using the test system's question answering capabilities at all. These subjects would simply type in a few general keywords and read the resulting document end-to-end. Since the experiment used high-quality encyclopedia articles and the questions were focused on coherent, well-defined topics, users were able to answer all the questions by simply reading the entire article (which varied in length, however).

Interestingly, Lin et al. found no statistically significant differences in the amount of time required to answer each series or the answer accuracy between each of the four conditions. They, however, did note a very significant decrease in the number of questions posed to the system with increasing amounts of context, suggesting a tradeoff between reading and typing time. Overall, users preferred the paragraph condition because it represented a good balance between establishing context and maintaining brevity.

This previous work brought to light several interesting user behaviors. On one extreme is a methodical question-asking behavior in which the user simply poses each question in the series sequentially. The other extreme is a behavior that involves minimal querying, but a significant amount of reading. These two approaches roughly translate into using a QA system and using an IR system to gather information about a target, respectively. In this paper, we explore both behaviors with the aid of user simulations.

The use of user simulations to evaluate information retrieval systems is relatively well-established; see (Harman, 1988; Magennis and van Rijsbergen, 1998; Chi et al., 2001; Mostafa et al., 2003; White et al., 2005) for a few examples. Such studies can be used as formative tools to rapidly assess the interactive capabilities of different information systems. As far as we are aware, this work represents the first attempt at applying such a methodology to question answering evaluation.

3 Evaluation Methodology

The current TREC evaluation methodology for question series is based on an aggregation of individual scores (first within a series, then across series) that does not take into account context, the relationship between each question, or a model of the user. Although questions in a series are related, they are evaluated in isolation. Furthermore, metrics focus on either precision (in the case of factoids) or a combination of precision and recall (list and "other" questions); these are not the best measures for quantifying performance if the overall goal is to gather a number of facts about a target.

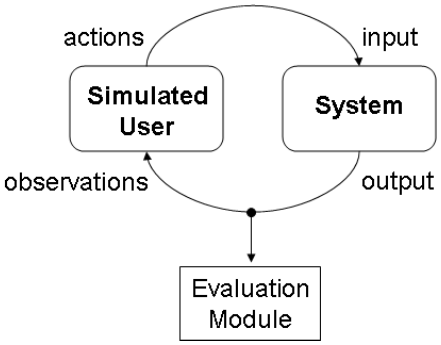


Figure 1: Setup of the simulation-based evaluation framework.

User model	**System**
Model A	top-ranking QA system from TREC 2004
Model A	2nd-ranking QA system from TREC 2004
Model A	3rd-ranking QA system from TREC 2004
Model A	median QA system from TREC 2004
Model B	Lucene
Model B	Indri

Table 2: Simulations explored in this paper.

We propose user simulations as an alternative evaluation methodology that addresses some of the shortcomings associated with the existing TREC paradigm. Our evaluation framework, which consists of three components, is shown in Figure 1. The user model, which simulates users' information-seeking behavior, is connected to the system under study in such a way that user actions become system inputs and system outputs become user observations. System output also serves as the input to the evaluation module, which measures the effectiveness of the interaction (i.e., user–system combination).

Using this evaluation methodology, we explore the two types of user behaviors described by Lin et al. (2003), in an effort to better understand QA and IR systems. The first model (model A) simulates a user who types in a question, reads the system's response, and repeats. Such a user is looking only for the answer to the current question, and sequentially goes through all questions in this narrowly focused manner—essentially, all questions are treated as if they occurred in isolation. This is, in fact, the implicit user model that underlies the current TREC QA setup. To complete the simulation, this user model is paired with systems that were evaluated in the TREC 2004 QA track: the top-ranking, second-ranking, and third-ranking systems (by different organizations), and the median system (across all submitted runs). The second user model (model B) simulates someone who prefers reading over typing: a user who issues a simple query, obtains a ranked list, and starts reading the documents in search of answers—i.e., a user employing a document retrieval system for a question answering task. This model is paired with two different document retrieval systems: Lucene, a popular open-source retrieval engine, and Indri (Metzler and Croft, 2004), a state-of-the-art language modeling toolkit for information retrieval. For both systems, the target itself was used as the query. To give a specific example, consider again the sample question series in Figure 1: the queries issued to the IR systems would be "the band Nirvana", "quarks", and"France wins World Cup in soccer", respectively. Since model B attempts to simulate a user who is interested in documents about the general topic (and is willing to examine the documents manually), additional query terms from individual factoid or list questions are not used. The simulations explored in this work are summarized in Table 2, and the realism of these runs is discussed in Section 8.

Instead of traditional TREC evaluation measures based on precision and recall, we propose a novel evaluation metric[1] based on the number of facts gathered (i.e., recall) as a function of time. Since time is not a directly measurable quantity in our simulations, response length is used as a surrogate, under an assumption of constant reading time. This evaluation model is attractive for several reasons. For one, it is easy to interpret and compare: naturally, a system that allows the user to gather more facts in a shorter amount of time is preferred. Single-point measures can still be computed at a given time cutoff, reflecting specific situations (e.g., writing a complete report in an hour, quick fact-checking in 5 minutes, etc.).

By measuring recall as a function of response length, we can create plots analogous to precision–recall curves in *ad hoc* retrieval that explicitly show the tradeoffs between completeness and brevity. Note that this tradeoff is especially important for list and "other" questions, whose responses potentially vary greatly in length; the current metric (F-measure) hides this important issue with arbitrary settings of the β parameter. In the following sections, we describe the results of each simulation in Table 2. Runs for factoid, list, and "other" questions are discussed separately.

4 Factoid Questions

Before the general evaluation methodology described in the previous section can be implemented, there are a number of details that must be worked out. The first involves preparing the dataset, and the second involves more concretely fleshing out the scoring methodology.

As discussed above, user model A is paired with the top-three scoring runs (by unique organizations) and median run (across all submitted runs) from the TREC 2004 QA track. However, it does not make sense to use those system outputs verbatim. Submitted factoid answers to TREC are by requirement "exact" (but they are paired with a supporting document). Such short answers (usually noun phrases) provide no context for the user to ascertain the correctness of the response. More realistically, users would require some fragment of a source document surrounding the answer to make sense of the response—this context is reconstructed by finding the first sentence in the supporting document that contains the answer string; if no exact match could be found, the sentence with the most terms in common with the exact answer is chosen.

The next issue with model A simulations concerns user backoff strategies when the QA system fails to return correct answers. Although the current TREC evaluation model does not specify a "repair strategy", it is reasonable to assume that users would revert back to keyword-based querying to find the remaining answers. This is simulated by appending Lucene results (described below) as the final system response with model A.

The results of Lucene and Indri, which are paired with model B, are also processed. Within the documents retrieved, it is likely that matched terms will be highlighted, drawing user attention to areas in the document that contain query terms. We assume that users will read those regions more carefully, and quickly skim (or altogether skip) areas of the document that do not contain query terms. To approximate this effect, we discard all sentences that do not contain at least one term from the target. Documents from the hit list were processed in this manner and concatenated together until a quota of 15000 non-whitespace characters was filled (the length at which we stopped plotting the performance graphs). We assume that the model B user will methodically read documents in the hit list, one after another, until either all questions have been answered or the user gives up. The length of 15000 non-whitespace characters represents an upper bound, but we expect that few real users would actually read that much text.

How do we actually evaluate answer recall as a function of response length? While intuitively simple, there are many details that must be considered. First, there must exist an automatic method

[1] Novel for question answering, that is.

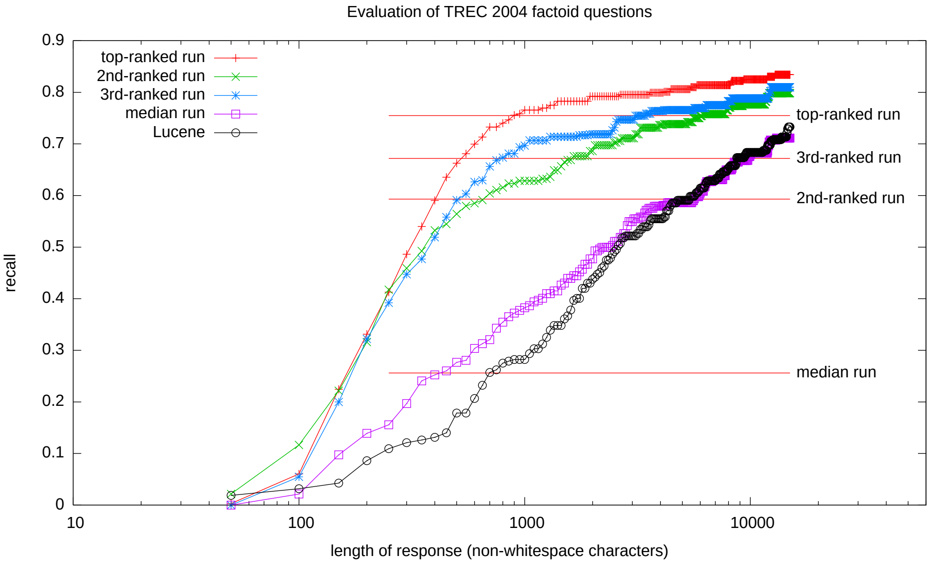


Figure 2: Evaluation of TREC 2004 factoid questions: answer recall vs. response length for top three runs, the median run, and Lucene.

for determining if an answer is contained within a span of text. For this, we employ regular expression answer patterns distributed by NIST, which have become a widely-accepted method for evaluating answers to factoid questions. Although there are a number of known issues with these patterns (Lin, 2005), they equally affect both model A and model B simulations—although absolute numbers may not be accurate, relative rankings will be stable. As a note, although it is possible to compute official answer accuracy scores as a function of response length given NIST judgments, doing so would prevent us from comparing model A simulations to model B simulations (which were not submitted to NIST for evaluation). In summary, what we measure is the fraction of factoid questions in a series that has been answered after going through a certain amount of system output, under the assumption that sentences are read one at a time and answers aren't acquired until the complete sentence has been read. No ordering relationship is enforced among the questions, i.e., a particular sentence might answer the first question in the series, the last, or even multiple questions simultaneously. Naturally, however, answers retrieved multiple times are given credit only for their first occurrence. The behavior of model A, however, ensures that questions are answered sequentially (since each is explicitly asked); the order of answers with model B, on the other hand, exhibit much greater freedom.

The next issue that needs to be addressed concerns the length unit of evaluation. Possibilities include (at least) characters, terms, and sentences. Since we have projected exact answers onto complete sentences, it makes sense to evaluate answer recall at the sentence level. However, since sentences vary greatly in length, characters seem like a more comparable unit of measurement (following TREC, we only count non-whitespace characters).

We adopted a method for interpolating between lengths and aggregating results that is very similar to the computation of precision–recall curves in *ad hoc* retrieval tasks. For each question series, recall values were interpolated to the nearest 50 character increment higher than the current length (i.e., the length count after the current sentence has been "read"). Furthermore, recall was interpolated to be monotonically increasing so that score variations at different lengths were smoothed out. In this manner, the graph for all factoid questions within a given series can be plotted. Results across different series were aggregated by averaging recall values at the fixed length increments (50, 100, 150, etc. characters).

Results of this evaluation are shown in Figure 2, where answer recall is plotted against response length (on a log scale). The user model is omitted in the key because it is unambiguously identified by the system. Furthermore, Lucene was found to outperform Indri, and hence Indri results were not plotted to reduce clutter. This graph has a very intuitive interpretation: after reading so much response from the system, one can expect to have obtained answers to a certain fraction of all factoid question in the series. Naturally, higher and faster-rising curves indicate better performance. The horizontal lines in Figure 2 indicate the performance of each QA system alone (i.e, if the IR results had not been appended to the end of the QA run). Note that for the appended IR runs, answers that have already been returned were not removed.

For the top three systems, recall rises very quickly at first due to their high accuracy in retrieving factoid answers, then tapers off because users must resort to IR results to find the remaining answers. As a reference, the top three runs in TREC 2004 obtained an official factoid accuracy of 0.770, 0.643, and 0.625, respectively (Voorhees, 2004). Note, however, that these official scores differ from the scores automatically computed using answer patterns. Interestingly, the median QA system does not appear to outperform Lucene at higher levels of recall. Because the median system is only able to answer 17% of all factoid questions (official NIST score), the user must resort to reading IR results to find most of the answers.

5 List Questions

Answers to list questions in TREC consist of an unordered set of strings; as with the factoid questions, the exactness criterion also applies. In the NIST evaluation, responses from all systems are pooled to create the known set of correct answers, which is then applied to evaluate each individual run. The official score is an F-measure with equal balance on precision and recall. As previously discussed, such a single-point measure hides important tradeoffs that are relevant for different situations.

We evaluated answers to list questions in the same way we evaluated answers to factoid questions: recall as a function of response length (non-whitespace characters). The systems under comparison were prepared in a similar manner. Answers from the top three runs from unique teams, along with the median across all runs (some overlap, but not the same as the factoid runs), were projected onto the sentence containing the exact answer (or closest match). The baseline Lucene and Indri runs were exactly the same as those in the factoid evaluation. The IR results were also appended to the end of the TREC runs. Answers were automatically evaluated with patterns distributed by NIST, and results were aggregated in the same manner as factoids: first on a per-series basis to the nearest higher 50 non-whitespace character increment (interpolated so that recall values were monotonically increasing), and then averaged at each length quanta. As with factoid questions, no ordering relationship was enforced among answer instances—even if a question series had multiple list questions, our evaluation method accepted answers in arbitrary order, including an interleaving of responses.

Results of this experiment are shown in Figure 3. As before, since we discovered that Lucene outperformed Indri, the latter results are not shown to reduce graph clutter. Horizontal lines mark the performance of each QA system alone, without contributions from the appended IR results. For reference, official NIST scores are shown in Table 3. Once again, these figures differ from the automatically-generated scores. As can be seen from the graph, the performance of the median QA run does not appear to be better than Lucene's; both were approximately equal to the third-highest scoring run at high recall levels. Based on this evaluation methodology, only the top-ranked and second-ranked run conclusively "beats" Lucene on list questions.

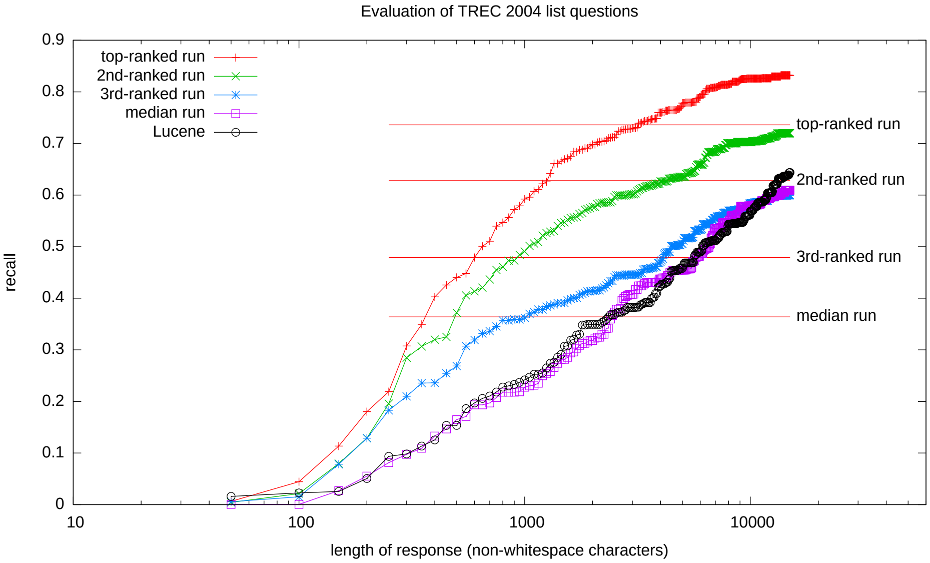


Figure 3: Evaluation of TREC 2004 list questions: answer recall vs. response length for top three runs, the median run, and Lucene.

run	**P**	**R**	**F$_1$**
top-ranked run	0.627	0.665	0.622
2nd-ranked run	0.488	0.551	0.485
3rd-ranked run	0.214	0.444	0.258
median run	0.108	0.107	0.094

Table 3: TREC 2004 list questions: official score of selected runs.

6 Other Questions

"Other" questions can be paraphrased as "Tell me interesting stuff about the target that I didn't explicitly ask about". System responses consist of a set of answer strings, making the task quite similar to passage retrieval. The goal is to retrieve as many "nuggets" (essentially, facts) about the target as possible (beyond information already asked about by the other questions in the series). NIST assessors create an answer key of "reference nuggets" by examining the pooled response of all systems (plus their own research), and each nugget is labeled as either "vital" or "okay" to qualitatively denote their importance; cf. (Lin and Demner-Fushman, 2006). There are two components to the evaluation metric: recall is computed on vital nuggets only, while precision is approximated by a length allowance per vital or okay nugget—more details are provided in (Voorhees, 2004; Voorhees, 2005). An F_3 score combines the precision and recall components, and the setting of $\beta = 3$ (weighting recall over precision by a factor of three) was arbitrarily determined.

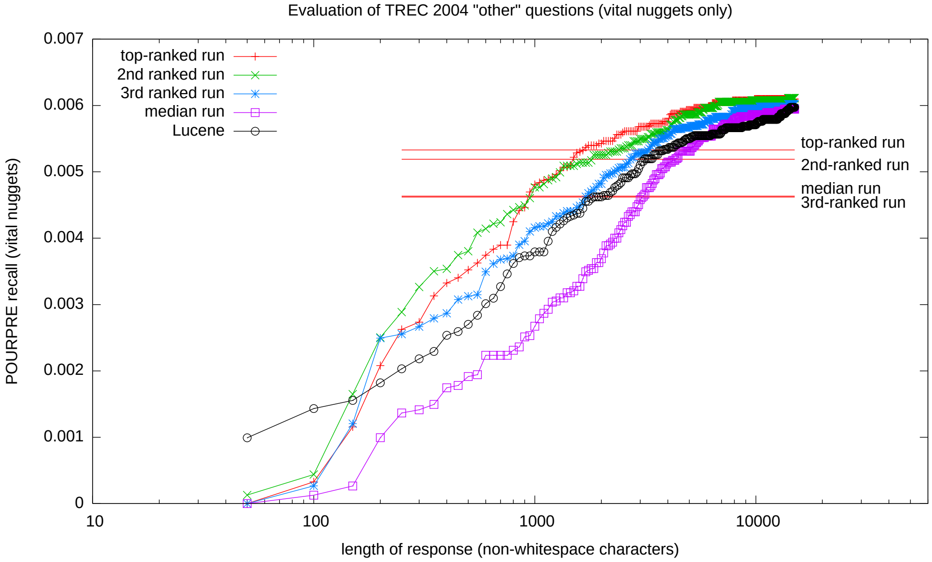
In this experiment, we examined the same user simulations outlined in Table 2: model A paired with the top three and median runs from TREC 2004, and model B paired with Lucene and Indri (same exact runs as before). We employed a similar evaluation methodology that characterizes recall as a function of response length. With factoid and list questions, regular expression patterns provided by NIST were used to automatically assess answer accuracy—a corresponding method to evaluate answers to "other" questions was needed for our experiments. For this purpose, we employed Pourpre, a recently-developed metric for automatically evaluating answers to complex questions (Lin and Demner-Fushman, 2005). The metric is based on unigram overlap between answer strings and the assessors' reference nuggets. We used the (term count, macroaveraging) variant, which was found to produce the highest correlation with official results. Since we were only interested in recall, we discarded the precision component generated by Pourpre. Note that although it would be possible to reconstruct official nugget recall scores as a function of response length given NIST judgments, doing so would prevent us from evaluating runs that did not participate in the original TREC evaluation. Thus, the use of Pourpre is critical to these experiments. As with the factoid and list experiments, Pourpre accepts relevant nuggets in any order.

The results of our experiments on TREC 2004 "other" questions is shown in Figure 4, under two different conditions: one in which only vital nuggets are considered, and one in which all nuggets are considered. In the same manner as before, the Lucene results were appended to the end of the QA system runs. As with the other graphs, Indri results are not plotted due to its lower performance. Similarly, horizontal bars denote the performance of each QA run without contributions from Lucene. For reference, the average response lengths and official F-scores are shown in Table 4. It does not appear that the median QA run (with model A) outperforms Lucene (with model B), and it is unclear if the top three QA runs are actually better, especially at high recall levels. Furthermore, despite relatively large differences in official scores, the top three QA runs don't seem to behave all that differently under this scoring model. The official metric represents a single tradeoff point between brevity and completeness—although these measures may appear different, the performance curves they lie on are actually quite similar. Our findings are consistent with results from the TREC 2003 QA track, where it was discovered that a baseline IR run beat all but the best submission at answering definition questions (Voorhees, 2003).

7 Aggregate Performance

In the previous sections, we described separate user simulations for factoid, list, and "other" questions. Given that a series is comprised of all three types, it would be desirable to compute aggregate performance on a per-series basis.

Unfortunately, there are many difficulties in developing an aggregation method that makes sense.



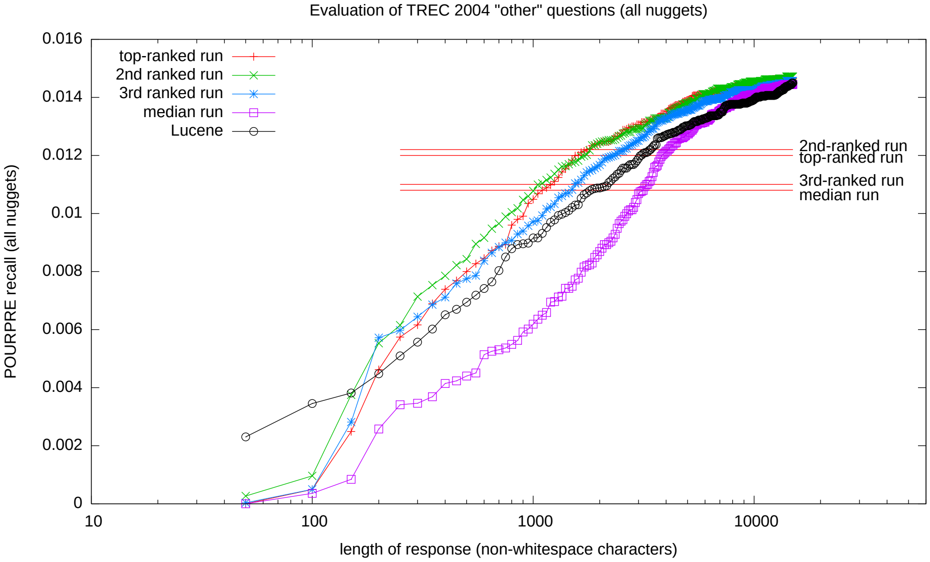


Figure 4: Evaluation of TREC 2004 other questions: recall vs. response length for top three runs, the median run, and Lucene; vital nuggets only (top) and all nuggets (bottom).

run	**avg. length**	$\mathbf{F}_3$
top-ranked run	1964	0.460
2nd-ranked run	1980	0.404
3rd-ranked run	2599	0.376
median run	3733	0.184

Table 4: TREC 2004 other questions: official score of selected runs.

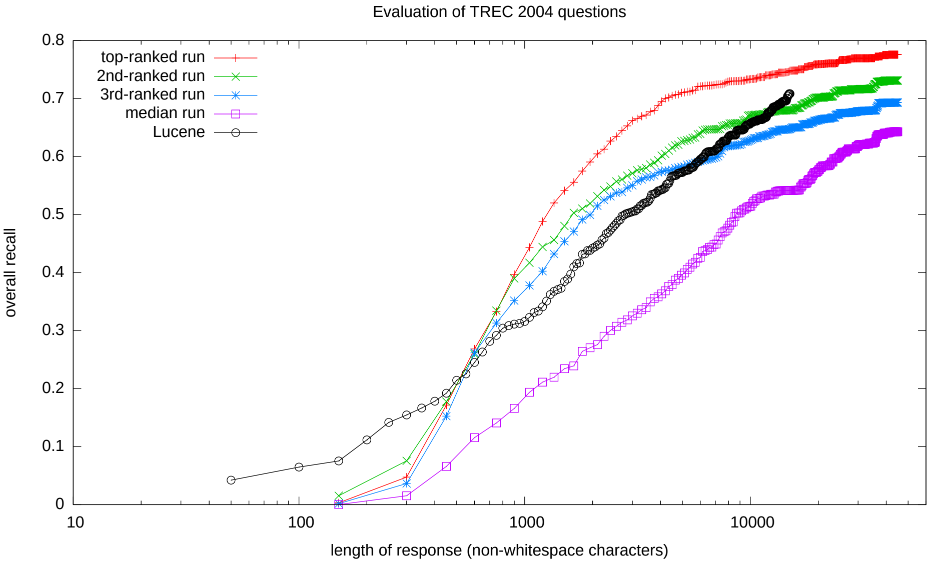


Figure 5: Overall evaluation of TREC 2004 questions: aggregated recall of all three question types vs. response length for top three runs, the median run, and Lucene.

The biggest barrier is the incomparability of basic answer units for each of the three question types. What is the relative importance of answering a factoid question, compared to retrieving a correct answer instance for a list question, compared to extracting a relevant nugget for an "other" question? In absence of a more well-defined task model to ground these individual facts, e.g., a final report to be prepared about the target, it is difficult to develop an aggregation method that makes sense. For TREC, NIST settled on an arbitrary weighting scheme: a weight of 0.50 to factoid questions (answer accuracy), 0.25 to list questions (F_1 score), and 0.25 to "other" questions (F_3 score). Each component score is aggregated on a per-series basis, and then averaged across all series.

We can straightforwardly apply this weighting scheme to produce an overall recall–length plot for the simulation with model B and Lucene. Recall that the input to each of the three individual evaluations is exactly the same; hence, we can simply compute the weighted average of all three recall scores at each length increment. We aggregated the scoring condition of the "other" questions that took into account all nuggets, both vital and okay.[2] For the model A simulations, the computation was a lot more complex because the output of the three different question types was different. In absence of a reasonable user model for reading answers to factoid, list, and "other" answers simultaneously, we settled on a "round-robin" baseline. We computed weighted recall averages in chunks of 150 non-whitespace characters, taking fifty from factoid, list, and "other" questions, respectively. Note that for both cases, we were aggregating results that already captured the performance of each question type across all question series. An alternative would be to first aggregate each question type within a series (i.e., take factoid, list, and "other" responses in a round-robin fashion), and then aggregate across all question series.

The results of this experiment are shown in Figure 5. A caveat: the graph shows the aggregated performance of the top factoid run with the top list run and the top "other" run (and similarly for the rest of the runs)—these were in actuality taken from different TREC submissions since no single run placed highest in all three categories. Although we readily concede that this aggregation method does not actually reflect any real user behavior, it nevertheless gives us a rough idea of how state-of-the art

[2] Since POURPRE recall scores were much lower, we arbitrarily decided to multiply them by fifty so that all recall ranges were roughly comparable.

QA systems compare to IR system.

The shape of these curves do make intuitive sense. If the goal is to accumulate a body of knowledge about a topic, then retrieving documents about the topic in general (using IR) is a good strategy because results can simultaneously provide answers to different questions and question types. Current question answering systems can only focus on one type of question at a time, to the exclusion of other potentially-relevant information. However, the rate of fact accumulation slows down for IR systems due to redundancy in the retrieved results (since documents lower in the ranked list might contain duplicate information), and performance falls below the top-ranking QA systems above a certain length threshold. Since all the QA runs were padded with IR results, Lucene again becomes competitive at longer answer lengths. Nevertheless, the Lucene performance curve lies above the median QA system performance curve, suggesting that unless an information seeker was using one of the top-ranking QA systems at TREC, she would be better off using an off-the-shelf IR engine.

8 Discussion

Question answering is an area of research that lies at the intersection of information retrieval and natural language processing. Since the development of large-scale open-domain factoid systems in the late nineties, the relative contribution between the two technologies has been a matter of debate. Some researchers have questioned the importance of linguistic analysis, compared to, say focused passage retrieval, while others have questioned the entire QA paradigm (Spärck Jones, 2003). This paper explores these important issues using a novel simulation-based framework. Surprisingly, our experiments reveal that an off-the-shelf retrieval engine (Lucene) is quite competitive, and actually outperforms the median TREC QA system in many cases. Under a few circumstances, the effectiveness of Lucene rivals that of the best question answering systems.

With the growing focus on complex questions and a better understanding user preferences, the “gap” between IR and QA has narrowed over the past few years. Researchers had believed that since answers to factoid questions consist of short noun phrases, a QA system with answer pinpointing capabilities would be superior to information retrieval systems, which operated at the level of documents and other more coarse-grained segments. However, two relatively recent developments have altered this landscape: the shift towards complex questions meant that system responses were now much longer, and it appears that humans prefer contextualized answers anyway, even for factoid questions. The results of our simulation experiments are consistent with these trends, and the growing emphasis on richer answers will reward systems with good passage retrieval (Monz, 2003; Tellex et al., 2003). Nevertheless, we believe that many of the linguistic analysis techniques from existing QA systems will remain relevant and applicable in the future.

This work demonstrates that user simulations provide a useful framework for exploring research issues in question answering. Nevertheless, there are a number of limitations that should be discussed. One limitation concerns the nature of the experimental setup: the placement of the evaluation module represents an oversimplification because it derives quantitative measures solely from system output, as opposed to the (simulated) user’s internal state. This choice means that our framework cannot, for example, actually measure how simulated subjects “learn”, only what they “see”—since it is entirely possible to read a piece of text without understanding its contents. Overcoming this limitation requires the development of more sophisticated user models that better capture cognitive states. The evaluation module can then “probe” different components of the user model to quantify appropriate aspects of the simulation.

Another potential limitation of the evaluation methodology concerns its reliance on automatic means for assessing the score of runs (answer patterns for factoid and list questions, and Pourpre for “other” questions). There is a certain amount of error associated with these scoring devices, which may potentially affect the results. However, this is tempered by the fact that both the TREC system runs and the IR runs were evaluated in the same manner, and hence whatever issues plague the use of these automatic evaluation devices affected all simulations equally.

Other limitations of this work concern the realism of the user models. Obviously, neither model A nor model B represents real users. When using a QA system, humans are likely to drill down into documents containing answers and examine surrounding contexts (if given the choice in the interface). On the other hand, a human using an IR system would not simply issue one query and read all resulting documents—more realistically, we would expect multiple iterations of broad and targeted searching combined with examination of the results. Although real-world user behaviors are vastly more complex, the two user models examined in this work are instructive because they bracket the space of user–system interactions. Furthermore, evaluations based on user-simulations provide a key advantage over TREC-style batch experiments in that they incorporate limited elements of interaction.

In designing model A and model B, we have made a number of simplifying assumptions that are unrealistic, beyond the points already mentioned above, in an attempt to balance evaluation complexity and insightfulness of evaluation results—it is worthwhile to point them out here. In projecting exact answers onto sentences for the TREC runs, we assume that sentences provide adequate answer justification. As Lin et al. (2003) discovered, this may not necessarily be the case. Many short sentences are burdened with linguistic phenomena such as dangling anaphoric references that make them difficult to comprehend in isolation. Although expanding the contextual window increases the response length, it may be balanced by other facts serendipitously appearing in the surrounding text. Another oversimplification is represented by appending IR results to the end of the TREC runs, simulating a user's backoff attempt at answering questions on which the QA systems failed. At that point, a user is much more likely to pose focused queries, which means that the current tail of the performance curves are conservative underestimates.

Similarly, a number of simplifying assumptions were made in model B. Sentences without query terms were thrown away, even though they may contain useful information. Recall that this represented a simplified model of scanning results—under the assumption that with proper keyword highlighting, users would naturally focus on regions in the documents with query terms. However, humans are very adept at skimming, and may potentially pick up other relevant information in the document. A more refined model of how users interact with information objects could be built from studies that employ eye-tracking, for example, (Granka et al., 2004).

It is important to note that most of the issues discussed above are criticisms about the realism of the specific implementations examined in our experiments, and not about the general approach. We believe that the evaluation methodology based on user simulations yields more insight and better models real world usage of retrieval systems, compared to traditional TREC-style batch evaluations. Most importantly, simulation-based evaluations are not limited to measuring one-shot retrieval effectiveness, but provide a method for assessing system performance in interactive settings.

9 Conclusion

The two primary contributions of this work are a general framework for simulation-based evaluations of question answering and a concrete instantiation of this general approach to compare QA systems from TREC 2004 to existing off-the-shelf IR engines. This work represents the first such application of user simulations that we are aware of. The methodology allows one to introduce the important roles of interaction and context in large-scale automatic evaluations, thereby simultaneously capturing many benefits of TREC-style batch experiments and user studies. Although the types of user models explored in this work are relatively primitive, our experiments do reveal interesting relationships between IR and QA technologies. Ultimately, we hope that the simulation-based evaluation methodology will drive the development more effective interactive retrieval systems.

10 Acknowledgments

This work has been supported in part by DARPA contract HR0011-06-2-0001 (GALE). Some of the ideas behind the user simulation approach to IR evaluation originated at the 2004 SWIRL workshop, organized by Alistair Moffatt and Justin Zobel. I would like to thank Doug Oard and Ryen White for many discussions that have contributed substantially to the shaping of the framework described in this paper. In addition, I would also like to thank Esther and Kiri for their loving support.

References

James Allan, Ben Carterette, and Joshua Lewis. 2005. When will information retrieval be "good enough"? User effectiveness as a function of retrieval accuracy. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*.

Ed H. Chi, Peter Pirolli, Kim Chen, and James Pitkow. 2001. Using information scent to model user information needs and actions and the Web. In *Proceedings of 2001 SIGCHI Conference on Human Factors in Computing Systems (CHI 2001)*.

Laura A. Granka, Thorsten Joachims, and Geri Gay. 2004. Eye-tracking analysis of user behavior in WWW-search. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*.

Donna Harman. 1988. Towards interactive query expansion. In *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1988)*.

William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kramer, Lynetta Sacherek, and Daniel Olson. 2000. Do batch and user evaluations give the same results? In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*.

Tsuneaki Kato, Jun'ichi Fukumoto, Fumito Masui, and Noriko Kando. 2004. Handling information access dialogue through QA technologies—a novel challenge for open-domain question answering. In *Proceedings of the HLT-NAACL 2004 Workshop on Pragmatics of Question Answering*.

Jimmy Lin and Dina Demner-Fushman. 2005. Automatically evaluating answers to definition questions. In *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*.

Jimmy Lin and Dina Demner-Fushman. 2006. Will pyramids built of nuggets topple over? In *Proceedings of the 2006 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting (HLT/NAACL 2006)*.

Jimmy Lin, Dennis Quan, Vineet Sinha, Karun Bakshi, David Huynh, Boris Katz, and David R. Karger. 2003. What makes a good answer? The role of context in question answering. In *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (INTERACT 2003)*.

Jimmy Lin. 2005. Evaluation of resources for question answering evaluation. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*.

Mark Magennis and C. J. van Rijsbergen. 1998. The potential and actual effectiveness of interactive query expansion. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998).*

Donald Metzler and W. Bruce Croft. 2004. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735–750.

Christof Monz. 2003. *From Document Retrieval to Question Answering.* Ph.D. thesis, Institute for Logic, Language, and Computation, University of Amsterdam.

Javed Mostafa, Snehasis Mukhopadhyay, and Mathew Palakal. 2003. Simulation studies of different dimensions of users' interests and their impact on user modeling and information filtering. *Information Retrieval*, 6(2):199–223.

Karen Spärck Jones. 2003. Is question answering a rational task? In *Proceedings of the 2nd CoLogNET-ElsNET Symposium on Questions and Answers: Theoretical and Applied Perspectives.*

Stefanie Tellex, Boris Katz, Jimmy Lin, Gregory Marton, and Aaron Fernandes. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003).*

Ellen M. Voorhees. 2003. Overview of the TREC 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003).*

Ellen M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Proceedings of the Thirteenth Text REtrieval Conference (TREC 2004).*

Ellen M. Voorhees. 2005. Using question series to evaluate question answering system effectiveness. In *Proceedings of the 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005).*

Ryen W. White, Ian Ruthven, Joemon M. Jose, and C. J. van Rijsbergen. 2005. Evaluating implicit feedback models using searcher simulations. *ACM Transactions on Information Systems*, 23(3):325–361.