Is Question Answering Better Than Information Retrieval? Towards a Task-Based Evaluation Framework for 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

This paper introduces a novel evaluation framework for question series and employs it to explore the effectiveness of QA and IR systems at addressing users' information needs. The framework is based on the notion of recall curves, which characterize the amount of relevant information contained within a fixed-length text segment. Although it is widely assumed that QA technology provides more efficient access to information than IR systems, our experiments show that a simple IR baseline is quite competitive. These results help us better understand the role of NLP technology in QA systems and suggest directions for future research.

1 Introduction

The emergence of question answering (QA) has been driven to a large extent by its intuitive appeal. Instead of "hits", QA technology promises to deliver "answers", obviating the user from the tedious task of sorting through lists of potentially-relevant documents. The success of factoid QA systems, particularly in the NIST-sponsored TREC evaluations (Voorhees, 2003), has reinforced the perception about the superiority of QA systems over traditional IR engines.

However, is QA really better than IR? This work challenges existing assumptions and critically examines this question, starting with the development of a novel evaluation framework that better models user tasks and preferences. The framework is then applied to compare top TREC QA systems against an off-the-shelf IR engine. Surprisingly, experiments show that the IR baseline is quite competitive. These results help us better understand the added value of NLP technology in QA systems, and are also useful in guiding future research.

2 Evolution of QA Evaluation

Although most question answering systems rely on information retrieval technology, there has always been the understanding that NLP provides significant added value beyond simple IR. Even the earliest open-domain factoid QA systems, which can be traced back to the late nineties (Voorhees and Tice, 1999), demonstrated the importance and impact of linguistic processing. Today's top systems deploy a wide range of advanced NLP technology and can answer over three quarters of factoid questions in an open domain (Voorhees, 2003). However, present QA evaluation methodology does not take into account two developments, discussed below.

First, despite trends to the contrary in TREC evaluations, users don't actually like or want exact answers. Most question answering systems are designed to pinpoint the exact named entity (person, date, organization, etc.) that answers a particular question—and the development of such technology has been encouraged by the setup of the TREC QA tracks. However, a study by Lin et al. (2003) shows that users actually prefer answers embedded within some sort of context, e.g., the sentence or the paragraph that the answer was found in. Context pro-

Proceedings of NAACL HLT 2007, pages 212–219,
Rochester, NY, April 2007. ©2007 Association for Computational Linguistics

3. Hale Bopp comet	
1. fact	When was the comet discovered?
2. fact	How often does it approach the earth?
3. list	In what countries was the comet visible on its last return?
4. other	

68. Port Arthur Massacre	
1. fact	Where is Port Arthur?
2. fact	When did the massacre occur?
3. fact	What was the final death toll of the massacre?
4. fact	Who was the killer?
5. fact	What was the killer's nationality?
6. list	What were the names of the victims?
7. list	What were the nationalities of the victims?
8. other	

Table 1: Sample question series.

vides a means by which the user can establish the credibility of system responses and also provides a vehicle for "serendipitous knowledge discovery"—finding answers to related questions. As the early TRECs have found (Voorhees and Tice, 1999), locating a passage that contains an answer is considerably easier than pinpointing the exact answer. Thus, real-world user preferences may erode the advantage that QA has over IR techniques such as passage retrieval, e.g., (Zobel et al., 1995; Tellex et al., 2003).

Second, the focus of question answering research has shifted away from isolated factoid questions to more complex information needs embedded within a broader context (e.g., a user scenario). Since 2004, the main task at the TREC QA tracks has consisted of question series organized around topics (called "targets")—which can be people, organizations, entities, or events (Voorhees, 2004; Voorhees, 2005). 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." See Table 1 for examples of question series. Separately, NIST has been exploring other types of complex information needs, for example, the relationship task in TREC 2005 and the ciQA (complex, interactive Question Answering) task in TREC 2006 (Dang et al., 2006). One shared feature of these complex questions is that they cannot be answered by simple named entities. Answers usually span passages, which makes the task very similar to the query-focused summarization task in DUC (Dang, 2005). On these tasks, it is unclear whether QA systems actually outperform baseline IR methods. As one bit of evidence, in TREC 2003, a simple IR-based sentence ranker outperformed all but the best system on definition questions, the precursor to current "other" questions (Voorhees, 2003).

We believe that QA evaluation methodology has lagged behind these developments and does not adequately characterize the performance of current systems. In the next section, we present an evaluation framework that takes into account users' desire for context and the structure of more complex QA tasks. Focusing on question series, we compare the performance of top TREC systems to a baseline IR engine using this evaluation framework.

3 An Evaluation Framework

Question series in TREC represent an attempt at modeling information-seeking dialogues between a user and a system (Kato et al., 2004). Primarily because dialogue systems are difficult to evaluate, NIST has adopted a setup in which individual questions are evaluated in isolation—this implicitly models a user who types in a question, receives an answer, and then moves on to the next question in the series. Component scores are aggregated using a weighted average, and no attempt is made to capture dependencies across different question types.

Simultaneously acknowledging the challenges in evaluating dialogue systems and recognizing the similarities between complex QA and query-focused summarization, we propose an alternative framework for QA evaluation that considers the quality of system responses as a whole. Instead of generating individual answers to each question, a system might alternatively produce a segment of text (i.e., a summary) that attempts to answer *all* the questions. This slightly different conception of QA brings it into better alignment with recent trends in multi-document summarization, which may yield previously untapped synergies (see Section 7).

To assess the quality of system responses, we adopt the nugget-based methodology used previously for many types of complex questions (Voorhees, 2003), which shares similarities with the pyramid evaluation scheme used in summarization (Nenkova and Passonneau, 2004). A nugget can be described as an "atomic fact" that addresses an aspect of an information need. Instead of the standard nugget F-score, which hides important tradeoffs between precision and recall, we propose to measure nugget recall as a function of response length. The goal is to quantify the number of relevant facts that a user will have encountered after reading a particular amount of text. Intuitively, we wish to model how quickly a hypothetical user could "learn" about a topic by reading system responses.

Within this framework, we compared existing TREC QA systems against an IR baseline. Processed outputs from the top-ranked, second-ranked, third-ranked, and median runs in TREC 2004 and TREC 2005 were compared to a baseline IR run generated by Lucene, an off-the-shelf open-source IR engine. Our experiments focused on factoid and "other" questions; as the details differ for these two types, we describe each separately and then return to a unified picture.

4 Factoid Series

Our first set of experiments focuses on the factoid questions within a series. In what follows, we describe the data preparation process, the evaluation methodology, and experimental results.

4.1 Data Preparation

We began by preparing answer responses from the top-ranked, second-ranked, third-ranked, and median runs from TREC 2004 and TREC 2005.[1] Consider the third-ranked run from TREC 2004 as a running example; for the two factoid questions in target 3 (Table 1), the system answers were "July 22, 1995" and "4,200 years" (both correct).

Since Lin et al. (2003) suggest that users prefer answers situated within some sort of context, we projected these exact answers onto their source sentences. This was accomplished by selecting the first sentence in the source document (drawn from the AQUAINT corpus) that contains the answer string.[2] In our example, this procedure yielded the following text segment:

> The comet was named after its two observers—two amateur astronomers in the United States who discovered it on July 22, 1995. Its visit to the solar system—just once every 4,200 years, will give millions of people a rare heavenly treat when it reaches its full brightness next year.

Since projected sentences are simply concatenated, the responses often exhibit readability problems (although by chance this particular response is relatively coherent). Nevertheless, one might imagine that such output forms the basis for generating coherent query-focused summaries with sentence-rewrite techniques, e.g., (Barzilay et al., 1999). In this work, we set aside problems with fluency since our evaluation framework is unable to measure this (desirable) characteristic.

System responses were prepared for four runs from TREC 2004 and four runs from TREC 2005 in the manner described above. As a baseline, we employed Lucene to retrieve the top 100 documents from the AQUAINT corpus using the target as the query (in our example, "Hale Bopp comet"). From the result set, we retained all sentences that contain at least a term from the target. Sentence order within each document and across the ranked list was preserved. Answer responses for this baseline condition were limited to 10,000 characters. Following TREC convention, all character counts include only non-whitespace characters. Finally, since responses prepared from TREC runs were significantly shorter than this baseline condition, the baseline Lucene response was appended to the end of each TREC run to fill a quota of 10,000 characters.

4.2 Evaluation Methodology

Our evaluation framework is designed to measure the amount of useful information contained in a system response. For factoid series, this can be quan-

[1] In cases where teams submitted multiple runs, we considered only the best performing of each.

[2] As a backoff, if the exact answer string is not found in the text, the sentence with the most terms in common with the answer string is selected.

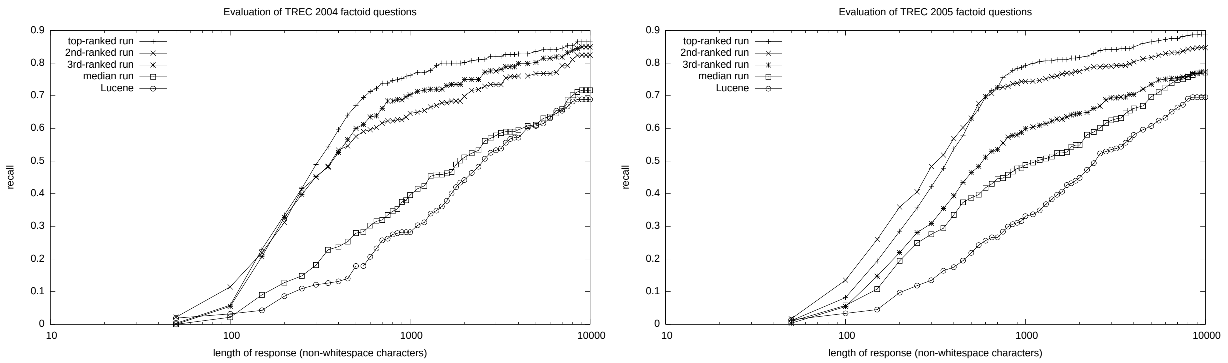

Figure 1: Factoid recall curves for runs from TREC 2004 (left) and TREC 2005 (right).

Run	**2004**	**2005**
top-ranked run	0.770	0.713
2nd-ranked run	0.643	0.666
3rd-ranked run	0.626	0.326
median run	0.170	0.177

Table 2: Official scores of selected TREC 2004 and TREC 2005 factoid runs.

tified by recall—the fraction of questions within a series whose answers could be found within a given passage. By varying the passage length, we can characterize systems in terms of recall curves that represent how quickly a hypothetical user can "learn" about the target. Below, we describe the implementation of such a metric.

First, we need a method to automatically determine if an answer string is contained within a segment of text. For this, regular expression answer patterns distributed by NIST were employed—they have become a widely-accepted evaluation tool.

Second, we must determine when a fact is "acquired" by our hypothetical user. Since previous studies suggest that context is needed to interpret an answer, we assess system output on a sentence-by-sentence basis. In our example, the lengths of the two sentences are 105 and 130 characters, respectively. Thus, for this series, we obtain a recall of 0.5 at 105 characters and 1.0 at 235 characters.

Finally, we must devise a method for aggregating across different question series to factor out variations. We accomplish this through interpolation, much in the same way that precision–recall curves are plotted in IR experiments. First, all lengths are interpolated to their nearest larger fifty character increment. In our case, they are 150 and 250. Once this is accomplished for each question series, we can directly average across all question series at each length increment. Plotting these points gives us a recall-by-length performance curve.

4.3 Results

Results of our evaluation are shown in Figure 1, for TREC 2004 (left) and TREC 2005 (right). These plots have a simple interpretation—curves that rise faster and higher represent "better" systems. The "knee" in some of the curves indicate approximately the length of the original system output (recall that the baseline Lucene run was appended to each TREC run to produce responses of equal lengths). For reference, official factoid scores of the same runs are shown in Table 2.

Results from TREC 2004 are striking: while the top three systems appear to outperform the baseline IR run, it is unclear if the median system is better than Lucene, especially at longer response lengths. This suggests that if a user wanted to obtain answers to a series of factoid questions about a topic, using the median QA system isn't any more efficient than simply retrieving a few articles using an IR engine and reading them. Turning to the 2005 results, the median system fares better when compared to the IR baseline, although the separation between the top and median systems has narrowed.

In the next two sections, we present additional experiments on question series. A detailed analysis is saved for Section 7.

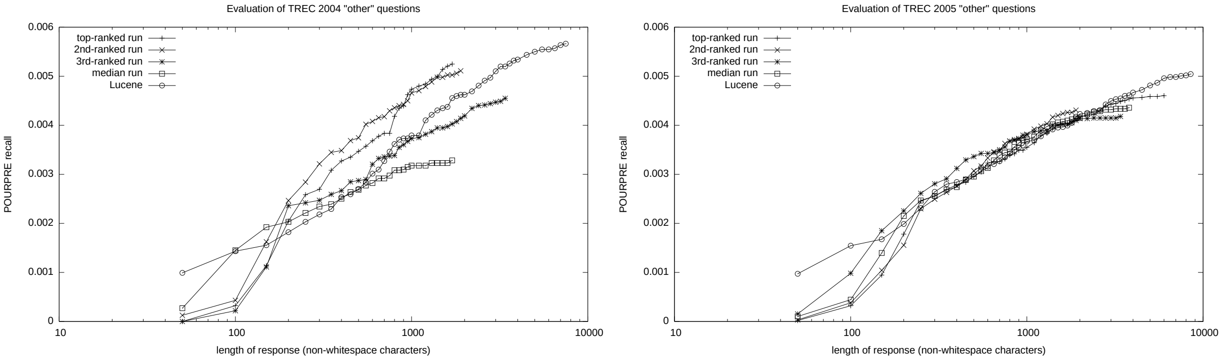

Figure 2: POURPRE recall curves for "other" runs from TREC 2004 (left) and TREC 2005 (right).

Run	2004	2005
top-ranked run	0.460	0.248
2nd-ranked run	0.404	0.232
3rd-ranked run	0.367	0.228
median run	0.197	0.152

Table 3: Official scores of selected TREC 2004 and TREC 2005 "other" runs.

5 "Other" Questions

Our second set of experiments examine the performance of TREC systems on "other" questions. Once again, we selected the top-ranked, second-ranked, third-ranked, and median runs from TREC 2004 and TREC 2005. Since system submissions were already passages, no additional processing was necessary. The IR baseline was exactly the same as the run used in the previous experiment. Below, we describe the evaluation methodology and results.

5.1 Evaluation Methodology

The evaluation of "other" questions closely mirrors the procedure developed for factoid series. We employed POURPRE (Lin and Demner-Fushman, 2005), a recently developed method for automatically evaluating answers to complex questions. The metric relies on *n*-gram overlap as a surrogate for manual nugget matching, and has been shown to correlate well with official human judgments. We modified the POURPRE scoring script to return only the nugget recall (of vital nuggets only).

Formally, systems' responses to "other" questions consist of unordered sets of answer strings. We decided to break each system's response into individual answer strings and compute nugget recall on a string-by-string basis. Since these answer strings are for the most part sentences, results are comparable to the factoid series experiments. Taking answer strings as the basic response unit also makes sense because it respects segment boundaries that are presumably meaningful to the original systems.

Computing POURPRE recall at different response lengths yielded an uninterpolated data series for each topic. Results across topics were aggregated in the same manner as the factoid series: first by interpolating to the nearest larger fifty-character increment, and then averaging all topics across each length increment.[3]

5.2 Results

Results of our experiment are shown in Figure 2. For reference, the official nugget F-scores of the TREC runs are shown in Table 3. Most striking is the observation that the baseline Lucene run is highly competitive with submitted TREC systems. For TREC 2004, it appears that the IR baseline outperforms all but the top two systems at higher recall levels. For TREC 2005, differences between all the analyzed runs are difficult to distinguish. Although scores of submitted runs in TREC 2005 were more tightly clustered, the strong baseline IR performance is surprising. For "other" questions, it doesn't appear that QA is better than IR!

We believe that relative differences in QA and IR

[3] It is worth noting that this protocol treats the answer strings as if they were ordered—but we do not believe this has an impact on the results or our conclusions.

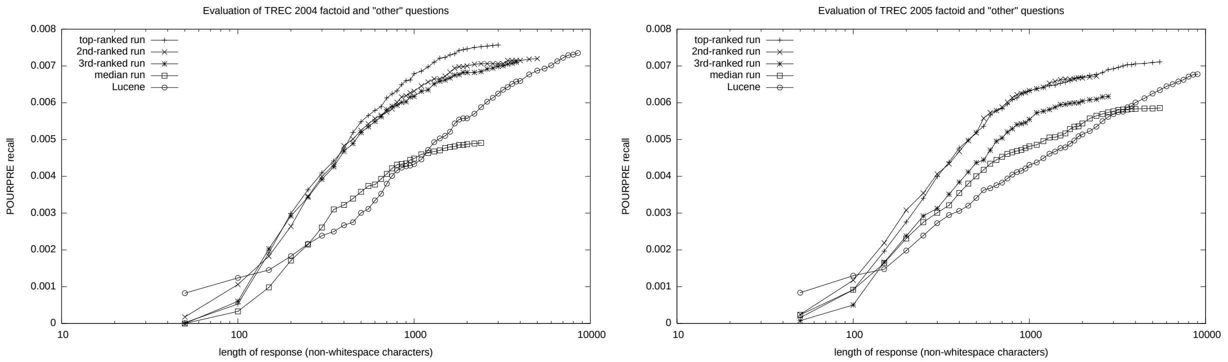

Figure 3: POURPRE recall curves for runs from TREC 2004 (left) and TREC 2005 (right), combining both factoid and "other" questions.

performance between the 2004 and 2005 test sets can be attributed to the nature of the targets. In TREC 2005, allowable semantic categories of targets were expanded to include events such as "Miss Universe 2000 crowned", which by their very nature are narrower in scope. This, combined with many highly-specific targets, meant that the corpus contained fewer topically-relevant documents for each target to begin with. As a result, an IR-based sentence extraction approach performs quite well—this explanation is consistent with the observations of Lin and Demner-Fushman (2006).

6 Combining Question Types

In the previous two sections, factoid and "other" questions were examined in isolation, which ignores their complementary role in supplying information about a target. To provide a more complete picture of system performance, we devised a method by which both question types can be evaluated together.

At the conceptual level, there is little difference between factoid and "other" questions. The first type asks for explicit facts, while the second type asks for facts that the user didn't know enough to ask about directly. We can unify the evaluation of both types by treating regular expression factoid patterns as if they were (vital) nuggets. Many patterns don't contain any special symbols, and read quite like nugget descriptions already. For others, we manually converted regular expressions into plain text, e.g., "(auto|car) crash" becomes "auto car crash".

To validate this method, we first evaluated factoid series using POURPRE, with nugget descriptions prepared from answer patterns in the manner described above. For both TREC 2004 and TREC 2005, we did not notice any qualitative differences in the results, suggesting that factoid answers can indeed be treated like nuggets.

We then proceeded to evaluate both factoid and "other" questions together using the above procedure. Runs were prepared by appending the 1st "other" run to the 1st factoid run, the 2nd "other" run to the 2nd factoid run, etc.[4] The Lucene baseline run remained the same as before.

Plots of POURPRE recall by answer length are shown in Table 3. These graphs provide a more complete picture of QA performance on question series. The same trends observed in the two previous experiments are seen here also: it does not appear that the median run in TREC 2004 performs any better than the IR baseline. Considering the TREC 2005 runs, the IR baseline remains surprisingly competitive.

Note that integration of list questions, the third component of question series, remains a challenge. Whereas the answer to a factoid question can be naturally viewed as a vital nugget describing the target, the relative importance of a single answer instance to a list question cannot be easily quantified. We leave this issue for future work.

7 Discussion

It can be argued that quantitative evaluation is the single most important driver for advancing the state

[4] Note that we're mixing sections from different runs, so these do not correspond to any actual TREC submissions.

of the art in language processing technology today. As a result, evaluation metrics and methodologies need to be carefully considered to insure that they provide proper guidance to researchers. Along these lines, this paper makes two arguments: that recall curves better capture aspects of complex QA tasks than the existing TREC evaluation metrics; and that this novel evaluation framework allows us to explore the relationship between QA and IR technology in a manner not possible before.

7.1 Advantages of Recall Curves

We see several advantages to the evaluation framework introduced here, beyond those already discussed in Sections 2 and 3.

Previously, QA and IR techniques were not directly comparable since they returned different response units. To make evaluation even more complex, different types of questions (e.g., factoid vs. "other") require different metrics—in TREC, these incomparable values were then aggregated based on arbitrary weights to produce a final composite score. By noting similarities between factoid answers and nuggets, we were able to develop a unified evaluation framework for factoid and "other" questions. By emphasizing the similarities between complex QA and summarization, it becomes possible to compare QA and IR technology directly—this work provides a point of reference much in the same way that IR-based sentence extraction has served as a starting point for summarization research, e.g., (Goldstein et al., 1999).

In addition, characterizing system performance in terms of recall curves allows researchers to compare the effectiveness of systems under different task models. Measuring recall at short response lengths might reflect time-constrained scenarios, e.g., producing an action-oriented report with a 30-minute deadline. Measuring recall at longer response lengths might correspond to in-depth research, e.g., writing a summary article due by the end of the day. Recall curves are able to capture potential system tradeoffs that might otherwise be hidden in single-point metrics.

7.2 Understanding QA and IR

Beyond answering a straightforward question, the results of our experiments yield insights about the relationship between QA and IR technology.

Most question answering systems today employ a two-stage architecture: IR techniques are first used to select a candidate set of documents (or alternatively, passages, sentences, etc.), which is then analyzed by more sophisticated NLP techniques. For factoids, analysis usually involves named-entity recognition using some sort of answer type ontology; for "other" questions, analysis typically includes filtering for definitions based on surface patterns and other features. The evaluation framework described in this paper is able to isolate the performance contribution of this second NLP stage—which corresponds to the difference between the baseline IR and QA recall curves.

For factoid questions, NLP technology provides a lot of added value: the set of techniques developed for pinpointing exact answers allows users to acquire information more quickly than they otherwise could with an IR system (shown by Figure 1). The added value of NLP techniques for answering "other" questions is less clear—in many instances, those techniques do not appear to be contributing much (shown by Figure 2). Whereas factoid QA technology is relatively mature, researchers have made less progress in developing general techniques for answering complex questions.

Our experiments also illuminate when exactly QA works. For short responses, there is little difference between QA and IR, or between all QA systems for that matter, since it is difficult to cram much information into a short response with current (extractive) technology. For extremely long responses, the advantages provided by the best QA systems are relatively small, since there's an upper limit to their accuracy (and researchers have yet to develop a good backoff strategy). In the middle range of response lengths is where QA technology really shines—where a user can much more effectively gather knowledge using a QA system.

7.3 Implications for Future Research

Based on the results presented here, we suggest two future directions for the field of question answering.

First, we believe there is a need to focus on answer generation. High-precision answer extraction alone isn't sufficient to address users' complex information needs—information nuggets must be synthesized and presented for efficient human consumption. The coherence and fluency of system responses should be factored into the evaluation methodology as well. In this regard, QA researchers have much to learn from the summarization community, which has already grappled with these issues.

Second, more effort is required to developed task-based QA evaluations. The "goodness" of answers can only be quantified with respect to a task—examples range from winning a game show (Clarke et al., 2001) to intelligence gathering (Small et al., 2004). It is impossible to assess the real-world impact of QA technology without considering how such systems will be used to solve human problems. Our work takes a small step in this direction.

8 Conclusion

Is QA better than IR? The short answer, somewhat to our relief, is *yes*. But this work provides more than a simple affirmation. We believe that our contributions are two-fold: a novel framework for evaluating QA systems that more realistically models user tasks and preferences, and an exploration of QA and IR performance within this framework that yields new insights about these two technologies. We hope that these results are useful in guiding the development of future question answering systems.

9 Acknowledgments

This work has been supported in part by DARPA contract HR0011-06-2-0001 (GALE), and has benefited from discussions with Ellen Voorhees and Hoa Dang. I would also like to thank Esther and Kiri for their loving support.

References

Regina Barzilay, Kathleen R. McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proc. of ACL 1999*.

Charles Clarke, Gordon Cormack, and Thomas Lynam. 2001. Exploiting redundancy in question answering. In *Proc. of SIGIR 2001*.

Hoa Trang Dang, Jimmy Lin, and Diane Kelly. 2006. Overview of the TREC 2006 question answering track. In *Proc. of TREC 2006*.

Hoa Dang. 2005. Overview of DUC 2005. In *Proc. of DUC 2005*.

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proc. of SIGIR 1999*.

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 *Proc. 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 *Proc. of HLT/EMNLP 2005*.

Jimmy Lin and Dina Demner-Fushman. 2006. Will pyramids built of nuggets topple over? In *Proc. of 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 *Proc. of INTERACT 2003*.

Ani Nenkova and Rebecca Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proc. of HLT/NAACL 2004*.

Sharon Small, Tomek Strzalkowski, Ting Liu, Sean Ryan, Robert Salkin, Nobuyuki Shimizu, Paul Kantor, Diane Kelly, Robert Rittman, and Nina Wacholder. 2004. HITIQA: towards analytical question answering. In *Proc. of COLING 2004*.

Stefanie Tellex, Boris Katz, Jimmy Lin, Gregory Marton, and Aaron Fernandes. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proc. of SIGIR 2003*.

Ellen M. Voorhees and Dawn M. Tice. 1999. The TREC-8 question answering track evaluation. In *Proc. of TREC-8*.

Ellen M. Voorhees. 2003. Overview of the TREC 2003 question answering track. In *Proc. of TREC 2003*.

Ellen M. Voorhees. 2004. Overview of the TREC 2004 question answering track. In *Proc. of TREC 2004*.

Ellen M. Voorhees. 2005. Using question series to evaluate question answering system effectiveness. In *Proc. of HLT/EMNLP 2005*.

Justin Zobel, Alistair Moffat, and Ross Wilkinson Ron Sacks-Davis. 1995. Efficient retrieval of partial documents. *IPM*, 31(3):361–377.