Natural Language Engineering **1** (1): 000–000. Printed in the United Kingdom

The TREC Question Answering Track

Ellen M. Voorhees
National Institute of Standards and Technology
Gaithersburg, MD 20899

(*Received 20 August 2001*)

Abstract

The Text REtrieval Conference (TREC) question answering track is an effort to bring the benefits of large-scale evaluation to bear on a question answering (QA) task. The track has run twice so far, first in TREC-8 and again in TREC-9. In each case the goal was to retrieve small snippets of text that contain the actual answer to a question rather than the document lists traditionally returned by text retrieval systems. The best performing systems were able to answer about seventy per cent of the questions in TREC-8 and about sixty-five per cent of the questions in TREC-9. While the sixty-five per cent score is a slightly worse result than the TREC-8 scores in absolute terms, it represents a very significant improvement in question answering systems. The TREC-9 task was considerably harder than the TREC-8 task because TREC-9 used actual users' questions while TREC-8 used questions constructed for the track. Future tracks will continue to challenge the QA community with more difficult, and more realistic, question answering tasks.

1 Introduction

The Text REtrieval Conference (TREC) is a workshop series designed to encourage research on text retrieval for realistic applications by providing large test collections, uniform scoring procedures, and a forum for organizations interested in comparing results (Voorhees 2000a). Started in 1992, the conference is co-sponsored by the National Institute of Standards and Technology (NIST) and the U.S. Department of Defense. The workshops have focused primarily on the traditional information retrieval problem of retrieving a ranked list of documents in response to a statement of information need (called a topic in TREC). For each TREC, NIST provides a test set of documents and topics. Participants run their retrieval systems on the data, and return to NIST a list of the top-ranked retrieved documents. NIST pools the individual results, judges the retrieved documents for correctness, and evaluates the results. The annual TREC cycle ends with a workshop at which participants share their experiences.

In addition to the main document retrieval task, TREC contains other tasks, called tracks, that focus on new areas or particularly difficult aspects of information retrieval. TRECs 8 and 9, the two most recent TRECs, sponsored a question answering track to foster research on the problem of retrieving answers rather than document lists, the assumption being that users would usually prefer to be given the answer rather than find the answer themselves in a document. In particular, the track focused on the problem of returning text snippets drawn from a large corpus of newspaper articles in response to fact-based, short-answer questions.

This paper reviews the findings of the TREC question answering track to date and discusses the directions future tracks will take. The next section situates the specific track task in the context of other question answering work. It also provides all the particulars regarding the data and evaluation methodology used in the track. Section 3 summarizes the results of the track and the methods used to achieve those results. Since one of the main benefits of TREC has been the construction of large document retrieval test collections, Section 4 examines the issue of building an equivalent reusable test collection for the QA task. The final section outlines new tasks to be included in future tracks.

2 Question Answering

"Question answering" covers a broad range of activities from simple yes/no responses for true-false questions to the presentation of complex results synthesized from multiple data sources. Question answering systems with widely varying capabilities have been developed, depending on different assumptions as to the precise task the system should perform and the resources available to it. In this section we briefly list some of the approaches taken to question answering in the past, and then give a detailed description of the task in the TREC QA track.

2.1 Previous work

The first question answering computer systems were developed as vehicles for natural language understanding research. For example, one of the earliest computer-based question answering systems, STUDENT, read and solved high school algebra word problems (Winograd 1977). Correctly solving the algebra problem was taken as a demonstration that the system understood the written statement of the problem. Since understanding language requires world knowledge, systems were limited by the amount of knowledge they contained. Thus Winograd's SHRDLU system was constrained to a simple block world (Winograd 1977), while the LUNAR system allowed geologists to ask questions about moon rocks (Woods 1977). The LUNAR system is notable in that it was the subject of one of the first user evaluations of question answering systems. LUNAR was demonstrated at the Second Annual Lunar Science Conference in January 1971, and geologists were encouraged to ask it questions. Of the 111 questions that were within the scope of the moon rock data (and were not comparatives), seventy-eight per cent were answered correctly, tewlve per cent failed for "clerical" reasons, and ten per cent had more serious errors.

Knowledge-intensive question answering systems continue to be developed as the result of natural language understanding research, but have also been developed to accomplish particular tasks (Webber 1987). The LUNAR system was an early example of natural language front-ends to database systems such as Microsoft's English

Query[1] (http://www.microsoft.com/technet/sql/engquer.asp) or ELF Software's Access ELF (http://www.elfsoftware.com/home.htm). Question answering is also the main method of interacting with expert systems, both to pose the problem to be solved and to view the system's justification for its response. The systems developed within the DARPA High-Performance Knowledge Bases Project (HPKB) are recent examples of systems designed to answer complex questions within a narrow domain (Cohen *et al.* 1998), though even here different parts of the system place different burdens on the knowledge base. For example, the START system can answer simpler questions using knowledge bases mined from the World Wide Web (Katz 1997).

In a separate body of QA research, no attempt was made to have systems understand text. Instead, the goal was to extract a small piece of text that answers the user's question from a much larger body of text. These systems do not rely on a knowledge-base, and are therefore domain-independent, but they do depend on the answer being present in the text that is searched. O'Connor described a method for retrieving "answer-passages" at a time that most commercial retrieval systems were returning bibliographic references (O'Connor 1980). Kupiec's MURAX system used an on-line encyclopedia as a source of answers for *closed-class* questions, which Kupiec defined as "a question stated in natural language, which assumes some definite answer typified by a noun phrase rather than a procedural answer" (Kupiec 1993). The FAQ Finder system used files containing question and answer pairs as developed for human readers of Usenet news groups to answer users' questions (Burke *et al.* 1997).

Information extraction (IE) systems—such as those used in the Message Understanding Conferences (MUCs, see http://www.itl.nist.gov/iad/894.02/related_projects/muc/)—recognize particular kinds of entities and relationships among those entities in running text. While not strictly question-answering systems, the goal of IE systems is to populate database-like tables with the extracted data to facilitate future tasks such as data mining, summarization, or question answering. Like traditional QA systems, IE systems generally depend on domain knowledge to find appropriate text extracts. However, the trend in IE research has been toward more shallow and less domain-dependent techniques (see, for example, the FASTUS (Appelt *et al.* 1995) or PLUM (BBN 1995) systems).

2.2 The TREC QA task

The goal of the TREC QA track is to foster research that will move retrieval systems closer to information retrieval as opposed to document retrieval. Document retrieval systems' ability to work in any domain was considered an important feature to maintain. In addition, the technology that had been developed by the information extraction community appeared ready to exploit. Thus the task used

[1] Products are given as examples only. The inclusion or omission of a particular company or product implies neither endorsement nor criticism by NIST.

- How many calories are there in a Big Mac?
- What two US biochemists won the Nobel Prize in medicine in 1992?
- Who is the voice of Miss Piggy?
- Where is the Taj Mahal?
- What costume designer decided that Michael Jackson should only wear one glove?
- In what year did Joe DiMaggio compile his 56-game hitting streak?
- What language is commonly used in Bombay?
- Where is Rider College located?
- Name a film in which Jude Law acted.

Fig. 1. Example questions used in the question answering track.

in the track was defined such that both the information retrieval and the information extraction communities could work on a common problem. The task was very similar to the MURAX system's task except that the answers were to be found in a large corpus of documents rather than an encyclopedia. Since the documents consisted mostly of newswire and newspaper articles, the domain was essentially unconstrained. However, only closed-class questions were used, so answers were generally entities familiar to IE systems.

Participants were given a document collection and a test set of questions. The test was a blind test: participants were not allowed to change their systems once they received the questions, and the answers were required to be produced completely automatically. The questions were generally fact-based, short-answer questions as shown in Figure 1. Each question was guaranteed to have at least one document in the collection that explicitly answered it.

Participants returned a ranked list of five [*document-id*, *answer-string*] pairs per question such that each answer string was believed to contain an answer to the question. Answer strings were limited to either 50 or 250 bytes depending on the run type, and could either be extracted from the corresponding document or automatically generated from information contained in the document. Human assessors read each string and decided whether the string actually did contain an answer to the question in the context provided by the document. Taking document context into account allowed a system that correctly derived a response from a document that was in error (for example, a typographical error or a misreported fact such as an incorrect number of casualties) to be given full credit for its response.

Given a set of judgments for the strings, the score computed for a submission was the mean reciprocal rank (MRR). An individual question received a score equal to the reciprocal of the rank at which the first correct response was returned, or zero if none of the five responses contained a correct answer. The score for a submission was then the mean of the individual questions' reciprocal ranks. The reciprocal rank has several advantages as a scoring metric. It is closely related to the average precision measure used extensively in document retrieval. It is bounded between zero and one, inclusive, and averages well. A run is penalized for not retrieving any correct answer for a question, but not unduly so. However, the measure also

Table 1. *Data used in the two TREC question answering tracks.*

	TREC-8	TREC-9
number of documents	528,000	979,000
megabytes of document text	1904	3033
number of questions released	200	693
number of questions evaluated	198	682
document sources	TREC disks 4–5: LA Times, Financial Times, FBIS, Federal Register	news from TREC disks 1–5: AP newswire, Wall Street Journal, San Jose Mercury News, Financial Times, LA Times, FBIS
question sources	FAQ Finder log, assessors, participants	Encarta log, Excite log

has some drawbacks. The score for an individual question can take on only six values (0, .2, .25, .33, .5, 1). Question answering systems are given no credit for retrieving multiple (different) correct answers. Also, since the track required at least one response for each question, systems could receive no credit for realizing they did not know the answer.

2.2.1 The data

While the same basic task was performed in both the TREC-8 and TREC-9 tracks, there were some differences between the tracks. Both the document set and the test set of questions was larger for TREC-9, as shown in Table 1. A more substantive difference was the source of the questions used in each TREC. Some of the questions used in TREC-8 were drawn from a log of questions submitted to the FAQ Finder system, but many of the questions in the log did not have answers in the document collection. As a result, the majority of questions used in TREC-8 were developed by either the participants or NIST assessors specifically for the track. Questions created for the track were often back-formulations of statements in the documents, which made the questions somewhat unnatural and also made the task easier since the target document contained most of the question words. For the TREC-9 track, NIST obtained two query logs and used those as a source of questions. An Encarta log, made available to NIST by Microsoft, contained grammatical questions. The other log was a log of queries submitted to the Excite search engine on December 20, 1999. Since the Excite log contains relatively few grammatically well-formed questions, the log was used as a source of ideas for NIST staff who created well-formed questions from query words without referring to the document collection. NIST assessors then checked whether each candidate question had an answer in the document collection, and a candidate question was discarded if no answer was found.

The TREC-9 question set contained 500 questions drawn from the logs, plus an additional 193 questions that were syntactic variants of an original question. The purpose of the the syntactic variants was to investigate whether QA systems are robust to the variety of different ways a question can be phrased. Once the first 500 questions were selected, NIST assessors were given a subset of the questions and asked to create "natural" variants of the question. The intent was that the variant should have the same semantic meaning of the original, as well as be phrased in a way that a native English speaker might ask the question. For example, the test set contained four variants for the question *What is the tallest mountain?*: *What is the world's highest peak?*, *What is the highest mountain in the world?*, *Name the highest mountain.*, and *What is the name of the tallest mountain in the world?*. The 193 variants included variants for fifty-four different original questions, with a range of one to seven new questions per original.

Despite the care taken to select questions that had answers in the document collection, some questions were removed from the test set after the human assessors judged the submissions because the guarantee of an answer in the document collection could no longer be made. Two questions were removed from the TREC-8 evaluation and eleven questions from the TREC-9 evaluation. Most of the eleven questions that were removed from the TREC-9 evaluation set were variants for which the rewording changed the focus of the original and resulted in a question that had no answer. The remaining TREC-9 questions and the two TREC-8 questions were removed because the assessor who did the judging disagreed with the answer that was accepted during the candidate verification phase.

2.2.2 Answer assessment

In many evaluations of natural language processing tasks, application experts create a gold-standard answer key that is assumed to contain all possible correct responses. An absolute score for a system's response is computed by measuring the difference between the response and the answer key. For text retrieval, however, different people are known to have different opinions about whether or not a given document should be retrieved for a query (Schamber 1994), so a single list of correct (or "relevant") documents cannot be created. Instead, the list of relevant documents produced by one person (the assessor) is used as an example of a correct response, and systems are evaluated using one example set of judgments per query. While the absolute scores of systems change when different assessors' opinions are used, relative scores generally remain stable, so scores computed using judgments from just one assessor per query are valid for comparing different retrieval techniques (Voorhees 2000b).

A sub-goal of the TREC-8 QA track was to investigate whether different people have different opinions as to what constitutes an acceptable answer, and, if so, how those differences affect QA evaluation. To accomplish this goal, each question was independently judged by three different assessors. The separate judgments were combined into a single judgment set through adjudication for the official track evaluation, but the individual judgments were used to measure the effect of differences in judgments on systems' scores.

Assessors were trained for the QA task before they did any judging. The purpose of the training was to motivate the assessors' task and provide general guidance on the issues that would arise during assessing rather than to drill the assessors on a specific set of assessment rules. To begin, each assessor was given the following instructions.

> Assume there is a user who trusts the answering system completely, and therefore does not require that the system provide justification in its answer strings. Your job is to take each answer string in turn and judge if this answer string alone were returned to the trustful user, would the user be able to get the correct answer to the question from the string.

Assessors then judged four sample questions whose response strings were concocted by NIST staff to illustrate various fundamentals of QA judging:

- that answer strings would contain fragments of text that were not necessarily grammatically correct;
- that the answer string did not need to contain justification;
- that the assessor was to judge the *string*, not the document from which the string was drawn;
- that document context must be taken into account; and
- that the string must be responsive to the question.

Document context was vital for questions whose answers change over time. For example, responses to questions phrased in the present tense (*Who is the prime minister of Japan?*) were judged as correct or incorrect based on the time of the document associated with the response. Requiring that the answer string be responsive to the question addressed a variety of issues. Answer strings that contained multiple entities of the same semantic category as the correct answer but did not indicate which of those entities was the actual answer (e.g., a list of names in response to a who question) were judged as incorrect. Certain punctuation and units were also required. Thus "5 5 billion" was not an acceptable substitute for "5.5 billion", nor was "500" acceptable when the correct answer was "$500". Finally, unless the question specifically stated otherwise, correct responses for questions about a famous entity had to refer to *the* famous entity and not to imitations, copies, etc. For example, two separate questions asked for the height of the Matterhorn (i.e., the Alp) and the replica of the Matterhorn at Disneyland. Correct responses for one of these questions were incorrect for the other.

The results of the TREC-8 assessment process demonstrated that assessors do have legitimate differences of opinion as to what constitutes an acceptable answer even for these deliberately constrained questions (Voorhees and Tice 2000b). Two prime examples of where such differences arise are the completeness of names and the granularity of dates and locations. For example, for the question *When did French revolutionaries storm the Bastille?*, some assessors accepted "July 14", others accepted "1789", and everyone accepted "July 14, 1789". Assumed context also mattered, and differed among assessors. For the question *Where is the Taj Mahal?* one of the three assessors accepted Atlantic City, NJ (home of the Taj Mahal casino) as an acceptable response even in light of the guidelines that stated replicas and imitations should not be used as answers for questions about a famous entity. For this assessor, the casino was sufficiently well-known to be an entity in its own right.

Fortunately, as with document retrieval evaluation, the relative scores between QA systems remain stable despite differences in the judgments used to evaluate them. Indeed, for the purpose of a comparative evaluation, the judgment sets that consisted of a single judge's opinion for each question were not only equivalent to one another, but were also equivalent to the adjudicated judgment set (Voorhees and Tice 2000b). The lack of a definitive answer key does mean that evaluation scores are only meaningful in relation to other scores on the same data set, but this is unavoidable. If assessors' opinions of correctness differ, the eventual end users of the QA systems will have similar differences of opinion, and an evaluation of the technology must accommodate these differences.

Given that single-opinion judgments are sufficient for comparative evaluations, each TREC-9 question was judged by only one assessor. (The savings resulting from only one judge per question permitted the increase from 200 to 693 questions in the test set.) Each individual variant in a variant question set was judged as a separate question, though the entire set was judged by the same assessor and variants were judged consecutively.

Another difference between the TREC-8 and TREC-9 judging was the addition of an "unsupported" judgment in TREC-9. There were a number of instances during the TREC-8 judging when an answer string contained the correct answer, but that answer could not possibly have been determined from the document returned. For example, the correct answer for *Who is the 16th President of the United States?* is Abraham Lincoln. One of the answer strings returned contained Abraham Lincoln, but the associated document discussed Lincoln's Gettysburg Address. The document does not even mention that Lincoln was president, let alone that he was the sixteenth president. Since the TREC-8 task did not specifically require that the document returned with the answer string support the string as the answer, these cases were judged as correct in TREC-8, even though the assessors were uncomfortable doing so. In TREC-9, the track guidelines required that the document returned with the answer string actually support the answer contained in the string. If the answer string did not contain a correct answer, the response was judged incorrect. If the string did contain a correct answer, but the document did not support that answer (such as the Lincoln/Gettysburg Address example above), the response was judged unsupported. Otherwise, the response was judged correct. Two scores were computed for each TREC-9 run, a *strict* score in which unsupported answers were considered incorrect, and a *lenient* score in which unsupported answers were considered correct.

3 Track Results

Both the TREC-8 and TREC-9 QA tracks offered two experimental conditions: answer strings limited to 250 bytes, and answer strings limited to 50 bytes. Participants were permitted to submit up to two runs for each condition (four runs total), where a run consisted of a ranked list of up to five [*document-id*, *answer-string*] pairs for each question in the test set.

This section lists the top ten scoring runs for each length for both tracks, and summarizes the general approach taken to the QA task. Note that the cut-off of ten was chosen for convenience; there were other runs submitted to the tracks whose scores were quite close to the ones listed here. Further, since TREC participants are free to choose which tasks they submit runs to, not all groups submitted runs for both conditions, and some groups participated in the QA track in one year only. For a more complete description of the runs submitted to the QA track, see the TREC proceedings at `http://trec.nist.gov/pubs.html`.

3.1 TREC-8 results

Twenty different organizations participated in the TREC-8 question answering track. A total of forty-five runs was submitted, twenty runs using the 50-byte limit and twenty-five runs using the 250-byte limit. Table 2 gives both the mean reciprocal rank and the number of questions for which no answer was found for the top ten runs of each length. Only one run per participant per length is shown in the table. The scores are computed over the 198 questions that comprised the official test set. The table is split between the 50-byte and the 250-byte runs and is sorted by decreasing mean reciprocal rank within run type.

The most accurate of the systems were able to answer more than two-thirds of the questions. When a system found the answer at all, it was likely to be ranked highly.

Not surprisingly, allowing 250 bytes in a response was an easier task than limiting responses to 50 bytes: for every organization that submitted runs of both lengths, the 250-byte limit run had a higher mean reciprocal rank. The submissions from AT&T Research Labs demonstrate that existing passage-retrieval techniques can be successful for 250-byte runs, but are not suitable for 50-byte runs (Singhal *et al.* 2000). Their question answering system used a traditional vector-based retrieval system to select fifty documents and then scored each sentence within those documents by the number of question words in the surrounding context. In one set of runs (their “passage-based” runs), the highest scoring sentences were returned as the response. In a second set (their “entity-based” runs), high scoring sentences were further processed by a linguistic module. The passage-based method was very competitive for the 250-byte limit, but was not nearly as successful when restricted to just 50 bytes. These results suggest that the relatively simple bag-of-words approaches that are successfully used in text retrieval are not sufficient for extracting specific, fact-based answers.

Most participants used a version of the following general approach to the question answering problem. The system first attempted to classify a question according to the type of its answer as suggested by its question word. For example, a question that begins with “who” (*Who is the prime minister of Japan?*) implies a person or an organization is being sought, and a question beginning with “when” (*When did the Jurassic Period end?*) implies a time designation is needed. Next, the system

Table 2. *Mean reciprocal rank (MRR) and number of questions for which no correct response was found (# not found) for top TREC-8 QA track submissions.*

Run Name	Participant	MRR	# not found	
textract9908	Cymfony, Inc.	.66	54	(27%)
SMUNLP1	Southern Methodist U.	.56	63	(32%)
attqa50e	AT&T Research	.36	109	(55%)
IBMDR995	IBM (Prager)	.32	110	(56%)
xeroxQA8sC	Xerox Research Centre Europe	.32	111	(56%)
umdqa	U. of Maryland	.30	118	(60%)
MTR99050	MITRE	.28	118	(60%)
nttd8qs1	NTT Data Corp.	.27	121	(61%)
CRL50	New Mexico State U.	.22	130	(66%)
INQ634	U. of Massachusetts	.19	140	(71%)
a) Runs with a 50-byte limit on the length of the response.				
SMUNLP2	Southern Methodist U.	.65	44	(22%)
attqa250p	AT&T Research	.55	63	(32%)
GePenn	GE/U. of Pennsylvania	.51	72	(36%)
uwmt9qa1	MultiText, U. of Waterloo	.47	74	(37%)
mds08q1	Royal Melbourne Inst. Tech	.45	77	(39%)
xeroxQA8lC	Xerox Research Centre Europe	.45	83	(42%)
nttd8ql1	NTT Data Corp.	.44	79	(40%)
MTR99250	MITRE	.43	86	(43%)
IBMDR992	IBM (Prager)	.43	89	(45%)
INQ635	U. of Massachusetts	.38	95	(48%)
b) Runs with a 250-byte limit on the length of the response.				

retrieved a small portion of the document collection using standard text retrieval technology and the question as the query. The system performed a shallow parse of the returned documents to detect entities of the same type as the answer. If an entity of the required type was found sufficiently close to the question's words, the system returned that entity as the response. If no appropriate answer type was found, the system fell back to best-matching-passage techniques.

This approach worked well provided the query types recognized by the system had broad enough coverage and the system was able to classify questions sufficiently accurately. Most systems could answer questions that began with "who" very accurately. However, questions that sought a person but did not actually begin with "who" (*Name the first private citizen to fly in space. What Nobel laureate was expelled from the Philippines before the conference on East Timor?*) were much more difficult. More difficult still were questions whose answers were not an entity of a specific type (*What is Head Start? Why did David Koresh ask the FBI for a word processor?*). Of course, pattern matching on expected answer types was not foolproof even when "good" matches were found. One response to the question *Who was the first American in space?* was Jerry Brown, taken from a document that says

Table 3. *Mean reciprocal rank (MRR) and number of questions for which no correct response was found (# not found) using strict evaluation for top TREC-9 QA track submissions.*

Run Name	Participant	MRR	# not found
LCCSMU2	Southern Methodist U.	0.58	229 (34%)
uwmt9qas0	MultiText, U. of Waterloo	0.32	395 (58%)
ISI0A50	ISI, U. of So. California	0.32	385 (57%)
IBMKR50	IBM (Prager)	0.32	402 (59%)
ibmhlt0050	IBM (Ittycheriah)	0.29	394 (58%)
pir0qas2	Queens College, CUNY	0.28	401 (59%)
SUT9p2c3c050	Syracuse U.	0.25	439 (64%)
ICrjc99a	Imperial College	0.23	454 (67%)
NTTD9QAa2S	NTT Data Corp.	0.23	439 (64%)
ALI9C50	U. de Alicante	0.23	451 (66%)

a) Runs with a 50-byte limit on the length of the response.

LCCSMU1	Southern Methodist U.	0.76	95 (14%)
pir0qal2	Queens College, CUNY	0.46	264 (39%)
uwmt9qal1	MultiText, U. of Waterloo	0.46	265 (39%)
ibmhlt00250	IBM (Ittycheriah)	0.46	263 (39%)
IBMKA250	IBM (Prager)	0.42	294 (43%)
lcat250	LIMSI-CNRS	0.41	307 (45%)
NTTD9QAa1L	NTT Data Corp.	0.39	299 (44%)
ICrjc99b	Imperial College	0.39	348 (51%)
SUT9p2c3c250	Syracuse U.	0.39	319 (47%)
KUQA250a	Korea U.	0.37	338 (50%)

b) Runs with a 250-byte limit on the length of the response.

```
  As for Wilson himself, he became a senator by defeating Jerry Brown, who
has been called the first American in space.
```

A similar response was returned for the question *Who wrote 'Hamlet'?*:

```
  'Hamlet,' directed by Franco Zeffirelli and written by...well, you know.
```

3.2 TREC-9 results

Twenty-eight organizations participated in the TREC-9 question answering track. A total of seventy-eight runs was submitted, thirty-four runs using the 50-byte limit and forty-four runs using the 250-byte limit. Table 3 gives the top ten results for each length for the TREC-9 track in the same format as Table 2, where the scores were computed using strict evaluation.

Two main conclusions can be drawn from Table 3: scores are generally lower than scores from TREC-8, and the best performing system (from Southern Methodist University) did substantially better than the other systems. Despite the drop in the absolute value of the evaluation scores, the performance of the TREC-9 systems represents a significant improvement in question answering technology. The switch to "real" questions, rather than questions created especially for the track, made the TREC-9 task much more difficult than the TREC-8 task. The motivation for using actual user questions was the belief that constructed questions are easier for QA systems because the question and answer document share the same vocabulary. However, the difference between the TREC-8 and TREC-9 question sets was larger than just vocabulary issues. TREC-8 questions had been restricted to those with an "obvious" answer. While the subsequent differences in opinion demonstrated that there is no such thing as an obvious answer, the questions were still far less ambiguous than the questions mined from logs. Real users ask vague questions such as *Who is Colin Powell?* and *Where do lobsters like to live?*. These questions are substantially harder for both the systems to answer and the assessors to judge.

The improvement in QA systems came from refinements to the individual steps of the general strategy used in TREC-8 rather than an entirely new approach. TREC-9 systems were better at classifying questions as to the expected answer type, and used a wider variety of methods for finding the entailed answer types in retrieved passages. Many systems used WordNet (Fellbaum 1998) as a source of related words for the initial query and as a means of determining whether an entity extracted from a passage matched the required answer type. Results from Queens College, CUNY demonstrated that high-quality document retrieval in the initial step is helpful (Kwok *et al.* 2001). This group used comparatively simple answer extraction techniques, yet performed relatively well, especially in the 250-byte condition.

The Southern Methodist University system, called Falcon, is also of this same general type (Harabagiu *et al.* 2001), though it includes successive feedback loops that try progressively larger modifications to the original question until it finds an answer that can be justified as an abductive proof. The system first parses the question and recognizes entities contained in it to create a question semantic form. The semantic form of the question is used to determine the expected answer type by finding the phrase that is most connected to other concepts in the question. The system uses an answer taxonomy that contains WordNet subhierarchies and thus has broad coverage. Falcon next retrieves paragraphs from the corpus using Boolean queries and terms drawn from the original question, related concepts from WordNet, and an indication of the expected answer type. The paragraph retrieval is repeated using different term combinations until the query returns a number of paragraphs in a pre-determined range. The retrieved paragraphs are parsed into their semantic forms, and a unification procedure is run between the question semantic form and each paragraph semantic form. If the unification fails for all paragraphs, a new set of paragraphs is retrieved using synonyms and morphological derivations of the previous query. When the unification procedure succeeds, the semantic forms are translated into logical forms, and a logical proof in the form of an abductive backchaining from the answer to the question is attempted. If the proof succeeds, the answer from the proof is returned as the answer string. Otherwise, terms that are semantically related to important question concepts are drawn from WordNet and a new set of paragraphs is retrieved.

In TREC-9 question variants were introduced into the test set to explore whether

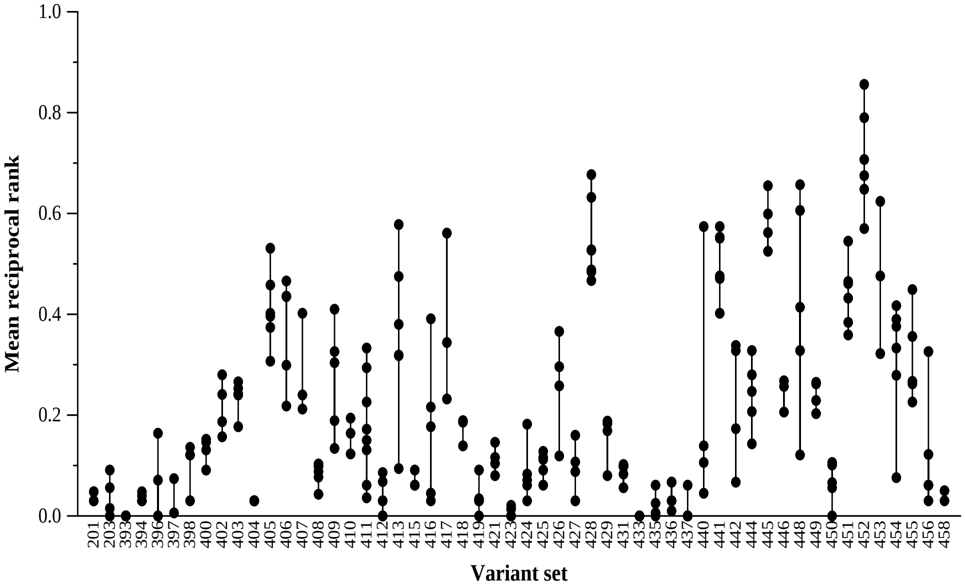

Fig. 2. Average reciprocal rank for each question variant computed over thirty-three 50-btye runs using strict evaluation. The x-axis represents different variant sets, identified by the question number of the original question from which the variants were generated.

systems' question processing—especially the determination of the expected answer type—could handle different formulations of the same basic question. While the intent had been for each variant to have identical semantics, assessment demonstrated that this was not always the case. Sometimes rewording a question caused the focus of the question to change slightly, so that some answer strings were acceptable for some variants but not others. For example, the assessor accepted "November 29" as a correct response for *What is Dick Clark's birthday?*, but required the year as well for *When was Dick Clark born?*. Similarly, the question *Where is the location of the Orange Bowl?* had many more acceptable responses than did *What city is the Orange Bowl in?*.

Systems that parsed questions into a common representation generally had fewer differences in their responses to question variants than did systems that relied on templates to classify questions by answer types. The Falcon system cached responses to questions and returned exactly the same response for a question that was sufficiently similar to an earlier question.

Figure 2 shows a plot of the average score for each question in a variant set. The average score for a question is the mean of the reciprocal rank scores averaged over the thirty-three runs that used the 50-byte limit on responses and using strict evaluation. The y-axis in the plot is the average score and the x-axis represents the different variant sets. The variant sets are identified by the question number of the original question that was used to generate the variants.

Many variant sets show little variability in the average score. Generally, the average score for each of these variants is low, indicating that the underlying information being sought was difficult to obtain no matter how the question was phrased. A few variant sets did have a wide range of average scores. Frequently the difference was caused by different word choices in the variants. For example, the variant set generated from question 413 asked for the location of the U.S. headquarters of Proctor & Gamble. The variant with the lowest average score was question 725 which used "corporate offices" instead of "headquarters". For the variant set generated from question 440, the original question was *Where was Poe born?*, which had a much higher score than any of the variants that all asked for Poe's birthplace. The unintentional change in focus of some variants also made differences in average scores. "New Jersey" was an acceptable (and common) answer to Question 448, *Where is Rider College located?*, but it was not acceptable for the variant *Rider College is located in what city?*.

4 Question Answering Test Collections

The primary way TREC has been successful in improving document retrieval performance is by creating appropriate test collections for researchers to use when developing their systems. A document retrieval test collection consists of a set of documents, a set of information needs, and a set of relevance judgments that list the documents that are relevant to (i.e. should be retrieved for) each information need. Obtaining an adequate set of relevance judgments for a large collection can be time-consuming and expensive, but once a test collection is created researchers can automatically evaluate the effectiveness of a retrieval run. One of the key goals of the QA track was to build a reusable QA test collection—that is, to devise a means to evaluate a QA run that uses the same document and question sets but was not among the runs judged by the assessors.

Unfortunately, the judgment sets produced by the assessors for the TREC QA track do not constitute a reusable test collection because the unit that is judged is the entire answer string. Different QA runs very seldom return exactly the same answer strings, and it is quite difficult to determine automatically whether the difference between a new string and a judged string is significant with respect to the correctness of the answer. Document retrieval test collections do not have this problem because the unique identifiers assigned to the documents makes it trivial to decide whether or not a document retrieved in a new run has been judged.

The problems caused by not having a reusable test collection are illustrated by the plight of researchers at the University of Ottawa. The pair of runs they submitted to the TREC-8 QA track were misnumbered, and the mistake was not discovered until judging was complete. They were unable to get an official score for their correctly numbered run because they could not map their answer strings to the judged answer strings. As an approximate mapping, they marked any string that completely contained a string that was judged correct in the official adjudicated judgment set as being correct, and all other strings were marked as incorrect (Martin and Lankester 2000). This is a conservative mapping that will almost never mark a string correct that would have been judged incorrect at the expense of marking as incorrect many strings that would have been judged correct. As such it provides a lower bound on the score of the unjudged run, but is sufficiently biased against unjudged runs to make the scores between judged and unjudged runs incomparable.

MITRE also developed an approximate mapping technique based on their work with reading comprehension tests (Breck *et al.* 2000). A human created an answer key for the question set, and strings are then marked based on word recall. If an answer string matches a sufficiently high proportion of the words in an answer key entry, the string is marked correct and otherwise it is marked incorrect. Their analysis demonstrated a high correlation between low word recall scores and judgments of incorrect, and high word recall scores and judgments of correct across the set of runs submitted to the QA track.

NIST created an approximate mapping algorithm similar in spirit to word recall (Voorhees and Tice 2000a). A set of Perl string-matching patterns is created (by a human) from the set of strings that the assessors judged correct. An answer string that matches any pattern for its question is marked correct, and is marked incorrect otherwise. The patterns have been created such that almost all strings that were judged correct would be marked correct, sometimes at the expense of marking as correct strings that were judged incorrect. Patterns are constrained to match at word boundaries and case is ignored.

An average of 1.7 patterns per question was created for the TREC-8 test set, with sixty-five per cent of the questions having a single pattern. The TREC-9 set averaged 3.5 patterns per question with only forty-five per cent of the questions having a single pattern. The increase in the number of patterns per question for the TREC-9 set is another indication that the TREC-9 test set was more difficult.

Using the patterns to evaluate the TREC-8 runs produced differences in the relative scores of different systems that were comparable to the differences caused by using different human assessors. However, unlike the different judgments among assessors, the patterns misjudge broad classes of responses—classes that are usually precisely the cases that are difficult for the original QA systems. For example, an answer string containing multiple entities of the same semantic category as the answer will always be judged correct if the correct answer is mentioned. Document context is also not taken into account. A more sophisticated pattern matching technique could eliminate some of this problem by conditioning patterns by the document ids that are in the judged set, so, for example, questions such as *Who is the prime minister of Japan?* only accept certain names for certain time ranges. But this does not solve the problem for completely wrong context that happens to contain a correct string.

Using patterns to evaluate the TREC-9 runs produced larger differences in the relative scores, especially for 50-byte runs. One way to quantify differences in relative scores is to use a measure of association between system rankings produced when MRR scores are first computed using the human judgments and then computed using the patterns. Kendall's τ is one such measure of association that has been used in earlier evaluations (Voorhees 2000b; Voorhees and Tice 2000a). Kendall's tau computes the distance between two rankings as the minimum number of pairwise adjacent swaps to turn one ranking into the other. The distance is normalized by the number of items being ranked such that two identical rankings produce a correlation of one, the correlation between a ranking and its perfect inverse is negative one, and the expected correlation of two rankings chosen at random is zero. For TREC-8, the τ computed between the rankings using all runs was .96. For TREC-9, the τ computed between the rankings of 250-byte runs was .94, while it was only .89 for 50-byte runs. This smaller correlation is probably the result of a several factors. The TREC-8 human judgment scores were produced using an adjudicated judgment set that was a combination of three different assessors' judgments, and therefore was of particularly high quality. The TREC-9 judgment set is known to contain more errors (i.e., mistaken judgments caused by the assessor hitting the wrong button or similar error) than the TREC-8 judgment set, and these errors could reduce the correlation with the pattern judgments. The more ambiguous questions in the TREC-9 test set were also harder to create patterns for. Nonetheless, the decrease in correlation suggests that concerns about pattern-based judgments are well-founded. Researchers need a fast, reliable method to receive feedback as to the relative quality of alternative question answering techniques. The full benefits of test collection evaluation will not be fully realized for the QA task until more satisfactory techniques for evaluating new runs are devised.

5 The Future

Evaluating competing technologies on a common problem set is a powerful way to improve the state of the art and hasten technology transfer. As the first large-scale evaluation of domain-independent question answering systems, the TREC question answering track brings the benefits of large-scale evaluation to bear on the question answering task.

A roadmap for question answering research was recently developed under the auspices of the DARPA TIDES project (Harabagiu *et al.* 2000). The roadmap describes a highly ambitious program to increase the complexity of the types of questions that can be answered, the diversity of sources from which the answers can be drawn, and the means by which answers are displayed. In particular, the roadmap lists twelve areas in which research is required:

- question taxonomies,
- semantic models of question understanding and processing,
- incorporating user context into answering strategies,
- heterogeneous data sources,
- answer justification,
- answer formulation (display),
- real time question answering,
- multilingual question answering,
- interactive question answering,
- advanced reasoning for question answering,
- user profiles for question answering, and
- collaborative question answering.

The roadmap also includes a five year plan for introducing aspects of these research areas as subtasks of the TREC QA track.

The QA track in TREC 2001 (TREC-10) will include the first steps of the roadmap. The main task in the track will be similar to the task used in TRECs 8 and 9, but there will be no guarantee that an answer is actually contained in the corpus. Recognizing that the answer is not available is challenging, but it is an important ability for operational systems to possess since returning an incorrect answer is usually worse than not returning an answer at all. The track will also contain a subtask in which each question will require information from more than one document to be assembled to produce the answer. For example, a list question such as *Name the countries the Pope visited in 1994.* will require finding multiple documents that describe the Pope's visits and extracting the country from each. The system will also need to detect duplicate reports of the same visit so that countries are listed only once per visit.

The call to participate in TREC is issued each December. Applications to participate are requested by mid-February, but are accepted whenever the call is posted on the main page of the TREC web site (`http://trec.nist.gov`).

Acknowledgements

The TREC QA tracks have been a collaborative effort. The QA participants made the track possible. Amit Singhal and Tomek Strzalkowski were the track coordinators. Paul Over and Dawn Tice helped implement many aspects of the evaluations, and selected the TREC-9 candidate question from the logs. Donna Harman helped define the track evaluation methodology. My thanks to all involved.

References

Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martin, D., Myers, K. and Tyson, M. (1995) SRI International FASTUS system MUC-6 test results and analysis. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 237–48. Morgan Kaufmann.

BBN Systems and Technologies. (1995) BBN: Description of the PLUM system as used for MUC-6. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 55–69. Morgan Kaufmann.

Breck, Eric, Burger, John, Ferro, Lisa, Hirschman, Lynette, House, David, Light, Marc and Mani, Inderjeet. (2000) How to evaluate your question answering system every day ... and still get real work done. *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, **3**:1495–1500.

Burke, Robin D., Hammond, Kristian J., Kulyukin, Vladimir A., Lytinen, Steven L., Tomuro, Noriko and Schoenberg, Scott. (1997) Question answering from frequently-asked question files: Experiences with the FAQ Finder system. Technical Report TR–97–05, Computer Science Department, The University of Chicago.

Cohen, Paul, Schrag, Robert, Jones, Eric, Pease, Adam, Lin, Albert, Starr, Barbara, Gunning, David and Burke, Murray. (1998) The DARPA high-performance knowledge bases project. *AI Magazine*, Winter:25–49.

Fellbaum, Christiane, editor. (1998) *WordNet: An Electronic Lexical Database.* The MIT Press.

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Gîrju, R., Rus, V. and Morarescu, P. (2001) FALCON: Boosting knowledge for answer engines. *Proceedings of the Ninth Text REtrieval Conference (TREC-9).*

Harabagiu, Sanda, Burger, John, Cardie, Claire, Chaudhri, Vinay, Gaizauskas, Robert, Israel, David, Jacquemin, Christian, Lin, Chin-Yew, Maiorano, Steve, Miller, George, Moldovan, Dan, Ogden, Bill, Prager, John, Riloff, Ellen, Singhal, Amit, Shrihari, Rohini, Strzalkowski, Tomek, Voorhees, Ellen and Weishedel, Ralph. (2000) Issues, tasks, and program structures to roadmap research in question & answering (q&a). `http://www-nlpir.nist.gov/projects/duc/roadmapping.html`.

Katz, Boris. (1997) From sentence processing to information access on the world wide web. Paper presented at the AAAI Spring Symposium on Natural Language Processing for the World Wide Web. `http://www.ai.mit.edu/people/boris/webaccess`.

Kupiec, Julian. (1993) MURAX: A robust linguistic approach for question answering using an on-line encyclopedia. *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 181–90.

Kwok, K.L., Grunfeld, L., Dinstl, N. and Chan, M. (2001) TREC-9 cross language, web and question-answering track experiments using PIRCS. *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

Martin, Joel and Lankester, Chris. (2000) Ask Me Tomorrow: The NRC and University of Ottawa question answering system. *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pp. 675–83.

O'Connor, John. (1980) Answer-passage retrieval by text searching. *Journal of the American Society for Information Science*, pp. 227–39.

Schamber, Linda. (1994) Relevance and information behavior. *Annual Review of Information Science and Technology*, 29:3–48.

Singhal, Amit, Abney, Steve, Bacciani, Michiel, Collins, Michael, Hindle, Donald and Pereira, Fernando. (2000) AT&T at TREC-8. *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pp. 317–30.

Voorhees, Ellen M., editor. (2000) Special issue: The sixth Text REtrieval Conference (TREC-6). *Information Processing and Management*, **36**(1).

Voorhees, Ellen M. (2000) Variations in relevance judgments and the measurement of retrieval effectiveness. *Information Processing and Management*, **36**:697–716.

Voorhees, Ellen M. and Tice, Dawn M. (2000) Building a question answering test collection. *Proceedings of the Twenty-Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 200–7.

Voorhees, Ellen M. and Tice, Dawn M. (2000) The TREC-8 question answering track evaluation. *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pp. 83–105.

Webber, B. (1987) Question answering. *Encyclopedia of Artificial Intelligence*, **2**:814–22. Wiley.

Winograd, Terry. (1977) Five lectures on artificial intelligence. In Zampolli, A., editor, *Linguistic Structures Processing*, volume 5 of *Fundamental Studies in Computer Science*, pp. 399–520. North Holland.

Woods, W. A. (1977) Lunar rocks in natural English: Explorations in natural language question answering. In Zampolli, A., editor, *Linguistic Structures Processing*, volume 5 of *Fundamental Studies in Computer Science*, pp. 521–69. North Holland.