TREC9 QA-Track Notebook Paper, NIST, Gaithersburg MD, 2000.

One Search Engine or Two for Question-Answering

John Prager, Eric Brown
IBM T.J. Watson Research Center
Yorktown Heights, N.Y. 10598
jprager/ewb@us.ibm.com

Dragomir R. Radev
University of Michigan
Ann Arbor, MI 48109
radev@umich.edu

Krzysztof Czuba[1]
Carnegie-Mellon University
Pittsburgh, PA 15213
kczuba@cs.cmu.edu

[1] Work performed while at the IBM T.J. Watson Research Center

Abstract

We present here a preliminary analysis of the results of our runs in the Question Answering track of TREC9. We have developed a complete system, including our own indexer and search engine, GuruQA, which provides document result lists that our Answer Selection module processes to identify answer fragments. Some TREC participants use a standard set of result lists provided by AT&T's running of the SMART search engine. We wondered how our results would be affected by using the AT&T result sets. For a variety of reasons we could not replace GuruQA's results with SMART's, but we could use document co-occurrence counts to influence our hit-lists. We submitted two runs to NIST for both the 50- and 250-byte cases, one with and one without consideration of the AT&T document result sets. The AT&T document set was only used for a subset of about a third of the questions. This subset exhibited an increase in Mean Reciprocal Answer Rank score of 13% and 8% for the two tasks.

1. Introduction

Question Answering is a computer-based activity that involves searching large quantities of text and understanding both questions and textual passages to the degree necessary to recommend a text fragment as an answer to a question. The TREC Question-Answering track is an attempt to bring together the Information Retrieval (IR) and Natural Language Processing (NLP) or Information Extraction (IE) communities. The strengths of IR lie in the search engines, while those of NLP lie in the ability to parse and analyze text. Indeed, some NLP groups have no expertise or interest in developing their own search engines, yet still wish to participate in the Question-Answering track. To enable these groups to readily participate, AT&T have made available the results of running their version of the SMART search engine (Buckley, 1985; Salton, 1971) on the questions. These datasets consist of the top 50 documents retrieved for each of the questions in the track. These document sets were used by, amongst others, the best-performing entry in TREC8 QA (Srihari and Li, 2000).

The principal advantage of using these hit lists is that a group can concentrate on the information extraction task of finding the answers from a relatively limited quantity of text. (The entire collection contains close to 1 million documents.) A secondary benefit is that all groups who operate this way are on a common footing regarding the IE activity. The principal disadvantage of this approach, of course, is that no custom question (or collection) pre-processing is possible. As a consequence it can happen that no document in the top 50 available contains the answer to a particular question.

Our group has its own search engine, GuruQA, based on Guru (Brown and Chong, 1998), and is thus able to control the entire processing operation, from question processing and text indexing to answer selection. The technique we use, called Predictive Annotation, involves indexing anticipated semantic types, identifying the semantic type of the answer sought by the question,

and extracting the best matching entity in answer passages (Prager et al. 2000b). Here we explore the question of whether we are at an advantage or disadvantage by not making use of a respected search engine such as SMART as used by AT&T, and we look at a particular way of gaining the best of both worlds.

2. Background

There is much evidence in the field of text processing that combining the results of a single system acting upon different problem formulations, or of different systems acting on the same queries, provides superior performance over individual systems. Belkin et al. (1993) discuss this in some depth for information retrieval. Amongst others, they cite Saracevic and Cantor (1987) and Turtle and Croft (1991) who demonstrated that combining the results from processing with a single search engine, but with different query formulations of a common information need, would produce increased retrieval performance. Foltz and Dumais (1992) used the same query formulation with multiple search engines, and again found increased retrieval performance.

A similar effect has been shown in other areas of the text-processing field, notably classification/part-of-speech tagging (Brill and Wu, 1998)

Given this history, we suspected that we could gain some improvement in the TREC Question-Answering task by combining the hit lists that our search engine produced with those produced by AT&T, and made available to the track participants. The search engine used by AT&T is the SMART system from Cornell; their internally modified version is described by Singhal (1998).

Unlike our system, described in the next section, the AT&T version of SMART used to generate the document sets for Question-Answering was not tailored to the task. It was the same system they used for participation in the TREC7 "Ad-hoc" task. It uses a standard series of IR techniques, such as stop-word removal, tokenization, rule-based and statistics-based phrase formation, *tf*x*idf* style term weighting and relevance feedback using Rocchio weights (Rocchio, 1971). It returned a ranked list of documents with no indication of relevant passages within the document.

3. Our System

Our Question-Answering system employs the technique of Predictive Annotation, introduced and described in (Prager et al. 2000a). The technique revolves around the concept of semantic class labels which we call QA-Tokens, corresponding loosely to some of the Basic Categories of Rosch et al. (1976). These are used not only as Named Entity descriptors, but are actual tokens processed by the indexer. The basic operation of our system is as follows.

The question is analysed and the desired answer type is determined. The "wh-words" are replaced by the corresponding QA-Token or set of QA-Tokens (thus "how hot" is replaced by TEMPERATURE$, "when" is replaced by @syn(DATE$, TIME$, YEAR$)). The QA-Tokens identified in the documents are indexed as if they were regular terms. A set of about 400 patterns is used for this conversion. We found in TREC8 that questions that failed to match this way were usually of the form "What X" where X was a relatively rare noun (e.g. "What debts did the Qintex group leave?"). For these cases we used WordNet (Miller, 1995) to find a hypernym synset of X that corresponded to one of our QA-Tokens. In the case of X= "debts", the synset for "monetary-value" corresponds to MONEY$.

WordNet was also used to generate synonym lists of head nouns in the questions. Word-sense disambiguation was performed by calculating co-occurrence counts, as described by Moldovan and Mihalcea (2000), but using the TREC collection instead of the Web.

Stop-words are removed, inflected terms are reduced to their lemma form, morphological variants that go beyond simple inflection are added as synonyms (thus "moved" -> "move" -> "@syn(move, motion)"). Weights are associated with terms, according to the scheme "QA-Tokens > proper names > common words". This in effect is a simple implementation of *idf* weighting, but applied to the lexical classes of the terms being indexed.

A set of text patterns is associated with each of the approximately 50 QA-Tokens. Before the text collection is indexed, it is processed by Textract (Byrd and Ravin, 1999; Wacholder, Ravin and Choi, 1997) which applies these text patterns; when a match is found the text is annotated with the corresponding QA-Token. The indexer indexes not only the base terms but all annotations too. Thus when the indexer encounters "France", for example, it will also index the tokens PLACE$ and COUNTRY$ at the same location.

Search is not document-oriented but passage-oriented, where a passage is one, two or three sentences. Scoring does not use *tf* but a kind of combination match, where each query term found in the passage contributes its weight to the passage's score, but only once for any number of occurrences. Only one (the "best") passage

is returned per document, thus inducing a document ranking.

The top *n* (normally *n*=10) passages returned by the search engine are then processed by the Answer Selection module, Ansel (Radev et al., 2000). Textract is used again to identify all of the named entities, including simple noun phrases, in those passages. The named entities are typed by QA-Token (simple noun phrases being THING$s), and seven features, such as search-engine ranking, distance from beginning of sentence, and presence of QA-Token in query, are calculated for each entity. A linear evaluation function, using weights discovered by a machine-learning algorithm, is used to associate a final score with each named entity. Finally, text fragments of the desired size (50 or 250 bytes for TREC9) centered on the best named entities are generated.

4. The experiment

The QA track permits participants to submit up to two runs in each of the 50- and 250-byte sub-tracks. Last year we had developed two different answer-selection modules, neither of which was clearly better than the other, so we submitted one run using each module. Since then we have combined the best of each module to give us a single answer-selection component, and we were looking for significant experimental variations we could develop to take advantage of the two-run opportunity. In particular we wanted to do more than just submit runs with different parameter settings. Consequently we decided to submit one run ("R", for Regular) using our system alone, and another ("A", for AT&T) with reference to the AT&T document set.

We will call this latter set SET-A. Our approach was simply to use these documents to increase the score of documents on our own hit lists if the documents also occurred in SET-A (per question).

We had arbitrarily set an internal hit-list size of 10; that is, the search engine would return the top 10 documents (passages) which would then be forwarded to the Answer-Selection module (Ansel). The scores from the search engine were in the range of 0-2000 (approximately). For the "A" run, we increased the internal hit-list size to 50. For every document that was also on the SET-A hit-list we increased its score on our hit-list by 10,000, and then sorted. This had the effect of putting all of the documents that occurred on both hit-lists ahead of those on ours alone, but keeping the relative order within the two groups. The top 10 documents were then forwarded to Ansel. Since search-engine score is a factor in Ansel processing, we subtracted off any 10,000s. This meant that the passage scores that Ansel saw were exactly the scores that our search engine had given. The effect of considering SET-A was therefore solely in determining whether a passage would appear in the input to Ansel. It is an open question, that we need to answer experimentally, whether this is the best way to combine the hit-lists, and whether we should give documents that occur on both lists a permanently increased score.

Note that we did not add to our hit-list any documents that occurred in SET-A but were not originally in our list. This was primarily because our search engine returns not only a document list, but for every document on the list the offset and length of the best-matching passage – for use in Answer Selection. This information was not available in SET-A. The secondary reason was that it was unclear (without any theory or extensive experimentation) how to assign scores to such additional documents.

The overall effect of considering SET-A was to give the "A" run a slightly improved score over the base "R" run. Before we look at the numbers in detail, we need to address the question of whether any improved performance might be due to the AT&T SMART search engine being intrinsically "better" than ours, at least as the two search engines were deployed for this exercise. To that end, we compared the search engines' performances in the following way.

In this comparison we do not ask whether the system extracted the right answer from the documents being considered, but solely whether the documents contained the right answer (a necessary but not sufficient condition for ultimate system success). We call such a document a "correct" document.

We had available lists of which documents out of the 50 per question in SET-A were correct, in the sense just defined. These lists were posted to the QA track mailing list by Ken Litkowski (`ken@clres.com`). For each question there was a list of 0 to 50 numbers in the range of 1 to 50, corresponding to the positions in the hit-list of documents that contained a correct answer. Whether a document contained a correct answer or not was determined by the document's association with a correct response in the `qa-judgments` file, made available on the TREC web site (http://trec.nist.gov) after the TREC9 submissions deadline. Note that this document-judging scheme admits of two possible sources of error: 1) errors by human judges in developing the judgments file, and 2) documents in the set which contained valid answers but were never chosen by any participating entry, so were never judged.

We generated a comparable set of correct-document lists for our own search engine; we'll call this set SET-R. We are now able to compare the two search engines.

Search engine comparison

The first measure we calculated was Mean Reciprocal Document Rank (MRDR) of the first correct response, in analogy to the way the answer fragments are judged in the QA-track. For each question, the Reciprocal Document Rank is 1 point if the top document in the hit-list contains a correct answer, ½ if the first doesn't but the second does, all the way down to 1/50 if only the 50th document does, or 0 if no document on the list does. The RDR scores are averaged over all of the 682 questions. (There were originally 693 questions but 11 were discarded by NIST for reasons of ill-definition.)

For both systems, the MRDR value was calculated to be 0.49 – in other words, on average both systems produced a document containing the right answer in the second position. The fact that both systems had identical MRDR scores indicates that any improvement of our overall score is due to the complementary nature of the two search engines, not to the intrinsic superiority of one or the other.

Before we leave the subject of search engine comparison, we look at two more measures. Choosing the "best" size for a hit list is a heuristic for which we have no firm data. Using a small hit list is desirable if there is frequently a correct document in a high position, since the subsequent processing will then have relatively little noise to contend with, and precision will increase. Long hits lists, on the other hand, offer greater recall. Therefore we looked at subsets of the 50-document hit lists (always starting at document #1). We ask for a hit list of size N (1<=N<=50) whether there was a correct document on the list. The results were very similar, but not identical, for the two document sets, as shown in Figure 1.

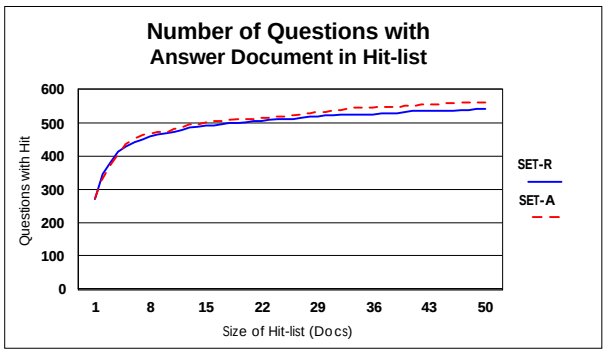

Figure 1. Comparison of number of questions with a "correct" document in the hit list against hit-list depth, for our search engine (SET-R) and AT&T's (SET-A).

The SET-R curve is the solid one, being higher for the first few documents but lower thereafter.

We can make a couple of observations from the data in Figure 1. First, the two search engines seem to have almost equivalent performance. Any significant change in our system's behaviour (good or bad) due to consideration of the AT&T documents will be due to the process of considering multiple result sets, not due to the inherent superiority of one or other set. Second, the curves suggest absolute values for hit list size. To the degree to which the answer-selection processes that operate on these document sets are imperfect, that is, suffer in precision due to the presence of incorrect documents in the set, the hit-lists should be cut back. An obvious "knee" in both curves occurs at around 6-10 documents. On the other hand, if the answer processing is sufficiently sophisticated to be able to easily reject incorrect documents based on their internal semantics, then the hit lists could be quite large. It would be useful then to know at what point the curves, if extended to the right, would asymptote to 100% (=682 questions for TREC9). The only data point we have in this regard is for GuruQA. 542 questions had at least one correct document on a hit-list of size 50; with hit-lists of size 200 we have a correct document for 576 questions. This suggests that the asymptote is far away, and that the correct place to concentrate effort on answering these residual questions is in question pre-processing, and possibly collection processing prior to indexing.

We also looked briefly at the overlap between the document sets. For the entire 50-deep hit lists, we asked which ones contained a correct document somewhere. The totals across 682 documents are presented in Table 1.

# of Questions with correct documents	SET-R Yes	SET-R No
SET-A Yes	483	80
SET-A No	59	60

Table 1. This shows how the two search engines overlapped in answering a question. A document set scores a "Yes" for a question if at least one of the documents in the set contains a strictly correct answer to the question.

It is difficult to make too many quantitative predictions about the potential advantages of using document result sets from multiple systems, since there is more processing to come after the result sets are established.

Actual Performance Increase

We did not attempt to use the AT&T documents for every question, since we did not have a chance to test the idea (using the previous year's sets) before the current year's submission. Instead, we just used them on those question types for which we previously had experienced inferior performance. These were questions, generally of the form "What X …", for which none of our QA-tokens was instantiated (save for THING$, matching a generic noun phrase, which was created just for such situations).

Of the 682 questions, there were 214 questions which we labelled type THING$. We calculated the Mean Reciprocal Answer Rank (MRAR) score for the THING$, non-THING$ and total sets of questions, both with ("A") and without ("R") the AT&T documents. The MRAR scores reported here are for the actual answers, not the correct documents as MRDR measures in the previous section.

The results are summarized for the 50-byte run in Table 2, and for the 250-byte run in Table 3. Two styles of judging were provided in TREC9: *strict*, in which the answer was present in the returned fragment and justified in the surrounding context, and *lenient*, where the correct answer was present but not necessarily in a context that addressed the question. All of the data reported here were for the *strict* interpretation.

.

50 byte task	THING$	Non-THING$	Overall
"R"	.151	.390	.315
"A"	.171	.372	.309

Table 2. MRAR scores calculated for THING$, Non-THING$ and all questions, for runs with and without consideration of SET-A documents, for the 50 byte sub-track.

250 byte task	THING$	Non-THING$	Overall
"R"	.335	.454	.416
"A"	.363	.454	.425

Table 3. MRAR scores calculated for THING$, Non-THING$ and all questions, for runs with and without consideration of SET-A documents, for the 250 byte sub-track.

It was expected that there would be a difference between the runs for THING$-type questions, but it can be seen that the Non-THING$ scores also differ between the "A" and "R" runs, in the 50 byte task. This occurred for two reasons, which curiously only affected the Non-THING$ questions. Firstly, there was a word-alignment error in the 50-byte fragment selection code that was present in the "A" system but not in the "R" system. This caused in some cases critical answer words to be truncated and hence disallowed. This affected 6 questions (actually, 2 questions plus 4 paraphrases of one of them), to the tune of a loss of .009 to the MRAR score. The remaining .009 of the discrepancy was due to inconsistent judging (the same answer submitted by both runs was judged differently). Eight questions were negatively affected by these judging problems: in seven cases the "A" run was affected, and in one the "R" run. Unfortunately, the deleterious effects of the inconsistent judging and our alignment bug swamped the positive effect of using the second document set, when the overall scores are calculated (for the 50-byte runs).

From Tables 2 and 3, we see that for the 214 THING$ questions, in the 50-byte sub-track MRAR improved by 13%, and in the 250-byte sub-track by 8%. An experiment that we need to do now is to try a run using the SET-A documents for the Non-THING$ questions too. Due to the labor-intensive nature of the document judging, we will await a set of answer patterns per question from NIST to enable us to judge such future runs automatically.

5. Conclusions and Future Work

Indexing QA-Tokens improves the precision of our IR system, since it gives it more semantics and provides a means of better matching questions to answers. The technique is not so useful in conditions when no semantic type is identifiable, such as "What X" type ques-

tions. The experiments reported here demonstrate that considering results sets from a second search engine can improve QA results, at least for those "What X" questions. This was achieved using search engines working under very different operational conditions and with a very basic method of combining the hit-lists. These results extend existing demonstrations of the benefit of using multiple systems in "ad-hoc"-style search and classification.

An incidental discovery was that our use of GuruQA with Predictive Annotation and passage ranking produced result sets with identical MRDR to AT&T's version of SMART, for 214 "What X" type questions. This finding may be related to the existence of theoretical and practical limits to the results achievable with statistical information retrieval.

As mentioned earlier, we plan to see what kind of improvement will be afforded if the approach is extended to all question types. It is also completely open what is the best way to incorporate the information from other result sets. We took the simplest possible approach, which was to move documents that occurred in both hit lists up towards the top of ours. We did not experiment with increasing the score, which we expect will positively effect the results, since Answer-Selection uses passage score as a feature.

References

[1] N.J. Belkin, C. Cool, W.B. Croft and J.P. Callan. "The Effect of Multiple Query Representations on Information Retrieval System Performance", *Proceedings of SIGIR'93*, Pittsburgh, PA, 1993.

[2] E. Brill and J. Wu.. "Classifier combination for improved lexical disambiguation", in *Proceedings of COLING-ACL*, 1998.

[3] E.W. Brown and H.A. Chong. "The Guru System in TREC-6." *Proceedings of TREC6*, Gaithersburg, MD, 1998.

[4] C. Buckley. "Implementation of the SMART information retrieval system." Technical Report TR85-686, Department of Computer Science, Cornell University, Ithaca, NY 14853, May 1985.

[5] R. Byrd and Y. Ravin. "Identifying and Extracting Relations in Text", *Proceedings of NLDB 99*, Klagenfurt, Austria, 1999.

[6] P.W. Foltz and S.T. Dumais. "Personalized information delivery: An analysis of information-filtering methods." *Communications of the ACM*, 35, 12: 51-60, 1992.

[7] G. Miller. "WordNet: A Lexical Database for English", *Communications of the ACM* 38(11) pp 39-41, 1995

[8] D.I. Moldovan and R. Mihalcea. "Using WordNet and Lexical Operators to Improve Internet Searches", *IEEE Internet Computing*, pp. 34-43, Jan-Feb 2000.

[9] J.M. Prager, D.R. Radev, E.W. Brown and A.R. Coden. "The Use of Predictive Annotation for Question-Answering in TREC8", *Proceedings of TREC8*, Gaithersburg, MD., 2000.

[10] J.M. Prager, E.W. Brown, A.R. Coden and D.R. Radev. "Question-Answering by Predictive Annotation", *Proceedings of SIGIR 2000*, pp. 184-191, Athens, Greece, 2000.

[11] D.R. Radev, J.M. Prager and V. Samn. "Ranking Suspected Answers to Natural Language Questions using Predictive Annotation", *Proceedings of ANLP'00*, Seattle, WA, 2000.

[12] J.J. Rocchio. "Relevance feedback in information retrieval" in *The SMART Retrieval System – Experiments in Automatic Document Retrieval*, pp. 313-323, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.

[13] E. Rosch et al. "Basic Objects in Natural Categories", *Cognitive Psychology* 8, 382-439, 1976.

[14] G. Salton (ed). *The SMART Retrieval System – Experiments in Automatic Document Retrieval*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971.

[15] T. Saracevic and P. Kantor, "A study of information seeking and retrieving. III. Searchers, searches, overlap." *Journal of the ASIS*, 39,3: 197-216, 1988.

[16] A. Singhal, J. Choi, D. Hindle, D.D. Lewis, F. Pereira. "AT&T at TREC7", *Proceedings of TREC7*, Gaithersburg, Md., 1999.

[17] R. Srihari and W. Li. "Question Answering Supported by Information Extraction", *Proceedings of TREC8*, Gaithersburg, Md., 2000.

[18] H. Turtle and W.B. Croft. "Evaluation of an inference network-based retrieval model." *ACM Transactions on Information Systems*, 9,3: 187-222, 1991.

[19] N. Wacholder, Y. Ravin and M. Choi. "Disambiguation of Proper Names in Text", *Proceedings of ANLP'97*. Washington, DC, April 1997.