Question Answering Experiments and Resources

Boris Katz, Gregory Marton, Sue Felshin, Daniel Loreto, Ben Lu, Federico Mora, Özlem Uzuner, Michael McGraw-Herdeg, Natalie Cheung, Yuan Luo, Alexey Radul, Yuan Shen, Gabriel Zaccak
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139

1 Introduction

MIT CSAIL's entries for the TREC Question Answering track (Voorhees, 2006) explored the effects of new document retrieval and duplicate removal strategies for 'list' and 'other' questions, established a baseline for other systems in the interactive task, and focused on question analysis and paraphrasing, rather than incorporation of external knowledge, in the factoid task. Many of the individual subsystems are largely unchanged from last year.

We found that document retrieval strategy has an influence on performance in the different kinds of tasks later in the pipeline. Our other changes from last year did not immediately yield clear lessons. We present a question analysis data set and interannotator agreement indicators for the ciQA task that we hope will spur further evaluation.

1.1 Factoid Question Answering

Our emphasis for factoid answering this year was on identifying answers within the collection, rather than finding answers on the Web and projecting them onto the collection.

For several years, systems have thrived on the latter strategy. The Aranea system (Lin and Katz, 2003), developed here four years ago, uses Web search engines (Google, Yahoo, Ask) to find many short snippets that appear with the question, and uses redundancy and heuristics about answer types to rank answers, which it then reranks based on occurrences in the source text. The major advantage is that the Web may have the answer phrased in a manner that is easier for text processing systems to identify. However, the best answer on the Web may not be the best answer in the collection of interest, and thus using this as the only or dominant strategy may be a weakness.

In renewing our focus on the target collection, we addressed issues of document retrieval, question analysis, and paraphrase-based answer extraction.

In document retrieval, we experimented with external relevance feedback using Google and Wikipedia search results, and with examining only high-confidence documents.

We used our START system for question analysis, and made improvements to its handling of sentential topics, anaphora, and answer type identification. START also produces an assertion, in which the question is transformed into a statement with a variable marker and its type at the position of the linguistic trace.

We developed a new system, SmartQA (Loreto, 2006), to select answers from text based on how well they match START's assertion. SmartQA uses the Stanford parser (Klein and Manning, 2003) to parse both the assertion and each candidate sentence, and a set of structural transformations and scoring heuristics that yield a final score for each sentence. We reranked these answers based on redundancy information from the Web and answer type heuristics, by passing them through Aranea as a last stage of processing.

1.2 List Question Answering

Our baseline list question answering strategy remained the same as in previous years: we look for phrases matching the expected an-

swer type, in contexts surrounding hot spots of question keywords or their synonyms. This year we added minor enhancements in the answer type analysis and duplicate removal mechanisms, and submitted runs with and without these enhancements.

We experimented with viewing list and factoid questions in analogous ways, processing the list question into an assertion using START as we do for factoid questions, and taking SmartQA's top k answers as the answer to the list question. In the end, however, we submitted only one run containing SmartQA results, and these simply added the top answer from SmartQA if it was not yet present in the otherwise normally generated list.

1.3 Other Question Answering

An important part of the 'Other' task, even emphasized by its name, is that it should not repeat information, either from the previously asked questions, or within its own list of answers. Our 'novelty' component last year in this task did as much harm as good—removing many redundant answers, but also removing answers that would have been marked as containing a nugget. This year we compared various other strategies for duplicate removal, and submitted the best one based on our previous tests.

Our 'other' system simply reranks all the sentences in its input based on how well they match the topic phrase, and on how little they overlap with previously given information. Thus, unlike in the factoid task, it is not necessarily a good retrieval strategy to aim for very high recall whatever the precision. We tested a document retrieval method that placed more emphasis on precision, and found an improvement.

1.4 Complex Interactive Question Answering

For the complex interactive question answering task, we attempted to provide a baseline for others based on last year's results. The ciQA organizers had suggested during the organizing meeting at TREC2005 that our system, which had the best performance on the Relationship task last year, would serve as a good baseline for evaluation this year. We therefore ran the same system this year, modified only in that it was tuned for the best performance on last year's data set.

For the interactive component, the organizers had suggested that one simple form of feedback might be simply asking the assessors which responses from the non-interactive engine had good answers, and later returning all and only those answers. This is almost the strategy we followed, except that, for the purpose of having ground truth for as many responses as possible, we also filled in previously below-cutoff responses, up to the character limit.

1.5 Results

Our results indicate that the different ways of using retrieved documents were sensitive to the characteristics of the retrieved set. Factoid question answering, which is relatively rich in information about the expected answer, benefitted from higher recall in the document retrieval component, though only slightly, whereas 'other' question answering, which simply looks for representative sentences, benefitted from the more focused set of documents in the strict retrieval condition. (Figure 1)

Our unchanged complex interactive question answering runs showed similar performance as last year, but many systems now exceeded this performance, showing an overall improvement in the state of the art over last year. (Figure 2)

2 Document Retrieval

Underlying each component of our question answering system is keyword-based document retrieval using Lucene[1]. Last year, we found that various keyword backoff mechanisms did not retrieve documents with higher recall than the default Lucene baseline. This year we experimented with obtaining relevance feedback

[1] `http://lucene.apache.org/`

Run	Factoid	List	Defn
csail01	default docs: 0.154	list06a: .122	default+novelty: .106, .116
csail02	(same as csail01)	list06b: .125	strict+novelty: .124, .140
csail03	strict docs: 0.149	list06+SQ_1: .120	strict+edit-dist. clus.: .117, .142

Figure 1: Main task results: document retrieval and average accuracy for each factoid run; method and average accuracy for each list run; document retrieval+choice of duplicate removal method, average F score and average pyramid score for each definition run.

Run	Description	Score	Nuggets	Responses	Length
csail1	reln2005 system output	.203	140	1116	174673
csailif1	selected by-response	.209	136	1102	173989
csailif2	selected by-word	.203	132	1127	173837

Figure 2: ciQA results: the result of our optimized relationship 2005 system on this new task, along with the results after returning those responses that assessors marked correct in interaction, plus previously below-cutoff responses.

from Wikipedia and Google results, as well as a "strict" method for returning a more focused set of documents.

For query expansion, we used Wikipedia synonymy and Web-based relevance feedback. For each topic-question pair p, we found the top ten Google snippets, and for each topic we found the first paragraph. To each of the words in these texts, we assigned a relevance $r(w)$, based on the frequency of that word in the corpus $c(w)$, of the topic and question within the corpus $c(p)$, and their intersection $c(w,p)$, such that[2] $r(w) =$

$$1 - \frac{\max(\log(c(w)), \log(c(p))) - \log(c(w,p))}{\log(|A|) - \min(\log(c(w)), \log(c(p)))}$$

Given this "similarity to the topic and question", we chose a cutoff, above which we added the resulting words to the query.

We also used Wikipedia's redirect structure to find "synonyms" for words and phrases in the topic and question. For example, if the question contains "TWA800", and we know that "TWA Flight 800" redirects to the same Wikipedia page, then we use that as an expansion.

The strict condition sought to impose a minimum relevancy cutoff on the document retrieval task. To do so, we restricted document retrieval to the subset of documents containing all keyword tokens in the topic phrase. To improve recall while maintaining this strict relevancy, we used the Wikipedia synonyms for each keyword as a bag-of-words expansion. We then ranked the resulting document subset according to the Lucene scores on the keywords in question, which were further expanded to include the terms from the Web-based context feedback process.

Because we used both expansion as described above and this strict cutoff for the alternate runs, the two are confounded. However, because we used the topic and the question in the expansion, we found few cases where expansion made a difference, so we attribute most of the change to the strictness parameter.

3 Question Analysis

We used START's question analysis module to turn questions into assertions for further processing. These assertions can be more easily matched against sentences of evidence than can the questions themselves. The Stanford parser, trained primarily on statements, was used in the following stage to obtain parses of both the assertion and the candidate sentences, to help score each candidate.

[2] $|A|$ is the size of the AQUAINT corpus.

	TREC 2005		TREC 2006	
	all	complete	all	complete
correct:	276	138	237	117
recall:	.608	.304	.508	.264

Figure 3: Question analysis performance: START finds the correct assertion in about six of ten cases, and correctly resolves all references in the question in about half of those.

To evaluate the question analysis component, we also annotated the correct assertions for all questions in the TREC 2004 and TREC 2005 data sets, and for most of the TREC15 data set. Annotation was performed by one person unfamiliar with the details of the START system, and in cases where it did not match START's output, was adjudicated with two others, until agreement was reached. At the time of the TREC 2006 submission, START generated an assertion matching our ground truth for 61% of the annotated TREC2005 cases. On its "test set", the TREC2006 data, START correctly generated 51% of the ground truth assertions. (See Figure 3.)

The annotation involved two kinds of assertions, one "almost correct" which is a close restatement of the original question, and one "complete" in which references to the topic and to any previous questions or answers are resolved.

START handles question-to-question and question-to-topic coreference using information about gender, animacy, proper/common distinction, number, discourse salience, and coreferentiality of partial names, and of synonyms or hypernyms.

START attempts to identify the focus of each question and provide it as the answer type. We have generated ground truth for these focus answer types, but have not yet vetted them as well as we have the assertions.

4 'Other' Questions

The 'other' question task asks us for relevant information not yet presented, and CSAIL's approach is, as in previous years, to look for the most representative sentences from the relevant collection of sentences that have the least overlap with information already presented. We have captured the notion of 'least overlap' in a keyword-based "novelty" algorithm (Katz et al., 2005).

Experiments with the novelty algorithm showed that it helped as much as it hurt: while it removed many sentences that did not contain nuggets, it also removed enough sentences that did to offset the score gain from shorter responses. We implemented alternate strategies based on edit distance and on Bleu (Papineni et al., 2001) overlap score, and tested the results on the TREC 2005 'other' task. The results are presented in Figure 4.

Type	Vital	Okay	avg. F-measure
Lucene only			
none	146	153	0.1409 ± 0.0298
edit-distance	144	153	0.1612 ± 0.0331
novelty	79	120	0.1447 ± 0.0390
bleu	142	145	0.1426 ± 0.0343
bleu + stop	143	145	0.1431 ± 0.0344
DB + Lucene			
none	165	180	0.0928 ± 0.0263
edit-distance	164	180	0.1122 ± 0.0304
novelty	78	101	0.1107 ± 0.0304
bleu	143	144	0.1084 ± 0.0308

Figure 4: A comparison of algorithms for removing duplicates in the 'other' task. *Edit-distance* uses clustering with an edit distance metric to group and remove duplicate nuggets. Our Bleu-based method uses the popular machine translation metric to detect similar pairs of sentences (with or without prior stopword removal) and removes the less novel of the pair. Novelty finds the sentence that maximizes weighted keyword overlap with information not yet presented and minimizes weighted keyword overlap with information already presented. The DB condition includes sentences that were previously identified as definitional contexts of the topic.

Because the edit distance clustering ap-

proach showed the best results on this data set, we chose to devote one run to seeing if the edit distance-based duplicate removal would do better than the novelty algorithm. It did, but only very slightly, as predicted by the preliminary results. (See Figure 1.)

A much greater difference came from the use of the strict document retrieval—returning fewer documents, the most relevant ones. This reflects the fact that, unlike in factoid question answering, we have very little extra information to use in selecting good answers—we select the most representative answers. The representative answers from a smaller, more focused collection can then be expected to contain more nuggets.

5 Complex Interactive Questions

In the complex interactive task, we submitted a run using our unmodified relationship 2005 system, with its best parameter settings for that task. Unlike the University of Maryland baseline, which only used the template fillers for sentence retrieval, we attempted to use the narrative, as we had in the relationship task. Our results thus represent a baseline of sentence selection after narrative question analysis.

We used two types of interactive forms. Both are based on the best answers from our non-interactive system. One type of form presents all of those answers and asks the assessor to check off those answers that they consider "good"—relevant, correct, etc. The second type of form offers a finer-grained interface: it again presents every answer, but instead of checkboxes for entire answers, it allows the assessor to click on individual words in the responses to designate those words as either being "good" themselves or belonging to a "good" phrase.

On receiving the results from both of these kinds of forms, we simply submitted, as the beginning of each of our interactive submissions, just those responses that either were marked "good," or contained any words or phrases that were marked "good," respectively. Since this necessarily decreases the overall sizes of our responses to each question, we filled the responses out to the limit of 7000 characters by adding additional responses that our non-interactive system had ranked below the responses already seen by assessors.

By submitting much the same responses for assessment three separate times, we are able to look closely at the agreement between judgements, as well as the agreement between the assessor clicking "good" items under time pressure, vs. doing so in the formal assessment environment. These results are summarized in Figure 5. Low kappa scores for nuggets vs. clicks occur because most clicks do not correspond to a nugget assignment. The kappa scores for assignment of nuggets, between 0.75 and 0.86, can be interpreted as inter-annotator agreement for this task, and thus as guidance for the likely significance of comparisons between systems.

6 Contributions

Our TREC entry this year showed a proof-of-concept factoid system in which the question is converted to an assertion, and the assertion then parsed and matched against parsed candidate answer sentences, to find candidate answers. We have created a dataset of correct assertion versions of the TREC questions for the past several years, which may be useful for other systems attempting to parse questions or to learn correct reference resolution. We have begun to create a data set of question focuses for each question, which may help others with expected answer type identification.

We tested the effects of different priorities in document retrieval as applied to the factoid and 'other' tasks, finding that there may be an interaction.

We established a baseline for ciQA performance, and gave indications of the extent of inter-annotator agreement on ciQA judging.

We look forward to working with other participants interested in using these resources for their own evaluations.

	orig vs. sentences	**orig vs. words**	**sentences vs. words**	**orig vs. agreeing subset of sentences and words**
κ **for nuggets**	0.75	0.80	0.79	0.86
Responses in Common	631	682	912	561
κ **for clicks**	0.07	0.07	0.65	0.08

Figure 5: ciQA evaluation consistency: for each comparison, the κ for agreement between nuggets assigned, the number of responses in common over which we evaluated that κ, and finally the κ for clicks, either between clicks and nuggets assigned in the cases of comparison between the original submission and the indicated form responses ($\kappa < 0.1$), or for the comparison between the two forms ($\kappa = 0.65$).
In the run descriptions, **orig** indicates *csail1*, the initial output of our relationship system on the question set, **sentences** indicates *csailif1*, the responses to forms where entire sentences could be selected, **words** indicates *csailif2*, the responses to forms where individual words could be selected.
The difference between the comparison of forms directly (third column) and the comparison of their agreeing subset with the original (fourth column), reflects the contribution of the previously unseen responses. On average, 11.7 unseen responses made it into each final response set, constituting a third of the length, but they yielded only ten new nuggets consistently assigned overall, or one tenth of the assigned nuggets.

References

Boris Katz, Gregory Marton, Gary Borchardt, Alexis Brownell, Sue Felshin, Daniel Loreto, Jesse Louis-Rosenberg, Ben Lu, Federico Mora, Stephan Stiller, Ozlem Uzuner, and Angela Wilcox. 2005. External knowledge sources for question answering. In *Proceedings of the 14th Annual Text REtrieval Conference (TREC2005)*, November.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics (ACL2003)*.

Jimmy Lin and Boris Katz. 2003. Question answering from the web using knowledge annotation and knowledge mining techniques. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM 2003)*, November.

Daniel Loreto. 2006. Exploiting syntactic relations for question answering. Master's thesis, MIT Infolab, Cambridge, MA, September.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL2002)*, July.

Ellen Voorhees. 2006. Overview of the TREC 2006 question answering track.