CSAIL at TREC 2007 Question Answering

Boris Katz, Sue Felshin, Gregory Marton, Federico Mora, Yuan K. Shen, Gabriel Zaccak, Ammar Ammar, Eric Eisner, Asli Turgut, L. Brown Westrick MIT Computer Science and Artificial Intelligence Laboratory

Cambridge, MA 02139

1 Introduction

MIT CSAIL's entries for the TREC 2007 question answering track built on our systems of previous years, updating them for the new corpora. Our greatest efforts went into the system that handles the 'other' questions, looking for new descriptive information about the topic. We noticed in our experiments with Nuggeteer (Marton and Radul, 2006)¹ that some of the parameters made a big difference in the results, and decided to restructure our scoring to be able to tune its parameters. This represents the first such use of the Nuggeteer software that we are aware of, and yielded excellent results.

2 Approach and Results

We discuss our approaches and results for each system component, from preliminary steps including document retrieval and question analysis, to the final systems that answered each kind of question.

2.1 Question Analysis

We used the START natural language question answering system (Katz, 1988; Katz, 1997) for question analysis. For each question, START identifies a question focus or answer type, and transforms the question into an assertion. In our evaluations last year, these had accuracies near 52% and 63% respectively, with correct resolution of all referent phrases occurring in 32% of questions. Our system this year added only bugfixes.

We notice that some of the question sets this year used entirely or almost entirely explicit references to the topic. When the exact list of such questions is published, we look forward to comparing our overall performance on the resolved and unresolved subsets.

Question analysis affects list questions primarily by specifying the expected answer type. The list processor then decides whether each candidate string is of that type, as described in Section 2.4. For factoid questions, the second output of question analysis, the question transformed into assertion, is used to match the environments that candidate answers appear in. Question analysis can affect 'other' questions by identifying the portions of the topic as being synonyms (as in the case of abbreviations) or else by separating nominal parts from actions for events.

2.2 Document Retrieval

We used Lucene² for indexing and retrieval on the AQUAINT2 corpus, and BLOG06 corpus, though we chose not to incorporate results from the BLOG corpus in most of our systems. Unlike in previous years, question analysis did not affect document retrieval, because we did not try any query expansion.

¹http://csail.mit.edu/~gremio/code/Nuggeteer

²lucene.apache.org

2.3 Factoid Question Answering

Our factoid question answering component, Aranea, remained much the same system as has been used in previous years. The minimum changes made, reflected in our csail1 entry, included adapting specialized extractors to recent changes in the web site layouts from which they gathered data, and updating the IDFs to reflect the new AQUAINT2 corpus. We did not use the BLOG corpus for factoid answering.

We had also noticed that we would have benefitted in previous years from disregarding one of the Aranea components, and we ran this ablated system as csail2. For this data set, the component turned out to help by 2% (p < 0.05).

In the third entry, we made the csail2 scoring modification, and we also experimented with an aggressive filtering mechanism based on answer type. We got six of the 16 NIL answers correct, compared to zero in the other runs. While this raised our score, the final score was not different from either the immediate csail2 baseline or the cross-year csail1 baseline (both p > 0.2).

2.4 List Question Answering

Our baseline list question answering strategy was the same as in previous years: we look for phrases which match the expected answer type, and are near question keywords or their synonyms (Katz et al., 2006). The csail2 list system was the same system as last year's, with new question analysis and bugfixes. For our csail1 entry this year, we added Wikipedia categories³ to the set of answer types with lists of known members.

The csail1 entry, using Wikipedia, performed better than the baseline by 1.4%, but not statistically significantly. It influenced 18 questions, and certainly enabled us to answer some questions we would otherwise not have been able to answer, such as about names of ships. In a pairwise ttest on those 18 questions, however, the score improvement remained shy of statistical significance (0.050 .

We did not use the BLOG corpus in answering list questions, but excluding BLOG data did not significantly affect our score.

2.5 "Other" Question Answering

Our system for answering "Other" questions, seeking new information about the topic of the question series, closely follows previous year's systems, but adds major infrastructural improvements to response ranking, and redefines and improves our ability to project descriptions from trusted sources onto the collection of interest.

The system has a candidate generation and a candidate ranking phase. Candidate generation aims to include all relevant nuggets among the available response snippets, and candidate ranking is responsible for selecting the most salient snippets and for minimizing redundancy.

2.5.1 Candidate Generation

In candidate generation, response snippets can come from two sources: direct keyword search and pre-extracted definitional contexts. Documents retrieved through direct keyword search are segmented into paragraphs and sentences, and both paragraphs and sentences become candidate snippets.

Pre-extracted definitions were generated using our ColForbin (Fernandes, 2004) system, which uses manually defined surface syntactic patterns to extract definitional text snippets from documents annotated with BBN IdentiFinder named entity tags (Bikel et al., 1999). Examples of extraction patterns include: *copula*

³http://en.wikipedia.org/wiki/Category

that describe is-relationships: "[Named-Entity: Ur] is [Definition: the home of capital of Sumeria]"; *appositives* that describe additional facts about the target entity: "[Named-Entity: Archimedes], [Definition: a philosopher and a mathematician], was celebrated in Ancient Greece". We identified approximately 1.25 million definitional snippets from the AQUAINT2 corpus, and did not run ColForbin on the BLOG06 corpus.

2.5.2 Candidate Ranking: Projection

Projection is the process of finding an answer in one source and then locating support for it in another. In the case of 'other' questions, the intuition is that others have put effort into compiling the most 'vital' nuggets of information about many topics in sources like Wikipedia and Google Timelines⁴, so we would like to find the same nuggets in our corpus of interest. We looked for similarity between the trusted sources and the candidate snippets using the BLEU similarity metric (Papineni et al., 2001).

For Wikipedia, we used as "trusted text" the contents of the top Wikipedia page in a Google search on the topic (as opposed to just the first paragraph, as in previous years). For Google Timelines, we used the text from the first page of responses.

2.5.3 Candidate Ranking: Redundancy

For redundancy elimination, we have experimented with a word-based editdistance metric to prune near-identical responses, and with complete-link hierarchical agglomerative clustering (both described in (Katz et al., 2006)). These were incorporated as the novelty feature, available to the final scorer.

2.5.4 Candidate Ranking: Parameters

Our ranking system uses machine learning to train parameters for the five features described in Table 1. The first four features, topic match, informativeness, source, and novelty, are carried over from the old system, whereas the projection feature is new—projection had been applied by modifying the keyword weights to be used in other parts of scoring. The features are real-valued functions over the query and response snippet, and the score function combined these features into a boolean relevance judgement.

We trained scoring function weights on data from previous years using the Nuggeteer program to assign likely nuggets to each of our generated responses. Nuggeteer memorizes all human-judged responses exactly, and uses a keyword-based similarity metric with the known answers to estimate nugget judgements for previouslyunseen responses. Unlike other systems, it provides a likelihood for the presence of each nugget in each proposed response. We used TREC2006 as the training set, but did not use a validation set.

We used the Weka toolkit (Witten and Frank, 2005) to create classifiers in several families, including support vector machines, logistic regression, radial basis function, and decision trees. We used the output of a logistic function over the binary classifiers to obtain a score between zero and one, which we then used to rank the responses. We simultaneously also computed the score using the method in our TREC2006 system, and used that score to break ties.

 $^{^{4}}$ Google Timelines is a recent Google Labs' product that returns a set of chronological facts for a given topic. See http://www.google.com/views? q=thomas+jefferson%20view%3Atimeline

Feature	Description		
Feature F_{topic}	DescriptionThe topic should be mentioned in each response, but may not match exactly. F_{topic} approximates the overlap between the actual topic (or synonyms) and words in the response.If we consider the set of unique terms in the topic Q (or a synonym), and the terms in the best named entity and key- word matches in the response R , and the exactly matching set between them M , then F_{topic} is an F-measure: $F(p, r, 2)$, of the topic precision and recall $p = \frac{\sum_{w \in M} t(w)}{\sum_{w \in B} t(w)}$ $r = \frac{\sum_{w \in M} t(w)}{\sum_{w \in Q} t(w)}$		
F _{inform}	Here $t(w)$ is the term weighing function, in this case IDF. As an example, if the topic were "Warren Moon retires" and the candidate: "Fred Moon is 52.", then F_{topic} would be close to F(0.5,0.33,2) (modulo term weights). The third value is the beta, weighting recall as more important than precision. In a future version, we will try separating the precision and recall portions of this into two features. IDF captures the observation that words that occur less frequently in the corpus are generally more informative. F_{inform} approximates the "informativeness" of a response by comparing their combined idf score with the corpus av- erage idf, idf_{avg} . For a response R ,		
	$F_{inform} = \frac{\sum_{w \in R} idf(w)}{idf_{avg} \cdot w \in R }$		
F_{source}	While we have only two main response sources, the re- sponses from pre-extracted definition source can vary in quality depending on the pattern type. F_{source} is an enu- meration of the possible sources, 15 sources are from the definition extraction source <i>e.g.</i> def:appositive (apposi- tive pattern), and 2 are from the lucene source (whether it was a paragraph or sentence.)		
$F_{novelty}$	If an answer is similar to another one already given, then it is unlikely to yield previously unseen nuggets. Our approaches to novelty scoring are described in (Katz et al., 2006)		
$F_{projection}$	This feature quantifies the ngram distributional similarity between a response d and a set of sentences from a trusted source about the topic $T(q)$. The trusted source could be the Wikipedia article for the given topic, or the Google Time- lines result. The similarity metric we used was BLEU (Pa-		

pineni et al., 2001).

2.5.5 Results

Our submitted responses to 'Other' questions consisted of two successful runs: csail2 included projection (on both Wikipedia and Google Timelines) but not the new scoring, and csail3 included projection and a ranking function trained using logistic regression. Our csail1 submission, meant as a baseline without either the new projection or the new scoring, unfortunately had errors that made it unusable for comparison.

In addition to the two successfully submitted runs, we also conducted several internal experiments on previous years' data. In our experiments, we varied: 1. projection sources: Google Timelines (gtl) or Wikipedia (wiki) or both; 2. sources of candidate responses: search (luc) or preextracted definitions (db) or both. 3. methods for tuning the score function. The baseline runs consisted of best settings for TREC 2006. Evaluation was done using Nuggeteer (Marton and Radul, 2006). Table 2 summarizes our experiments, and includes the F-measures for previous years and pyramid scores for the current year.

The projection feature alone improved performance significantly. Experiments on previous years' data showed a remarkable gain of 3-5% in F-measure (and a gain of 7-10% in recall) as compared to our baseline. Experiments varying the source of projection showed that using Wikipedia alone resulted in a better performance (17.0%) versus using Google Timelines alone (14.7%).

In some examples, we noticed that Google Timelines mix references to various items with the same name, whereas the top Wikipedia page is either on topic, or badly off topic, so the projected data will be consistent.

Experiments in varying the algorithm for optimizing the ranking function showed that simple methods such as logistic regression gave significant improvements, a 2% Fmeasure gain for 2004, and a 3.7% improvement in this year's submissions (csail2 vs. csail3). The more sophisticated methods did worse, but this may be due to the decision tree's overfitting. For example decision trees are known to often overfit the training data, and indeed, examination of the generated decision trees (for TREC 2006) pointed to an overly complex (multi-level) rule, that clearly indicated overfitting as a problem.

Lastly, experiments comparing our two response sources, pre-extracted definitions (db) and Lucene search engine results (luc), showed that the Lucene source was predominant in returning valid responses. Lucene-only results provided the system with much needed answer recall. The combination of the two sources (db+luc) perform slightly better than Lucene-alone but not by any large margin. While preextracted definitions were of higher quality, its low recall, greatly diminished its usefulness as a source of descriptive responses. This confirms our analyses from last year (Katz et al., 2006).

3 Contributions

Our TREC-2007 baselines for factoid and list made minimal modifications from last year, but one cannot immediately conclude that the questions were harder, because we did not use the BLOG06 corpus. In closer analysis for list questions, excluding answers that were listed with BLOG supporting documents, our systems gained about 0.6% across the board, indicating that at least the newspaper part of the list question answering is more difficult than last year, where our scores with the same system were some 6% higher.

A common thread was that using Wikipedia in both list and 'other' questions seemed to improve our performance, and

TREC 2007: 70 topics							
	Pyramid score						
csail2 gtl+wiki	0.198						
csail3 logistic gtl+wiki	0.235						
	$F(\beta=3)$	Precision	Recall				
TREC 2006: 75 topics (Vorhees, 2006)							
gtl+wiki ¹	$0.1718 \hspace{0.1 in} \pm 0.0404$	0.0457 ± 0.0081	0.3342 ± 0.0790				
wiki ¹	$0.1702 \ \pm 0.0387$	$0.0450\ {\pm 0.0081}$	$0.3360 \ \pm 0.0761$				
gtl^1	$0.1474 \ \pm 0.0371$	$0.0405 \ \pm 0.0080$	$0.3007 \ \pm 0.0754$				
$db+luc^2$	$0.1308 \ \pm 0.0351$	$0.0380 \ \pm 0.0076$	$0.2586 \ \pm 0.0712$				
luc $only^2$	$0.1225 \ \pm 0.0347$	$0.0382 \ \pm 0.0075$	$0.2351 \ \pm 0.0655$				
$db only^2$	$0.0625 \ \pm 0.0256$	0.0243 ± 0.0105	$0.1336 \ \pm 0.0553$				
baseline	$0.1233 \ \pm 0.0386$	$0.0415 \ \pm 0.0094$	0.2382 ± 0.0749				
TREC 2005: 75 topics (Voorhees and Dang, 2005)							
logistic ³	$0.2445 \hspace{0.1 cm} \pm 0.0408$	$0.0692 \ \pm 0.0126$	0.4131 ± 0.0738				
$linear^3$	0.2253 ± 0.0391	0.0589 ± 0.0116	0.3984 ± 0.0763				
svm^3	$0.2052 \ \pm 0.0395$	$0.0556 \ \pm 0.0119$	$0.3539 \ {\pm 0.0712}$				
$rbfnetwork^3$	0.2048 ± 0.0399	$0.0580 \ {\pm 0.0111}$	$0.3643 \ \pm 0.0725$				
$j48^3$	$0.2016 \ \pm 0.0357$	0.0535 ± 0.0111	0.3751 ± 0.0742				
$\mathrm{gtl}\mathrm{+wiki}^\dagger$	$0.2282 \ \pm 0.0383$	0.0573 ± 0.0113	0.4111 ± 0.0758				
baseline	$0.1906 \ \pm 0.0345$	$0.0509 \ \pm 0.0102$	0.3463 ± 0.0689				
TREC 2004: 65 topics (Voorhees, 2004)							
logistic ³	$0.3522 \ \pm 0.0564$	$0.0920 \ \pm 0.0193$	$0.5905 \ \pm 0.0886$				
$linear^3$	$0.3432 \ \pm 0.0560$	$0.0869 \ {\pm 0.0187}$	$0.5840 \ \pm 0.0914$				
$rbfnetwork^3$	$0.3240 \ {\pm 0.0544}$	$0.0836 \ \pm 0.0187$	$0.5501 \ \pm 0.0894$				
svm^3	$0.3218 \ \pm 0.0566$	$0.0815 \ \pm 0.0190$	$0.5475 \ \pm 0.0911$				
$j48^{3}$	0.3138 ± 0.0564	0.0836 ± 0.0191	0.5327 ± 0.0914				
$gtl+wiki^{\dagger}$	$0.3388 \ \pm 0.0557$	0.0850 ± 0.0186	0.5840 ± 0.0914				
baseline	$0.2830 \hspace{0.1 in} \pm 0.0503$	$0.0760\ {\pm 0.0152}$	$0.4826 \ \pm 0.0881$				

Table 2: Summary of TREC 2007 results for 'other' questions and experimental results on TREC 2004-2006 data. Reported are f-measure, precision, recall of responses from Nuggeteer under differing system settings. **baseline** used only settings from previous year.

¹ Runs that varied projection sources, where gtl is using Google Timelines and wiki is using Wikipedia.

 2 Runs that varied candidate generation strategies, where db is using the ColForbin database and luc is using Lucene.

³ Runs that varied ranking methods. Training data was derived from TREC 2006 questions. Experiments used gtl and wiki as projection sources, and both db and luc as candidate generation methods. Classifiers were trained using default settings from Weka (Witten and Frank, 2005).

 † Used both db and luc for candidate generation.

	Factoid	Lists	'Other'	Per-Series
csail1	baseline	wiki	baseline	
	0.131	0.068	(0.076)**	0.093
csail2	scoring variant	no-wiki	projection	
	0.111	0.054	0.198	0.122
csail3	scoring variant and	wiki [†]	projection and logis-	
	filtering		tic regression	
	0.119	0.067	0.235	0.142
Best	0.706	0.479	0.329	0.484
Median	0.131	0.085	0.118	0.108

Table 3: A summary of our TREC 2007 results, as compared with Best and Median values for all participants.

† the second wiki run used a different question analysis input.

** the first "other" run used failed document retrieval, so the score should not be considered as a true baseline.

learning to make the best use of trusted outside sources is a future priority.

Our greatest effort and best results this year were in the 'other' questions, where a more systematic approach to parameter estimation and better use of projection from trusted descriptive sources raised our performance significantly in a traditionally difficult task.

Our effort on 'other' questions was fueled by our success with using Nuggeteer as a tool for system performance estimation and tuning. Our experiments using Nuggeteer correlated with the eventual human judgements, validating the tool and the approach.

References

- Daniel M. Bikel, Richard L. Schwartz, and Ralph M. Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231.
- Aaron D. Fernandes. 2004. Answering definitional questions before they are asked. Master's thesis, Massachussetts Institute of Technology.
- Boris Katz, Gregory Marton, Sue Felshin, Daniel Loreto, Ben Lu, Federico Mora, Ozlem Uzuner, Michael McGraw-Herdeg, Natalie Cheung, Yuan Luo, Alexey Radul, Yuan Shen, and

Gabriel Zaccak. 2006. Question answering experiments and resources. In *Text Retrieval Conference (TREC)*.

- Boris Katz. 1988. Using English for indexing and retrieving. In Proceedings of the 1st RIAO Conference on User-Oriented Content-Based Text and Image Handling (RIAO 1988).
- Boris Katz. 1997. Annotating the World Wide Web using natural language. In Proceedings of the Conference on the Computer-Assisted Searching on the Internet, (RIAO 1997).
- Gregory Marton and Alexey Radul. 2006. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *HLT-NAACL*.
- 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 M. Voorhees and Hoa Trang Dang. 2005. Overview of the trec 2005 question answering track. In *Text Retrieval Conference*.
- Ellen M. Voorhees. 2004. Overview of the trec 2004 question answering track. In *Text Retrieval Conference*.
- Ellen Vorhees. 2006. Trec 2006 question answering overview. In *Text Retrieval Conference* (*TREC*).

Ian H. Witten and Eibe Frank. 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition edition.