External Knowledge Sources for Question Answering

Boris Katz, Gregory Marton, Gary Borchardt, Alexis Brownell, Sue Felshin, Daniel Loreto, Jesse Louis-Rosenberg, Ben Lu, Federico Mora, Stephan Stiller, Özlem Uzuner, Angela Wilcox
MIT Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139

1 Introduction

MIT CSAIL's entries for the TREC Question Answering track (Voorhees, 2005) focused on incorporating external general-knowledge sources into the question answering process. We also explored the effect of document retrieval on factoid question answering, in cooperation with a community focus on document retrieval. For the new relationship task, we present a new passage-retrieval based algorithm emphasizing synonymy, which performed best among automatic systems this year.

Our most prominent new external knowledge source is the Wikipedia[1], and its most useful component is the synonymy implicit in its subtitles and redirect link structure. Wikipedia is also a large new source of hypernym information.

The main task included factoid questions, for which we modified the freely available Web-based Aranea question answering engine; list questions, for which we used hypernym hierarchies to constrain candidate answers; and definitional 'other' questions, for which we combined candidate snippets generated by several previous definition systems using a new novelty-based reranking method inspired by (Allan et al., 2003).

Our factoid engine, Aranea[2] (Lin and Katz, 2003), uses the World Wide Web to find candidate answers to the given question, and then projects its best candidates onto the newspaper corpus, choosing the one best supported. Candidate generation uses snippets from Google and Teoma, and this year from Yahoo and from the newspaper corpus as well. This year, we added an Aranea module that, in the spirit of our approach to list questions, boosts candidate answers which are hyponyms of the question focus. Finally, we rank answers using a combination of Web-based and corpus-based prominence rather than using only the top Web-based answer.

Our list engine (Tellex et al., 2003) retrieves passage-sized chunks of text relevant to the question using information retrieval techniques, and projects onto them the fixed lists associated with the question focus. This year, we augmented our knowledge base with lists extracted from Wikipedia, and attempted to use a relevant Wikipedia article to initially search for answers, paralleling Aranea's approach. As in previous years, question analysis was performed primarily by START[3] (Katz, 1988; Katz, 1997; Katz et al., 2002). This year, in anticipation of new event topics, we expanded START's question analysis for TREC to include relative clauses and some additional constructions.

Our definition engine combines snippet rankings from several independent components. We identify syntactic structures associated with definitional context *a priori*, then match topics against the resulting database of target–nugget pairs (Hildebrandt et al., 2004; Fernandes, 2004). We also rank snippets from IR using a *tf*idf*-based score which heavily favors matching targets. Finally, we use a keyword-based novelty score to select the best

[1] http://en.wikipedia.org/

[2] http://www.umiacs.umd.edu/~jimmylin/projects/aranea.html

[3] http://start.csail.mit.edu/

answers. This year, we incorporated external knowledge by adding Wikipedia-based synonymy, and testing two methods of selecting snippets based on matches with a Wikipedia article.

Our relationship engine scores snippets from Lucene using precision- and recall-like measures: recall is based on how many of the synonym groups in the question were covered, and their relative importance, while precision is based on a heuristic semantic distance from each synonym group to the words in a candidate passage that were used to fill it. Prior context outside the snippet is permitted to augment recall. We generated results based on manual and heuristic-based automatic question analysis.

We will describe each of the systems in more detail below, and expand on official results to explore component contributions.

2 Document Retrieval

Underlying each component of our question answering system is keyword-based document retrieval using Lucene[4]. We explored three modifications to the default (baseline) query behavior: idf weighting, idf-based backoff, and idf-based backoff treating the most important ("anchor") phrases in the question as undroppable. These strategies are described in greater detail below, and summarized in Figure 1. Our list and definition systems use a single Wikipedia article, which we selected via a combination of Lucene and Google queries.

2.1 Baselines

Each year NIST distributes the top 1000 document results of its PRISE document retrieval system for each question along with the questions, so that participating teams need not do their own document retrieval. A second baseline was the default Lucene behavior, as specified by a disjuction of the query terms.

2.2 Idf Backoff

Our first experimental document retrieval strategy uses successive conjunctive queries, gathering up to 1000 hits by successively dropping the lowest idf term from the query. For each term, the final query specifies all inflectional variants in a disjunction (Bilotti et al., 2004). This strategy was used in the majority of our experimental results.

2.3 Anchored Backoff

A new experimental strategy made use of "anchor" words or phrases identified by START in the question. These anchor terms may be named entities or known collocations. Term disjunctions in this strategy include derivational as well as inflectional term variants, and for multiword terms they allow a fixed distance between the terms. The overall strategy is still to drop least important terms first, but now non-anchor words are successively dropped and re-added for each anchor word that is dropped. The minimal query then is the term disjunction for the single most important anchor term. In a variant of this strategy, we can look for anchors using the list knowledge base (Section 5.1).

2.4 Idf Weighting

The relationship engine issues a single disjunctive query for each topic, but weights each term by the portion of the entire query's idf that it and its synonyms are responsible for. In the case where no two terms are synonymous, this is just each term's own idf. Where two words are synonymous, each term's weight will be the sum of both terms' idfs. We treat other variation, such as morphology, as we do synonyms. Unlike the other two strategies, we do not expand each query term to be a disjunction of that term's inflectional variants.

2.5 Wikipedia

The first step in employing Wikipedia for a question is finding the relevant article for a topic. Topics in previous years were restricted to simple noun phrases, and over 90% of them appeared in Wikipedia in some form. Of the 75 topics this year, 10 were "event" noun phrases like "1998 Nagano Olympics" and 4 were headline-like events, like "Liberty Bell 7

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

Default:	(A+B+C+D)
Idf weighting:	(A^a+B^b+C^c+D^d)
Idf backoff:	(A*B*C*D); (A*B*C); (A*B); (A)
Anchored bck:	(A*NE*C*D); (A*NE*C); (A*NE); (NE)

Figure 1: Document Query Strategies: small "a" is the idf of query term capital "A", and $a > b > c > d$. Semicolons indicate a subsequent query, executed if not enough documents have been retrieved. *NE* refers to a named entity or some other anchor term (previously undistinguished as term B) which is deemed to have special significance to the query.

space capsule recovered from ocean". Fewer of these appear in Wikipedia. We found the correct Wikipedia article for 87% of the noun phrase topics, for 70% of the noun phrase event topics, and for none of the headline topics—81% accuracy overall.

We found relevant articles in Wikipedia by varying capitalization and noun number, looking for topic words in the body as well as the title of an article, and as a last resort doing a Google search restricted to Wikipedia's main namespace. We resorted to Google for 39 topics, among them all 14 of the incorrect articles mentioned above. Some Wikipedia articles were not about, but only briefly mentioned, the topic of interest; if matching content was low, then the matching paragraphs were used as if they formed an article.

2.6 Document Retrieval Results

Our document retrieval strategies did not yield different results in the top 50 candidates that matter for factoid question answering. Some systems like our definition engine may use more than this number of documents, and our anchor-based model yielded a significantly higher recall over all 1000 documents than our baseline. We showed that strategies using Lucene and inflectional variation offer above-median performance on this task. (See Figure 2)

3 Wikipedia Synonymy

Many system components made use of synonymy information extracted from Wikipedia. This information is implicit in subtitles and the redirects between pages: "TWA800" and "Trans World Airlines flight 800" redirect to "TWA flight 800", and "Woodrow Wilson Guthrie" redirects to "Woody Guthrie". This sort of synonymy is relatively ad-hoc, and unpredictable in the sense that humans unfamiliar with the particular domains of these synonymy facts would also have trouble or uncertainty in deciding whether the pairs were synonyms. In these cases, the encyclopedic knowledge of synonymy can be instrumental.

4 Factoid Questions

We have been using the Aranea system for question answering for four years, and it has recently become open source. We submitted one run based on the latest open-source Aranea ("Aranea04", adapted minimally to our topic-based infrastructure), and two runs using an improved Aranea ("Aranea05"). We used the two improved runs to test the difference in end-to-end performance between the *baseline* and *anchor* document retrieval strategies described above. (See results in Figure 3.)

Attention to Topic: Aranea04 treats each question separately, relying on question analysis to substitute the topic before processing begins. Aranea05 uses the format introduced at TREC13, and performs a preliminary search on the topic alone. Subsequent Web queries use the concatenation of the topic and question, and scores of candidates that were prominent in the topic alone are damped.

Attention to Focus: Like list questions, factoid questions often include a noun-phrase question focus. For "In what sea did the submarine sink?", the focus would be "sea". We used WordNet to find hyponyms of this focus, where available, and boosted scores of candidate answers that matched such hyponyms, so that "Barents Sea" would become a more likely candidate than "icy Barents".

Document Retrieval Method vs. Factoid Performance

Method	R-prec.	R@10	R@50	Avg. Prec.	Factoid_{48}	Factoid_{362}	MRR_{prj}
Oracle	1	1	1	1	.354	n/a	.281
Best in TREC2005	$.705_{\pm.06}$	$.846_{\pm.06}$	?	$.706_{\pm.07}$	?	.713	?
Default Lucene	$\mathbf{.345}_{\pm.07}$	$\mathbf{.498}_{\pm.10}$	$.576_{\pm.09}$	$\mathbf{.342}_{\pm.07}$	.271	.234	.188
Idf-weighted	$.311_{\pm.08}$	$.414_{\pm.10}$	$.515_{\pm.10}$	$.308_{\pm.08}$	.271	.222	.188
PRISE	$.304_{\pm.07}$	$.432_{\pm.10}$	$\mathbf{.578}_{\pm.09}$	$.285_{\pm.07}$	.292	.234	**.255**
Anchored backoff *	$.270_{\pm.07}$	$.462_{\pm.10}$	$.548_{\pm.10}$	$.282_{\pm.07}$	**.313**	.251 §	.147
Idf-backoff *	$.258_{\pm.07}$	$.456_{\pm.10}$	$.561_{\pm.10}$	$.289_{\pm.07}$	**.313**	**.256** §	.140
Median TREC2005	$.167_{\pm.04}$	$.308_{\pm.08}$	?	$.157_{\pm.04}$	?	.152	?

* Documents evaluated in rank order instead of score order due to nondescending scores in backoff runs.

§ Recent web results caused differences from official .273 anchored and .260 idf-backoff Factoid_{362} scores.

Figure 2: **Document retrieval results**: While our experimental document retrieval strategies performed below baseline, they nevertheless yielded above-baseline performance in final question answering. We evaluate recall at 50 because our factoid engine, Aranea, used the top 50 documents for projection, and indeed the best recall at 50 (PRISE) also showed the best factoid projection MRR. We show document and factoid evaluation on the 48 factoid questions out of 50 total questions for which document retrieval was directly evaluated. We also show corresponding factoid results for all 362 factoid questions, though these cannot be directly compared with document retrieval results. The MRR_{prj} column indicates the mean reciprocal rank of the correct factoid answer over the 48 relevant questions in the Aranea projection step, which finds candidate answers in the top 50 Lucene documents and ranks these by quality of match—thus this is the component that directly translates document retrieval performance to an effect on factoid answering performance. Factoid performance here reflects the Aranea05 system. For our anchored and idf backoff experiments, recall at 10 or 50 is out of an average of $31.5_{\pm 17}$ supporting documents per question. Missing values (?) could not be measured without detailed results for best and median runs. Factoid_{362} results cannot be measured for oracular document retrieval because human assessments were not performed.

run (doc+factoid)	R @ 50	correct / nonzero	all questions
csail1 (idf bck+A04)	$.561_{\pm.10}$	9 / 41	$.207_{\pm.04}$
csail2 (anchors+A05)	$.548_{\pm.10}$	13 / 42	$.273_{\pm.05}$
csail3 (idf bck+A05)	$.561_{\pm.10}$	13 / 41	$.260_{\pm.05}$

Figure 3: **Factoid question answering results:** Our three conditions are shown. Ranking did not have an effect for the 48 factoid questions where document rankings were evaluated, but it did have an effect overall. The difference in answering performance on the restricted question set is marginally significant ($p = .052$). The difference in answering performance on the entire question set is significant for Aranea05 over Aranea04 ($p < 0.01$), but not between the two document retrieval strategies for Aranea05 ($p = .113$).

New Modules	Score	MRR_{web}
-Topic, -Focus	.248	.278
-Topic	.248	.278
-Focus	.254	.270
Full system	.251	.270

Figure 4: Factoid score and MRR of web-based candidates, ablating the new Topic and Focus modules from Aranea05.

Newspaper candidates: Aranea04 searches only the Web for candidate answers, but we felt that the corpus might occasionally have relevant information. Thus as we added new Web search engines like Yahoo, we also added the top Lucene results from the newspaper corpus, just as if they had come from a Web search engine.

Integrated answer projection: Once a set of candidate answers is found on the Web, Aranea04 selects the top answer and finds a best match in the newspaper corpus. On the assumption again that the newspapers might be a good source of information, Aranea05 instead scores the top 10 Web-based answers and combines Web-based and newspaper-based scores using F-measure to select its best answer.

4.1 Factoid Results

We observed a significant improvement in question answering performance due to changes to Aranea. We did not observe a great difference in results due to document retrieval (see Figure 3), but this is unsurprising, because the differences between our document retrieval strategies themselves were not significant.

In analyzing individual components, we found that attention to topic may have been helpful, while attention to the question focus may have been detrimental, but neither difference was large (see Figure 4). Combining web-based and newspaper-based rankings, however showed a clear win (see Figure 5).

Projection Reranking	Score	MRR
Web ranking	.202	$.270_{web}$
Project top 10	.199	$.287_{prj}$
Combined	.251	.329

Figure 5: Factoid score and MRR of the web rankings alone, the projected rankings alone, and the ranking combining these with F-measure.

5 List Questions

Our list question architecture identifies an answer type for each question, retrieves newspaper articles, identifies contexts with query terms, and selects phrases matching known answer types from those contexts. We reorder these candidates using frequency and return the top 50.

5.1 Answer Types

The START Natural Language Question Answering System identifies the *focus* of a list question, a noun phrase most descriptive of the expected answer type. Three of START's internal functions were exposed in a TREC-specific API, and enhanced to work with a wider array of questions:

- Noun-phrase parsing for the topic itself,
- anaphoric substitution to place the topic into each question as appropriate, and
- focus extraction to find for each question the type of answer sought.

In anticipation of the more complex topics this year, we improved START's handling of relative clauses and other complex noun phrase constructions, and its handling of anaphoric hypernyms of the antecedent. We incorrectly assumed that no questions would refer to noun phrases from previous questions, or to the answers to those questions. [5] START also identifies possible *focus* phrases for each

[5]Several questions did, in fact refer to the previous answer, including 68.5, 68.7, 71.4, 81.3, 84.2, 84.3, 120.4, 120.6, 136.3, 136.5, 137.3, and also possibly 67.2, 67.3, 70.6, 81.4, 84.5, 114.6, 131.6, and 137.6.

list question and offers several reformulations for further analysis. For example the reformulations for "Name famous people who have been Rhodes scholars" included:

- "famous people who have been Rhodes scholars"
- "famous people"
- "Rhodes scholars"
- "people"

We hypothesized that a larger number of lists and a wider variation of list names would improve both coverage and specificity: coverage by matching more question focuses, and specificity by matching more specific reformulations that had smaller associated lists.

We expanded our answer type knowledge base using Wikipedia. Lists in Wikipedia fall into three categories also observed in other kinds of corpora: A list might be the entire purpose and content of an article, it might make up part of a larger description, or the article might mention that it is about a particular topic, in which case the set of articles on that topic is a list. In Wikipedia we found 48,412 full-article lists, 166,263 lists within larger articles, and more than a million category mentions. In comparison to the 3000 lists we used last year, and to the 150 the year before, these represent a very large potential increase in coverage. Wikipedia thus provides what is to our knowledge the largest source of manually generated list information.

We used only full-article lists because they afforded the most straight-forward means to associate the list with a descriptive phrase: their article title. We generated alternate names for each list by heuristically removing modifiers, and by using Wikipedia subtitles and link structure (see Section 3).

For one experimental condition we used our lists from last year (csail1) where for the other conditions (csail2 and csail3) we added the full-article Wikipedia lists.

For each possible question focus, and each possible reformulation of that focus from START, we select all matching list names, and treat the union of matching list members as the answer type, the set of possible answers to the question.

5.2 Candidate Generation

Candidates are generated from an answer type (a set of possible known answers), and a text, by looking for instances of the answer type within the text. For many answer types we also generated "guess answer" candidates using a named entity recognizer. Candidate scores were assigned based on quality of the reformulation used and the proximity of question keywords to the candidate list item.

In csail1 and csail2 we used newspaper documents as the text for selecting list answers (using baseline and anchors document retrieval respectively), while in csail3 we used a Wikipedia article. After finding answers in a Wikipedia article, we used Aranea's new projection module to find those answers again in the newspaper corpus, that being our final target corpus.

5.3 Answer Selection

Answer selection is based on the scores assigned during candidate generation, the number of times a candidate is proposed, heuristic filters and a top-k cutoff. List answers were always used in preference to guess answers.

5.4 List Results

Adding Wikipedia lists to our knowledge base improved the recall upper bound for 2003 (32% to 39%) and 2004 (33% to 39%), but did not do so for 2005 (32% for both plain and Wikipedia). The changes in the underlying document collection used account for about 1% of the precision changes shown.

Figure 6 summarizes our list results.

Figure 7 summarizes the differences in list results for each system due to the various answer type components. The difference in lists-2004 result between csail1 and the other two runs is due to a bug: we removed knowledge about people, so all questions requiring a person answer type fell back on the other two answer type sources, usually the IdentiFinder "guess" answer source (Bikel et al., 1999). It

is difficult to separate the influence of this bug from the influence of adding the Wikipedia hyponyms, so we are reluctant to draw conclusions from these data.

6 Definition Questions

We call the final "Other" question in each topic a "definition" question, seeking nuggets of interesting information about the topic that were not addressed in the previous questions.

Our baseline sought candidate nuggets in three ways: by looking up the topic in a pre-compiled database of definitional contexts (Hildebrandt et al., 2004; Fernandes, 2004), by searching the corpus for a short context that includes many keywords from a Webster's Dictionary definition for the topic, and by simply positing each sentence from the top retrieved documents. This strategy used Wikipedia synonyms of the target for matching, if available.

In two experimental conditions, we also generated candidate nuggets by looking for newspaper sentences that had a high overlap with the first paragraph of the best Wikipedia article according to the BLEU metric (Papineni et al., 2001). Our three experimental conditions were: without Wikipedia/BLEU (csail1), with Wikipedia/BLEU (csail2), and with Wikipedia/BLEU on newspaper articles where anaphora had been resolved.

We ranked candidates first by topic accuracy, and then by *tf*idf* of non-topic terms, where *tf* is frequency within the set of candidates. Topic accuracy is an F-measure based on word-based precision and recall of the topic; "Clinton" in a candidate where the topic is "Hillary Clinton" would have 100% precision and 50% recall. Wikipedia synonyms received 100% F-measure.

We subsequently removed candidates that were too similar, again based on keywords, to nuggets that had been selected at lower rank. The algorithm is described and evaluated in detail in (Marton et al., 2006). This similarity score takes *idf* into account, but focuses on how many of the keywords are new vs. old. We did not use Wikipedia synonymy in this similarity computation, though it might have been a good idea.

We used various heuristic score-based cutoffs, and submitted at most 24 sentences for each topic.

6.1 Definition Results

We observed no significant differences between our systems in their ability to find nuggets for definitional or "Other" questions, nor in any particular strategy of our system, but we do note that there is a significant difference in response length favoring our new BLEU-based strategy (Figure 8). Though length is the deciding factor, we would have done better to include more results.

7 Relationship Questions

Our approach to the relationship track is based on passage-retrieval methods. Banking on mutual disambiguation among the question terms, we extend standard passage retrieval scoring with a precision- and recall-like approach to synonymy. We used as one source of synonyms a thesaurus we developed for last year's pilot, inspired by the "spheres of influence" in the task description. Unlike other passage ranking systems, we also incorporate an effect of prior context. As with definition questions, we incorporate a model of novelty to iteratively select the best and most novel passage at each rank.

7.1 Question Analysis

Training data were available from a pilot conducted last year. Many question keywords are extraneous or misleading, most notably "The analyst ...", but also the subsequent "... is interested in ..." and more. Our question analysis aims primarily to eliminate these non-contributory phrases, using regular expressions developed on questions from 2004.

Some phrases are salient only together, e.g., "United Nations" should be penalized if the words are separated. We marked those phrases that appeared in our synonymy knowledge bases.

List Results				
run	zero	precision	recall	F($\beta = 1$)
csail1: lists-2004	47	.1066±.038	.1773±.048	.1102±.033
csail2: +WikiKB	43	.0841±.033	.1575±.045	.0883±.026
csail3: +WikiSearch	43	.0829±.025	.1820±.045	.1004±.027

Figure 6: List questions: We answered all 93 questions, but gave no correct answer for nearly half. Using the expanded knowledge base was detrimental. The 43 zero-questions in csail2 and csail3 represent different sets of questions. F-measure difference between csail1 and csail2 is marginally significant ($p = .026$). The precision difference between csail1 and csail3 is significant ($p = .014$).

List Breakdown by Contributor												
run	total			wiki			lists 2004			guess		
	all	sel.	precision	all	sel.	precision	all	sel.	precision	all	sel.	precision
csail1	354	271	.121±.04				220	161	.118±.05	193	158	.110±.05
csail2	319	247	.101±.04	37	21	.047±.04	84	39	.045±.04	231	205	.084±.03
csail3	231	277	.105±.03	64	45	.057±.04	115	70	.071±.04	241	223	.102±.03

Figure 7: Credit assignment for correct list answers. *All* shows correct answers from the entire set of candidates, while *sel.* and *precision* include only those submitted, after answer selection. Total precisions are high because document support is not considered. Sums of rows are higher than total because some answers were selected by multiple sources.

We would have liked to exclude some background information phrases, e.g., "Osama Bin Laden" from the first question, "We know Osama Bin Laden is in charge, but what other organizations or people are involved...". We can do so with manual question analysis, but were unable to exclude them in our automatic run.

It was possible to mark a word or phrase as "important", but the automatic question analysis did not attempt to do so.

7.2 Candidate Ranking

Sentence ranking is based on precision- and recall-like measures. Each question term is assigned a weight based on its *idf*. Words that are synonymous according to our lexicons are pooled and their weights summed. The weights of words in the final sentence, and of some other useful terms, are boosted.

We retrieve 500 newspaper articles using Lucene on a disjunction of all terms with the weights described above. Synonymous terms from the question are included in the Lucene query as well, each with the pooled weight. We note each document's Lucene *DocScore*.

Candidate recall reflects how many terms from the query had matching terms in the candidate, or its prior context, as a portion of the sum of *idf*s in the query. Each matching term or synonym contributes its matching query term's full *idf* to recall, however poor the match.

Candidate precision reflects how well the terms in the candidate matched the terms in the query. Exact term matches have a similarity of 1, and other similarity values come from each source of synonyms. Some, for example in the manually built thesaurus, are manually assigned. If more than one variant of a query term appears in the candidate, then the variants reinforce each other, so that the combined similarity for those terms is one minus the product of their *dis*similarities. The precision score for the candidate is the average similarity of all matching terms.

The final candidate score combines candidate precision, candidate recall, and a docu-

Definition Official Results						
run (id)	nuggets	items	#returned	char/resp	zero	$F(\beta = 3)$
csail1 (91): syns	154	144	1554	142	34	0.1557±0.0389
csail2 (96): +bleu	145	132	1571	118	38	0.1606±0.0471
csail3 (101): +anphr	150	139	1591	118	39	0.1602±0.0479

Figure 8: Definition official results: none of the F-measure differences are significant ($p > .40$). *Nuggets* is the number of nuggets assigned. *Items* is the number of responses with at least one nugget. *#Returned* is the number of responses returned. *Char/resp* is the number of non-whitespace characters per response, on average. *Zero* is the number of qids with no correct nuggets. The length of csail1 nuggets is significantly greater than the lengths of the other two ($p < .0001$); however, the difference between csail2 and csail3 is non-significant ($p = .20$).

Definition Corrected				
run	nuggets	items	zero	$F(\beta = 3)$
csail1	156	148	33	0.1593 ± 0.0391
csail2	154	152	37	0.1689 ± 0.0485
csail3	159	155	37	0.1745 ± 0.0495

Figure 9: Corrected results: if a string was assigned a nugget in *any* run, that nugget was assigned automatically here. This only applies to complete string matches. Differences are still non-significant, and this result emphasizes the variability of judgements, reversing the difference between csail2 and csail3.

Definition Breakdown by Contributor					
run	bleu	lucene	database	webster	total
csail1		103/1050	50/495	1/9	1554
csail2	111/1190	22/ 253	11/124	1/4	1571
csail3	111/1201	28/ 262	11/124	0/4	1591

Figure 10: Credit assignment for definitional nuggets: The *bleu* strategy is useful, and appears to be orthogonal to whether the newspaper sentence is a "definitional context". *Bleu* shows contributions of using the BLEU similarity metric to choose newspaper sentences that were most similar to sentences from the first paragraph of the best Wikipedia article. The *lucene* strategy simply selected corpus sentences that matched the target. *Database* is our collection of definitional contexts. *Webster* shows the contribution of selecting newspaper sentences by their similarity with a dictionary definition. The identical numbers of *bleu* matches is coincidence—the answers judged correct are not the same. The answers from the database are the same for csail2 and csail3.

ment score using F-measures:

$$F_{\beta=3}(F_{\beta=2}(precision, recall), DocScore)$$

Given the ranked list of candidate sentences, we then use our keyword-based novelty filtering algorithm to select those to display. Of those, we submitted the top 24 candidates for each question.

7.3 Synonymy

A central component of the algorithm above is the synonym knowledge base[6]. We used synonyms from Wikipedia as described in Section 3, nominalizations from Nomlex (Macleod et al., 1998), and the small thesaurus we developed for the pilot last year. We also treated case and morphology as synonymy.

Our thesaurus has a two-level structure: a *synset* level has a high precision and contains closely related terms, while a *topic* level relates financial terms, exchange of goods terms, terms related to crime, and nine other topics.

In our experience, candidate selection is more sensitive to the relative precisions of the different kinds of synonyms than to the absolute weights. Our manually created thesaurus was grouped into topics inspired by the "spheres of influence" (SOI) from the task definition.

7.4 Results

The relationship task was comparable to the definition task in difficulty, with top partially-manual and fully-automatic systems performing at 28% and 23% respectively.

Our two experimental conditions were whether the process above was performed manually or automatically. Due to a single-character human error we submitted our same automatic run for evaluation twice. The slight difference in our results is actually due to annotator error.

That automatic run performed well (see Figure 11). We have used Nuggeteer (Marton, 2006), a Pourpre-like automatic analysis tool, and Pourpre itself (Lin and Demner-Fushman,

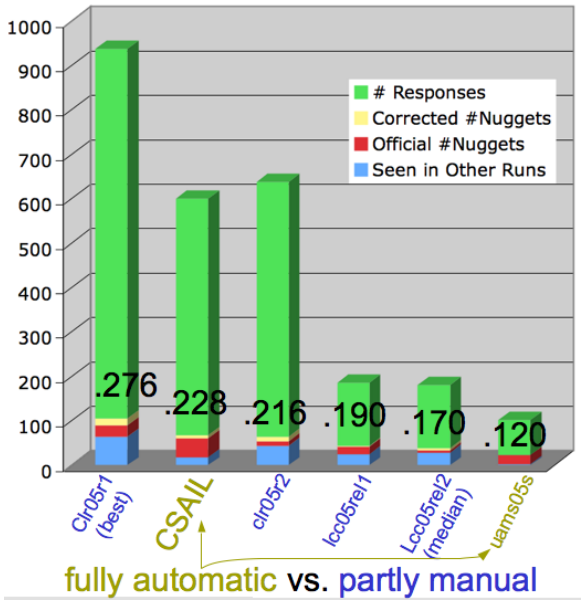

Figure 11: Relationship system performance: official scores are shown for the top six systems. Bars are cumulative: in blue are the number of responses shared between systems—there was much variability; in red the number of correct responses as judged by TREC assessors; in yellow the number above that which ought to be correct because the exact responses were judged correct in other systems; finally the total number of responses for each system. Systems are identified by their run id.

[6]We use the term "synonym" loosely here.

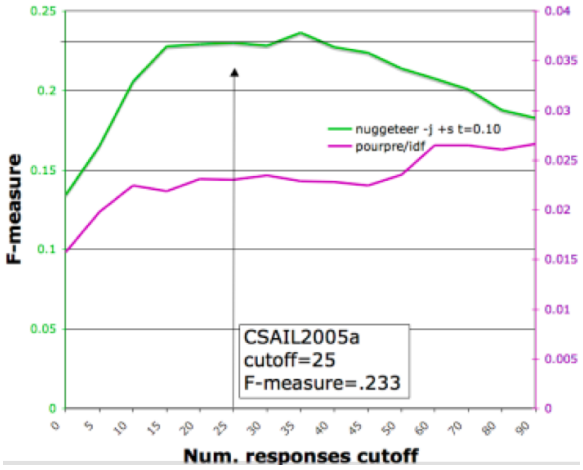

Figure 12: Answer cutoff vs. performance as estimated by Nuggeteer and by Pourpre. In hindsight we might have done better to include slightly more results.

2005)[7], to estimate performance of our automatic run at different cutoffs, as shown in Figure 12. We show additional analysis of component contributions in (Marton et al., 2006).

Using Nuggeteer, we estimate that our run with manual question analysis would have performed only slightly better than our fully automatic run, with F-measure 0.269±.0864 F-measure (recall=0.4652, precision=0.0671).

8 Contributions

We submitted three runs for the main task—summarized in Figure 13—in which we tested the effects of document retrieval and of large external resources for question answering. In particular, we:

- Tested the effect of two document retrieval strategies on document ranking and on end-to-end factoid accuracy, finding surprisingly that better document retrieval did not locally correlate with better end-to-end question answering.
- Used Wikipedia's link structure as a robust source of synonyms in a number of question answering tasks.
- Presented a new method for sentence retrieval based on a new model of synonymy and context, and showed state-of-the-art performance on the relationship task.

[7]Pourpre and Nuggeteer are both ultimately similar to the Qaviar system for automatic 250-byte factoid assessment (Breck et al., 2000), but though open source, no version of Qaviar was publicly available at the time Pourpre and Nuggeteer were developed.

References

James Allan, Courtney Wade, and Alvaro Bolivar. 2003. Retrieval and novelty detection at the sentence level. In *Proceedings of the ACM SIG in Information Retrieval (SIGIR2003)*.

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.

Matthew Bilotti, Boris Katz, and Jimmy Lin. 2004. What works better for question answering: Stemming or morphological query expansion? In *Proceedings of the SIGIR 2004 Workshop IR4QA: Information Retrieval for Question Answering*, July.

Eric J. Breck, John D. Burger, Lisa Ferro, Lynette Hirschman, David House, Marc Light, and Inderjeet Mani. 2000. How to evaluate your question answering system every day ... and still get real work done. In *Proceedings of the second international conference on Language Resources and Evaluation (LREC2000)*, June.

Aaron Fernandes. 2004. Answering definitional questions before they are asked. Master's thesis, Massachusetts Institute of Technology.

Wesley Hildebrandt, Boris Katz, and Jimmy Lin. 2004. Answering definition questions with multiple knowledge sources. In *Proceedings of the 2004 Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics Annual Meeting, HLT/NAACL-04*, April.

Boris Katz, Sue Felshin, Deniz Yuret, Ali Ibrahim, Jimmy Lin, Gregory Marton, Alton Jerome McFarland, and Baris Temelkuran. 2002. Omnibase: Uniform access to heterogeneous data for question answering. In *Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002)*, June.

	csail1	csail2	csail3
Factoid Doc Rank	Aranea2004 baseline	Aranea2005 baseline	Aranea2005 anchors
List (kb/corpus)	plain/newspaper	wiki/newspaper	wiki/wiki
Other (wiki projection)	none	BLEU	BLEU+anaphora

	Factoid accuracy	**List** F(β=1)	**Other** F(β=3)
best	.713	.468	.248
csail1	.207	**.110**	.156
csail2	**.273**	.088	**.161**
csail3	.260	.100	.160
median	.152	.053	.156

Figure 13: Experimental conditions and overall system performance in the main task.

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)*.

Jimmy Lin and Dina Demner-Fushman. 2005. Automatically evaluating answers to definition questions. LAMP 119, University of Maryland, College Park, February.

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.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. NOMLEX: A lexicon of nominalizations. In *Proceedings of EURALEX'98*, August.

Gregory Marton, Christine Moran, and Boris Katz. 2006. Component analysis of retrieval approaches to the TREC question answering track's nugget-based subtasks. ~~In~~ Submitted to *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, August.

Gregory Marton. 2006. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. Work Product 1721.1/30604, MIT CSAIL, January.

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.

Stefanie Tellex, Boris Katz, Jimmy Lin, Gregory Marton, and Aaron Fernandes. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, July.

Ellen Voorhees. 2005. Overview of the trec 2005 question answering track.