Using Semantic Overlap Scoring in Answering TREC Relationship Questions

Gregory Marton, Boris Katz

{gremio, boris}@csail.mit.edu
MIT CSAIL

Abstract

A first step in answering complex questions, such as those in the "Relationship" task of the Text REtrieval Conference's Question Answering track (TREC/QA), is finding passages likely to contain pieces of the answer—passage retrieval. We introduce *semantic overlap scoring*, a new passage retrieval algorithm that facilitates credit assignment for inexact matches between query and candidate answer. Our official submission ranked best among fully automatic systems, at 23% F-measure, while the best system, with manual input, reached 28%. We use our Nuggeteer tool to robustly evaluate each component of our Relationship system *post hoc*. Ablation studies show that semantic overlap scoring achieves significant performance improvements over a standard passage retrieval baseline.

1. Introduction

The Question Answering track of the Text REtrieval Conference (TREC/QA) (Voorhees, 2005) introduced a new task this year called the "Relationship" task, seeking ways that entities of interest can influence each other. At least one of the entities, and often a particular relationship of interest, is specified. Answers are in the form of snippets, in our case sentences, and are evaluated similarly to the definition or "other" questions in the main task, based on the number of "vital" or "okay" nuggets of information in the response vs. the length of the output. Answers must be found in or concluded from a newspaper corpus. Relationship questions are intended to more closely model real world information needs than the factoid, list, and "other" questions in the main task.

Though the task leaves open the possibility of generating answers through reasoning or summarization, in fact for this year we decided to attack only the first step: selecting those passages from the text that are most likely to contain components of the answer. Passage retrieval is not a new problem, but the evaluation criteria in this task are more stringent: unique facts ("nuggets") are rewarded rather than the passages containing them, while length is penalized.

The best previous passage retrieval methods focused on keyword coverage, finding a "hot spot" of question keywords in candidate passages (Tellex et al., 2003; Roberts and Gaizauskas, 2004). We instead score candidate passages as if we were evaluating them using standard IR measures. We break keyword matches down into a recall-like component and a precision-like component. The recall component is intended to model how much of the question is addressed. The relevant notion of precision models *not* how little extra information a candidate contains, but how well it addresses the question.

The precision component invites a straightforward model for the effect of morphological variants, synonyms, and related words. We group variants within the question and combine their recall; once we know that one variant matches, a second match to the same group intuitively adds little information. Similarly, when a candidate word or phrase matches a group in the question, then we assign full recall for that group, regardless of the quality of the match: the match quality is measured in precision.

With this precision and recall framework in place, we were able to incorporate various sources of word variation: morphological variants, synonyms, closely related words such as nominalizations, and more distantly related words from the same topic.

We attempted to filter redundant information using well-known keyword-based methods, adapted to our task. Finally, we returned the top-k passages for each question.

The resulting system left much room for improvement in absolute terms (35% nugget-recall and 6.7% nugget-precision), but it performed best among fully automatic systems in the official TREC/QA Relationship evaluation, well above its competition, and not far behind the best system that used manual input.

With only the single official score, however, it is impossible to tell what part of the architecture described above was responsible for our performance. After the fact, we were able to use our new tool, Nuggeteer (Marton, 2006a; Marton, 2006b), to evaluate performance under alternate choices. We explored the effects of:

- manual vs. automatic question analysis,
- the number of documents examined (*input depth*),
- the choice of passage scoring algorithm,
- each word variation source,
- the novelty filtering component and,
- the number of passages finally returned (*output cutoff*).

Our new model for passage scoring made the biggest difference. We substituted Clarke *et al.* 's MultiText scoring (Clarke et al., 2000) into our existing system, comparing it against our precision and recall scoring model (using keyword exact match only). Under the best settings using MultiText, output precision reached half, and recall two thirds, of the best settings under our new scoring. Manual vs. automatic question analysis also made a difference in that manual preprocessing resulted in worse performance than automatic when only exact keyword matches were allowed, but somewhat better performance when variations were allowed. The number of passages returned had a broad plateau of best performance, which included the cutoff we used for our submission. The other variables examined had little effect.

In Section 2. we describe each of our algorithms in detail. In Section 3. we discuss Nuggeteer and its application to our experiments. Section 4. contains detailed results for each experiment, and we discuss these results in Section 5.

2. Methods

Our relationship engine is a pipeline of modules that, for each question:

1. Preprocesses the question, finding key phrases and related terms
2. Retrieves documents, separating passages
3. Scores each passage
4. Filters passages for novelty
5. Selects the top-k passages

In the version that we submitted for official evaluation, we used a heuristic question analysis module (§ 2.1.), sentences as passages, a novel scoring method for each passage (§ 2.3.), a new novelty filtering algorithm (§ 2.5.), and top-24 passages selected.

2.1. Heuristic Question Analysis

We annotated non-relevant phrases from the previous year's 50 pilot questions, and iteratively removed these phrases from the start of each clause in a new question. There were a few phrases that we removed from anywhere in the question.

For the example in Figure 1, only the word "How" is removed in question analysis, but for other questions, leadins like "The analyst is interested in" were removed.

For the relevant phrases, we grouped question words and phrases that had nonzero similarity (§ 2.4.) with each other, i.e., those that were synonyms or morphological variants of each other. Each group of word or phrase variants, called *question-word groups*, was then assigned a weight equal to the sum of inversed document frequencies (*idf*s) of its members. These *overlap-weights* were normalized to one,[1] and used for document retrieval and for our *overlap-recall* calculation.

During annotation, we also marked some key background words or phrases as important, especially words in the first sentence of a multi-sentence question that provided a frame of reference. These received a boosted overlap-weight.

2.2. Document Retrieval and Passage Chunking

We used Apache's freely available Lucene search engine to index the AQUAINT collection, and retrieved documents using the (remaining relevant) keywords from the question, weighted by their phrase's or group's overlap-weight.

In our TREC/QA 2005 document ranking task experience (Katz et al., 2005), we found that specifying the overlap weights as keyword weights in the Lucene query resulted in lower document ranking performance than simply giving the original query to Lucene's default weighting system. This experience with factoid questions need not translate to a similar effect in relationship questions, but we will show that it does.

We used the freely available `Lingua::EN::Sentence` perl module to separate these documents into sentences. In one experimental condition we used paragraphs instead. We ignored documents that contained only a headline, and performed some other rudimentary filtering.

[1] In the submitted version, we normalized, squared, and renormalized the overlap-weights to sharpen the distinction between the most and least important words.

2.3. Semantic Overlap Passage Scoring

Our passage scoring algorithm has a recall-like component that we will call *overlap-recall* and a precision-like component that we will call *overlap-precision*. Ideally, if a candidate passage is relevant to all of the concepts in a question, then it will have perfect overlap-recall, and if it is relevant only to those concepts then it will have perfect overlap-precision.

We use words and phrases as proxies for concepts. *Question-word groups*, groups of related words and phrases, were identified and their *overlap-weights* assigned during question analysis. If a word or phrase from the passage has a nonzero similarity (§ 2.4.) with a group from the question, then that group's overlap weight is counted towards overlap-recall. Overlap-precision is the average similarity score of matched terms. The case where a question group is matched by multiple different candidate words bears attention: the likelihood that the terms share meaning with the question-word group is the inverse of the likelihood that *none* of the matching terms shares meaning with the question-word group, which we obtain by multiplying the dissimilarities of the matching terms.

Consider question-word groups $q_1, q_2, ..., q_m$ and associated overlap-weights $o(q_i)$, and consider a candidate $W = w_1, w_2, ..., w_n$ with similarities $s(w_j, q_i)$ to some question-word group. Let q^* be the set of groups q_i for which there exists a w_j where $s(w_j, q_i) > 0$ (matched question groups). Similarly, let w^* be the set of candidate words w_j for which there exists a q_i where $s(w_j, q_i) > 0$ (matched candidate words). Then:

$$\text{recall}(W) = \sum_{q_i \in q^*} o(q_i)$$

$$\text{precision}(W) = \frac{\sum_{q_i \in q^*} (1 - \prod_{w_j \in w^*} 1 - s(w_j, q_i))}{|q^*|}$$

We also attempted to simulate salience: if a group was matched in previous passages from the same document, and not matched in the current passage, then the current passage still gets a partial recall contribution for that group. This topic salience feature is ablated in one of our experiments.

An example of the precision and recall calculations is shown in Figure 1.

Overlap-recall is normalized by the sum of question group overlap weights, so that it is between zero and one. Overlap-precision has a range of zero to one because it is an average of similarities between zero and one. We incorporate a document score, ds, the Lucene score for the document normalized by the highest returned Lucene score.

We combine the score components using F-measure:[2]

$$F_{\beta=3}(F_{\beta=2}(\text{recall}(W), \text{precision}(W)), ds)$$

[2] F_β is meant to give its first argument β times more importance than its second argument, when combining them in a harmonic mean.

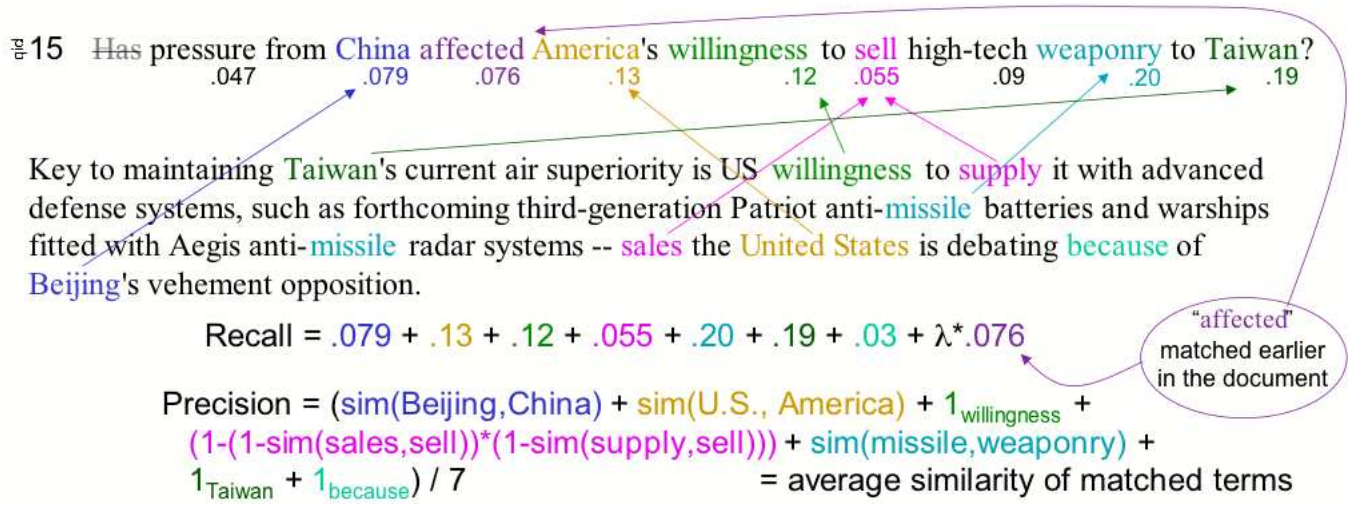


Figure 1: Recall and Precision between relationship question ID 15 and our top candidate answer. "Has" was removed as part of question analysis. "supply" and "sales" both matched "sell" in our thesaurus; their combined difference is the product of the individual differences. We use average similarity as a precision-like measure, and the sum of idfs of matched words divided by that of all topic words as recall. "Affected" was not seen in the current candidate, but was seen in the preceding sentences, and thus receives partial recall credit.

2.4. Word and Phrase Variation

In our TREC submission, we allowed four kinds of variation: morphological variants, Nomlex variants (Macleod et al., 1998), Wikipedia synonyms (see below), and variants from a small manually compiled thesaurus inspired by the "Spheres of Influence" in the Relationship task definition. Each source of variants must provide a similarity score between any pair of words or phrases it contains.

The Wikipedia (`en.wikipedia.org`) is a free online encyclopedia. Some titles redirect to an entry under a different title, e.g. USA redirects to United States. We treat these links as symmetric and use them as we would WordNet synsets. Hence the name "Wikipedia synonyms". For sources like Wikipedia that do not provide similarity scores of their own, we invented scores. For Wikipedia in particular, all pairs had similarity 1.

2.5. Output Filtering

The Novelty algorithm, described in (Marton and Moran, 2006), selects well-supported, non-redundant responses. The algorithm was inspired by a combination of the New Words and Set Difference methods described in (Allan et al., 2003), which performed best on sentences from relevant documents, where not all sentences were relevant, as is the case here.

We determine a novelty score for every candidate answer, select and report one candidate, add that to the "already selected" bag, and repeat. Words already selected are to be avoided. Words frequent in the current sample but not yet selected are rewarded. Question words are neutral because we expect to see them in every candidate, but they add no new information.

The effect in this application is mostly redundancy filtering: at each iteration we choose the candidate with the highest overlap-score, only using support to break ties, but we chose an overlap-score granularity that results in few ties.

3. Evaluation

Nuggeteer (Marton, 2006a; Marton, 2006b) is a new automatic evaluation tool for nugget-based tasks like the Relationship task. [3] Nuggeteer uses keyword recall against known answers to make binary judgements for each candidate response, as to whether it contains each possible nugget.

For the relationship task, Nuggeteer reports a perfect ranking agreement (Kendall's τ=1) on the 10 participating systems. Ranking agreement with official scores in cross-validation experiments is perfect using a variety of settings. To gauge the reliability of Nuggeteer's absolute scores, we found the square root of the mean of squared differences between Nuggeteer's reported score and the official score ("root mean squared error"). Nuggeteer's root mean squared error on this task is 1.1%—scores are, on average, one percent different from official scores. It is unknown what the variation in human agreement is, but for the similar 2003 definition task, it was measured at 10% (Voorhees, 2003), and Nuggeteer's root mean squared error for that task was 7.4%.

When reporting scores on a new system response, Nuggeteer gives a confidence interval based on the distribution of scores over questions. Confidence intervals for most Relationship systems were between five and ten percent. Nuggeteer's confidence interval for the official scores for our system is 9.4%.

Our (CSAIL) entry significantly outperformed the next best automatic run, the *uams05s* system ($p < 0.0028$) in paired evaluation (see Figure 2).

For this comparison, because we are using official judgements, root mean squared error does not come into play. For all other comparisons in this paper, it must. Effectively,

[3] Also: the 2003 TREC/QA definition task, 2004 and 2005 TREC/QA other tasks, and the AQUAINT opinion pilot

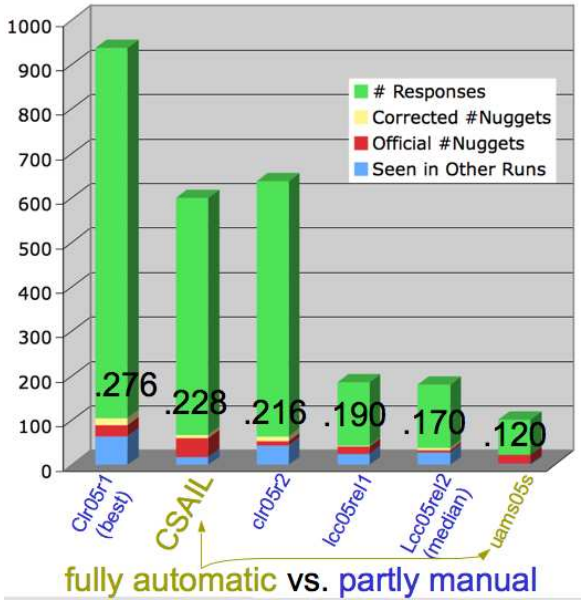


Figure 2: **Relationship system performance:** Official scores are shown for the top six systems. Bars are cumulative: in blue are the number of correct responses shared between systems—there was much variability; blue+red are correct responses as judged by TREC assessors (with numeric values shown); blue+red+yellow responses 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.

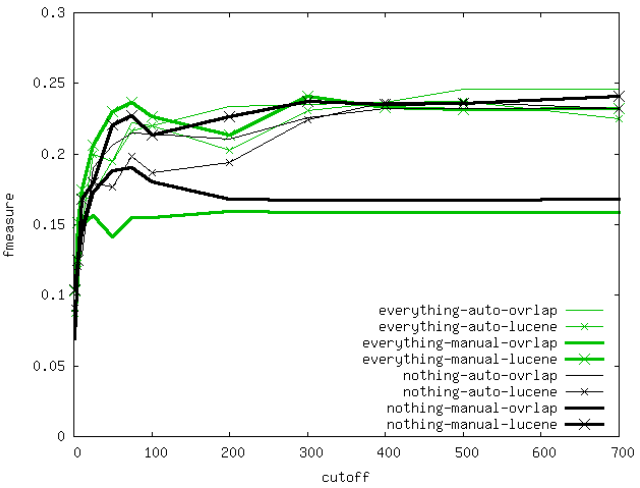


Figure 3: **Question Analysis:** Performance as a function of number of documents examined (*input depth*) while varying question analysis method (manual, bold; automatic, thin), document retrieval method (overlap-weighted, plain; default Lucene, crossed), and word variation (exact-match only, black; all variation sources, green). Question analysis interacts with document retrieval method because in manual question analysis we reweighted some terms, which caused those terms to overwhelm document results. Once we separate question analysis from document retrieval, manual question analysis makes little difference.

the 1% difference in root mean squared error can be seen as expanding Nuggeteer's confidence intervals by 1% each way. The variability in performance on each question, combined with the small number of questions, makes statistical significance between runs hard to establish in this task.
Pourpre (Lin and Demner-Fushman, 2005), another automatic measure for this task, while quite useful for rapid qualitative comparison, does not produce statistical significance results, and it is not obvious how much relative change one should expect from a change in Pourpre score.

4. Results

CSAIL's relationship engine performed well in the evaluation, though performance of all systems shows the task to be difficult (see Figure 4.). CSAIL's entry performed significantly better than the second fully automatic system. At each step of the process we evaluated variants of the submitted system to uncover which components contributed most to our systems effectiveness, and which components we might improve.
For question analysis, we compared system performance with our heuristic analysis to performance with manual question analysis (Figure 4.1.). For document retrieval, we compared a number of input depths (Figure 4.2.). For passage scoring, we compared our semantic overlap scoring metric to Clarke *et al.* 's MultiText algorithm (Figure 4.3.), also testing sentence and paragraph passages, and two minor scoring variants. For word variation we compared various sources either alone, or ablated from the submitted set of four (Figure 4.4.). For redundancy filtering, we compared our novelty filtering to no filtering (Figure 4.3.). For output depth, we present all of the other results at a variety of output depths.

4.1. Question Analysis

To test the effect of our heuristic question analysis, we manually annotated non-relevant phrases in the question set, just as we had done for the pilot questions in developing our heuristic algorithm.
Initially, manual question analysis appears to be significantly worse (see Figure 4.1., bold plain lines), but this is due to an interaction with document retrieval method. Words marked *important* during the manual question analysis process overwhelmed document results to the exclusion of other relevant terms. When this confound is removed by using the default Lucene weighting on query terms (crossed lines), then manual question analysis becomes as good as or better than automatic, as expected.

4.2. Input Depth

For pipeline-based architectures, the set of documents initially retrieved on a topic sets an upper bound on recall. To balance performance with the cost of processing more documents, we need to know how additional documents affect end-to-end performance. Of course parameters later in the process can affect how well the end-to-end system takes advantage of its input.
Figure 4.1. shows performance against input depth for eight parameter settings of our relationship engine. Among all of them, it is clear that performance changes above 300 documents are marginal. Our official run used 500 documents, as do all other experiments reported here.

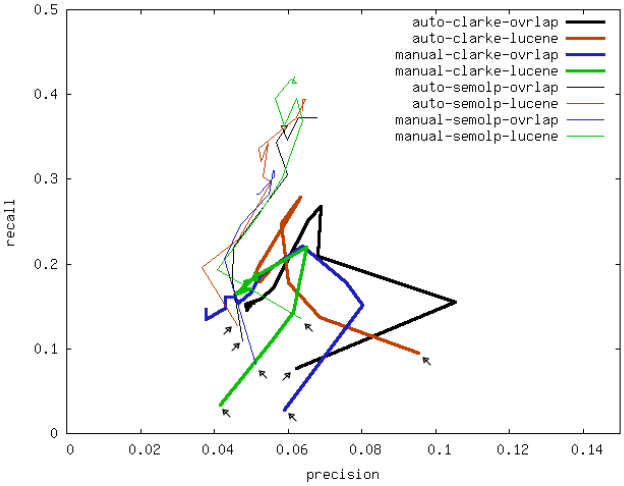


Figure 4: **Document Retrieval vs. Scoring:** Precision and recall as a function of IR input depth, under eight parameter settings. Arrows indicate fewest documents available to each run. As more documents become available, recall should improve. Under MultiText scoring (bold lines) recall improves at first, but then both precision and recall drop. Under semantic overlap scoring (thin lines) recall increases much farther. Corresponding settings of manual or automatic question analysis, and overlap weighted or default lucene retrieval, appear in the same color. These experiments do not use word variation, and all have output cutoff of 24.

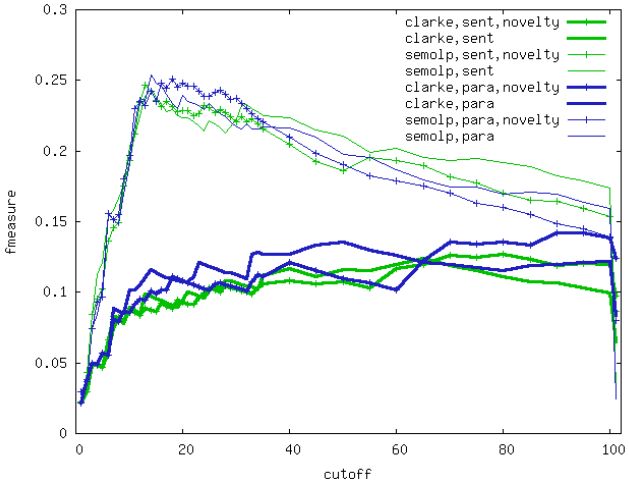


Figure 5: **Scoring:** F-measure as a function of output cutoff while varying Clarke MultiText vs. semantic overlap scoring (bold vs. thin), sentence vs. paragraph scoring (green vs. blue), and novelty filtering vs. unfiltered output (crossed vs. plain).

4.3. Scoring

We compared semantic overlap scoring to Clarke *et al.* 's MultiText scoring of the same passages, disallowing word variation for semantic overlap scoring so as to get the closest comparison to MultiText. Already in Figure 4.2. we saw that semantic overlap scoring resulted in increased performance. In Figure 4.3., we show the same trend with two extra variables, paragraph vs. sentence passages, and novelty filtering vs. no filtering. As in Figure 4.2., the only factor that makes a visible impact is Clarke (bold lines) vs. semantic overlap (thin lines) scoring.

The differences between corresponding settings at 500 input documents in Figure 4.2. are statistically significant. Uncorrected nuggeteer p-values are all less than .0052, and they are less than 0.025 when corrected for root mean squared error. The differences between corresponding settings at 24 output passages in Figure 4.3. are also statistically significant, with uncorrected p-values less than .012, and corrected p-values less than .032.[4]

Two tweaks to the passage scoring algorithm were described in section 2.3.: squaring and renormalizing the recall weights, and giving partial recall credit to keywords that appeared in prior context. Both were used in generating our submitted run. Neither one made a significant difference, but both tweaks improved absolute estimated performance slightly at peak output cutoff.

4.4. Word Variation

We measured performance for our relationship engine with no word variation enabled ("nothing"), with each of the four possible variants ("morph", "nomlex", "wiki", "thesaurus") alone, with all four enabled ("everything") and with each combination of three out of four enabled.

The results, presented in Figure 4.4., paint a surprising picture. There is a clear critical region of output cutoffs where results are best, but all-or-nothing variation makes no difference in this region. This seems to be because the different forms of variation cancel each other out in final performance. None of the differences is statistically significant.

4.5. Filtering

The novelty component made no significant difference, producing slightly better or worse results under different conditions (see Figure 4.3.).

4.6. Output Depth

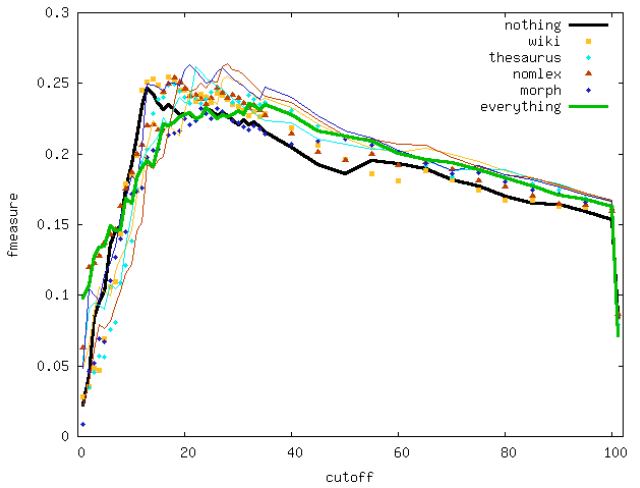
A cutoff on the number of answers is a crude method for limiting the response length, but we were able to find no other systematic strategy that was qualitatively better. We plan to explore other options, but wanted to make sure that our guessed output cutoff was not unreasonable. We submitted the top 25 system responses, which is within the region where performances are highest (see Figure 4.4.a).

5. Discussion

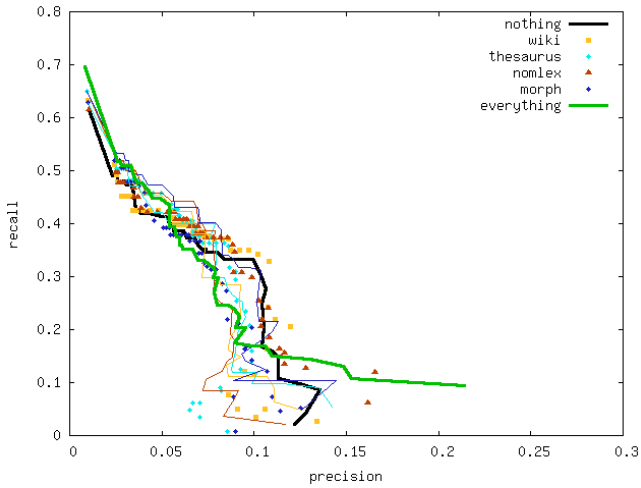
In performing this component evaluation of our 2005 TREC/QA Relationship system, we have separated those components that worked well from those that made little difference, or had a negative impact.

For the particular questions posed, heuristically removing extraneous verbiage from the question performed about as well as doing so manually. Our system was apparently examining enough documents, and it is interesting to see how the different components respond to different amounts of input. Choosing paragraph or sentence passages made little difference, though this barely begins to explore the range of possibilities for synthesizing a response. The minor components of our scoring may have contributed positively to

[4] All p-values were less than .01 except paragraph,no-novelty.



(a)



(b)

Figure 6: **Word variation:** bolder lines show exact match only ("nothing") and all sources used ("everything"). Each of the sources has its own color, with points representing the source's performance alone, and plain thin lines representing the performance of the other three sources (without this one). We visualize results in terms of (a) output cutoff vs. F-measure, and (b) precision vs. recall, where each line proceeds from top-1 output cutoff (lower right in (b)) to unlimited output (shown just beyond 100 in (a)). Sources appear to cancel each other out, with each conferring marginal improvement, and any three doing better than all.

performance, but not significantly so. Our novelty filtering algorithm did not make a significant impact on overall performance. We chose a resonable number of responses to return, which falls within a plateau of peak performance for our scoring metric, though a single global threshold may not be the best strategy overall.

Most interestingly, we introduced a new passage scoring algorithm, *semantic overlap scoring*, that separates the notion of covering all components of a question (recall) from matching each one closely (precision). This model worked quite well even with exact keyword matches, because we could set the importance of recall vs. precision, in this case strongly favoring recall, so the strong presence of one or a few keywords was insufficient for a high score.

The separation of recall and precision also allowed us to define a model for synonymy, where synonyms contribute fully to recall, and contribute to precision proportionally to a similarity score with question keywords. Many other models of synonymy are possible with semantic overlap scoring, and the combination of this model with the particular choice of word variant sources did not create a significant improvement. However, we are planning to pursue other sources of variants and other models of language variation to plug in to semantic overlap scoring.

We hope that semantic overlap scoring will prove to be a useful formulation of the role of language variation in information retrieval.

6. References

James Allan, Courtney Wade, and Alvaro Bolivar. 2003. Retrieval and novelty detection at the sentence level. In *Proceedings of 26th Annual ACM SIGIR Conference*.

Charles L. A. Clarke, Gordon V. Cormack, Derek I. E. Kisman, and Thomas R. Lynam. 2000. Question answering by passage selection (MultiText experiments for TREC-9). In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*.

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 for question answering. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2005)*, November.

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

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

Gregory Marton and Christine Moran. 2006. Component analysis of retrieval approaches to the TREC question answering track's nugget-based subtasks. In *Proceedings of the 29th Annual ACM SIGIR Conference (submitted)*.

Gregory A. Marton. 2006a. Nuggeteer: Automatic nugget-based evaluation using descriptions and judgements. In *Proceedings of the North American Association for Computational Linguistics and the Human Language Technologies Conferences (NAACL/HLT2006)*, June.

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

Ian Roberts and Robert Gaizauskas, 2004. *Evaluating Passage Retrieval Approaches for Question Answering*, volume 2997/2004, pages 72–84. Springer-Verlag GmbH.

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*, July.

Ellen Voorhees. 2003. Overview of the TREC 2003 question answering track.

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