Twease at TREC 2006: Breaking and fixing BM25 scoring with query expansion, a biologically inspired double mutant recovery experiment.

Kevin C. Dorff[1†], Matthew J. Wood[1,2] and Fabien Campagne[1,2†*]

[1]Institute for Computational Biomedicine and [2]Dept. of Physiology and Biophysics, Weill Medical College of Cornell University; 1300 York Ave; New York, NY 10021, USA.

Abstract. This is the first year that our group has participated in the genomics track of TREC. We enter the evaluation with a new biomedical search engine, developed as an open-source project which relies heavily on MG4J, and is publicly available at http://www.twease.org. We designed our runs to test the features of Twease that most distinguish it from other biomedical search engines. Such features include: (1) a custom biomedical query expansion module (which draws on biomedical thesauri, but also includes a statistical model of morphological word variations observed in the biomedical domain); (2) the ability to search the index simultaneously at the whole document level, or at the sentence level. Our official runs evaluated the performance of minimal interval semantics when used with custom morphological word expansion on a biomedical corpus. Our best official run scored MAP=0.30 at the document level, slightly above the median of other submissions. Our non-official runs compared minimal interval semantics to BM25 on the same topics and corpus. We varied the level of query expansion for both methods, and demonstrated how easily BM25 breaks down as more related terms are added to a query, while minimal interval semantics is robust to such change. We investigated the origin of the issue with a biologically inspired experimental design (mutation recovery). Our results help understand why certain groups observe performance drops with thesaurus-based query expansion, while other groups report the opposite effect. The best of our non-official runs achieves a MAP document level of 0.32 when an intermediate level of query expansion is used. This manuscript also describes our first attempt at passage retrieval in full text articles (we achieved a MAP of 0.052, above the median of 0.028).

Introduction

We enter the TREC 2006 genomics track with the Twease biomedical search engine [1]. Twease is a web front-end to inverted indices prepared with MG4J [2] and offers features customized for biomedical users that build upon and extend MG4J. The following extended features implemented in Twease were exercised in TREC 2006:

- Interactive query expansion, from multiple sources of expansions:
    - Morphological word variants are discovered at run-time with an algorithm customized for biomedical corpora (Campagne, F manuscript in preparation). While similar in purpose to stemming, word variant discovery supports discrete levels of query expansion at user-adjustable thresholds.
    - Short words are indexed in a case-sensitive manner and users can choose to (i) either include all case variants of a word in a search or (ii) include only some variants.

[†] These authors contributed equally to this work. *To whom correspondence should be addressed (fac2003@med.cornell.edu).

    - Quoted phrases are automatically expanded to the common abbreviations found in the corpus (e.g., “Protein Kinase C” will produce query expansion terms such as PKC, PK-C, aPKC, cPKC, pkC or nPKC).
    - Quoted phrases are used to lookup equivalent MeSH term entries which can be used as expansions.
    - Gene names are expanded to their HUGO synonyms.
- The proportion of terms used for interactive query expansion is exposed in the web user interface (see the Slider at the top right of the Twease search engine).
- BM25ec scoring (see below).
- Indices can be constructed such that whole document level or sentence-level searches can be performed at runtime.

This manuscript reports on a feature recently introduced in Twease after the submission of our official runs: namely the BM25ec scorer. The BM25ec scorer, briefly described in this manuscript, is a modification to the Okapi BM25 scoring approach [3, 4] that helps maintain retrieval performance when additional terms are included in the query, for instance when biomedical thesauri are used, or when post-indexing stemming techniques are used. This manuscript describes an experimental design inspired from biology that characterize and suggest reasons why (1) the performance of BM25 with query expansion degrades as more query expansion terms are included in the query, (2) query expansion terms must be assigned smaller weights than primary query terms. (3) shows how performance can be recovered with a simple modification to the BM25 scorer when relationships among expansion terms and terms of the query are known.

ICB TREC genomics track 2006 runs

Table 1 summarizes our official and non-official runs. The rows of this table marked ‘batch-‘ represent several individual runs where we varied the value of a parameter in the way indicated. For instance, slider 0-200 in row batch-1 indicates that we varied the value of slider parameter from 0 to 200.

The parameters that were varied during our runs are:

- Slider: the amount of query expansion (Values in the range 0 to 200, 200 indicates maximum query expansion; a range indicates that we varied the slider value by 20 unit increments. The run labeled batch-1 therefore includes 11 individual runs).
- Scoring scheme: Vigna (minimal interval semantics) as described in [5], BM25 [3, 4] (such as implemented in MG4J, and using parameters k=1.2, b=.5, suggested by Clarke for TREC GOV2), or BM25ec (this manuscript).
- Type of query preparation: manual or automatic (see section Query Preparation).
- Synonyms: yes/no, whether synonyms from established biomedical resources were used (when yes, terms from HUGO and MESH are used to expand queries).

Query Preparation

Manual. Manual queries were crafted from the topics’ narrative. MJW and FC used their knowledge of biology to formulate queries likely to yield relevant documents. Queries were performed against the TREC-gen 2006 corpus to make sure they returned results at full slider recall. Queries that returned no or very few results where reformulated.
Automatic. Narratives were filtered for stop words (words that occur in more than 50% of the documents in the corpus were considered stop words). Remaining words where filtered unique

and ordered by increasing corpus frequency. The eight words with the lowest corpus frequencies were used to create a fully disjunctive query (i.e., A | B | C| D | E). Our automatic queries therefore included at most eight words from the topics.

Table 1. Summary of experimental conditions.

run-id	official	scoring	slider	run type	synonyms	comment
icb1	yes	Vigna	200	manual	yes	Whole document level
icb2	yes	Vigna	200	manual	yes	Sentence-level restrictions and whole document scoring.
icb3	yes	Vigna	200	manual	yes	Context queries
batch-1	no	BM25ec	0-200	automatic	no	Whole document level, BM25 scoring with class equivalences. Morphological word variants, case variants, abbreviations activated.
batch-2	no	BM25ec	0-200	automatic	yes	Same as batch-1, with MESH and HUGO synonym expansion activated.
batch-3	no	Vigna	0-200	manual	yes	Whole document level. Minimal interval semantics.
batch-4	no	BM25	0-200	automatic	no	Same as batch-1, but standard BM25 scoring, no equivalence classes.
batch-5	no	BM25	0-200	automatic	yes	Same as batch-2, but standard BM25 scoring, no equivalence classes.

Results

Table 2 Scored Runs

run-id	slider	document-level MAP	Passage-level MAP	Aspect-level MAP
icb1	200	0.300	0.052	0.110
icb2	200	0.185	0.035	0.078
icb3	200	0.115	0.008	0.031
batch-1-best	160	0.311	N/A	N/A
batch-2-best	80	0.318	N/A	N/A
batch-3-best	60	0.325	N/A	N/A
batch-4-best	0	0.254	N/A	N/A
batch-5-best	0	0.254	N/A	N/A

Document vs. sentence-level searches. Table 2 presents an overview of the results obtained when each run is evaluated against the gold standard. In summary, the results indicate that whole document-level searches (icb1) on average clearly outperform sentence level searches (icb2) and the specific type of context level search that we tried this year (icb3). Indeed, icb1 achieves a document-level MAP of 0.300, while the average MAP for sentence-level searches is only 0.185. While average MAP is decreased by sentence-level searches, we note that sentence-level searches can outperform abstract-level searches on some individual topics (i.e., see the topics marked in italics in Supplementary Table 1 on http://icb.med.cornell.edu/ supplementary_materials/ trec_2006). This happens in about 25% of the topics.

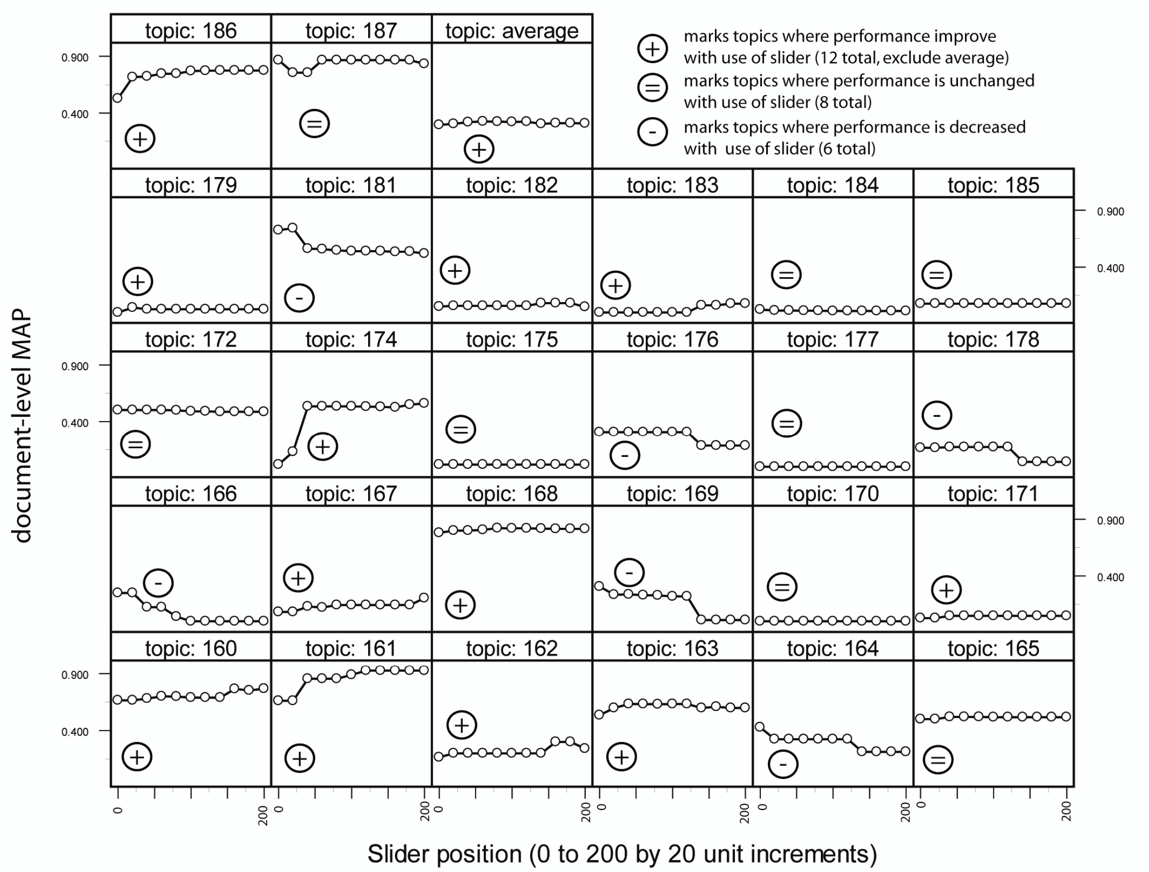

Figure 1. Parameter scan for the Twease slider property. Performance variations are shown for each topic and for the average of all topics. Results are from run batch-3 described in Table 1 and text. This figure illustrates how performance varies with the level of query expansion applied to each topic.

User adjustable query expansion level. Since Twease supports user adjustable levels of query expansion (through the Twease slider, top right of the user interface), we evaluated the impact of the slider parameter on performance retrieval. Data in Table 2 indicates that the slider parameter value that yields the best performance is 60, corresponding to the slider at 30% (0% corresponds to the slider fully towards precision). We conjectured that different slider positions would be optimal for different topics and this conjecture drove the design of the Twease user interface. Figure 1 plots the document-level MAP for each topic and value of the slider. This Figure shows that indeed, performance on distinct topics reacts differently to various levels of query expansion. In the TREC-gen 2006 evaluation, 12 topics (46%) benefit from the use of the Twease slider, eight topics (31%) are not affected and six topics (23%) incur a performance decrease when the Twease slider is moved away from precision (see Figure 1). These data empirically confirm our conjecture and indicate that there is no level of query expansion that is optimal for each one of the 26 topics in this year's evaluation. This suggests that users would benefit from applications where the level of query expansion can be adjusted. Whether the user has enough information to adjust the slider to an optimal value in the interactive setting provided by Twease is unclear at this stage.

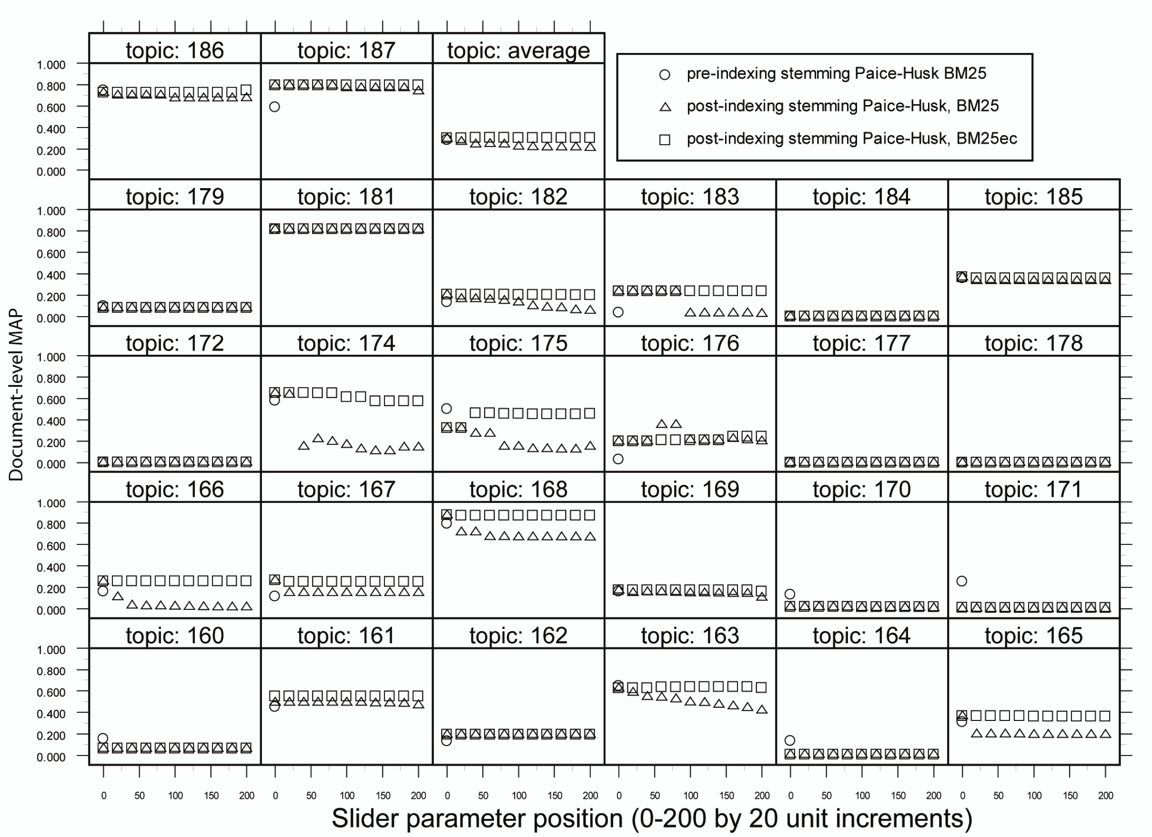

Figure 2. Breaking and fixing BM25, a double mutant recovery experimental design. The data point shown as a circle is the baseline obtained by stemming each word of the corpus before indexing. The points shown as triangles plot the change in performance seen when stemming is done post-indexing and scoring is done with the BM25 scorer. This curve corresponds to the first experimental condition, where we introduce a factor that breaks the performance of the BM25 scorer (observe how the performance drops as more query expansions are introduced for many topics). The points shown as squares plot the results obtained when the BM25ec scorer is used with post-indexing stemming. Topics 160, 170, 171 and 175 represent conditions where the word similarity search fails to retrieve words similar to the query word, which have the same stem as the query word, and improve performance when included in the query. Word similarity search is used for both post-indexing stemming runs and therefore cannot affect the comparison between BM25 and BM25ec. These data demonstrate that a non-intuitive drop of performance occurs when BM25 scoring is used with post-indexing query expansion, and shows that the BM25ec scorer (introduced in this manuscript) can recover performance.

BM25 and post-indexing stemming. While our official runs only used minimal interval semantics (using the Vigna scorer implemented in MG4J), our follow up runs include runs scored with BM25 (introduced to MG4J in version 1.1). We initially tested the MG4J BM25 scorer on TREC-gen 2005, and discovered that increasing the level of query expansion (higher values of the Twease slider parameter) decreased performance. Because more query expansion was generally beneficial with the minimal interval semantic scorer, this result was very counter-intuitive. We therefore performed additional experiments to try and understand the origin of this effect. We found that using the Paice-Husk stemmer prior to index construction helped both BM25 and minimal interval semantics scoring, but that expanding the query post-indexing, with terms that shared the same Paice-Husk stem [6], decreased performance for BM25. Since minimal interval semantic scoring benefited irrespectively of whether stemming was done before or after indexing, the origin of the drop in performance was clearly due to the combined use of the BM25 scorer with the process of query expansion. Similar results were observed on data from TREC-gen 2006, and are shown in Figure 2. Figure 3 shows that performance also decreases when another source of query expansion terms is used (i.e., synonyms from HUGO or MeSH) with BM25 scoring.

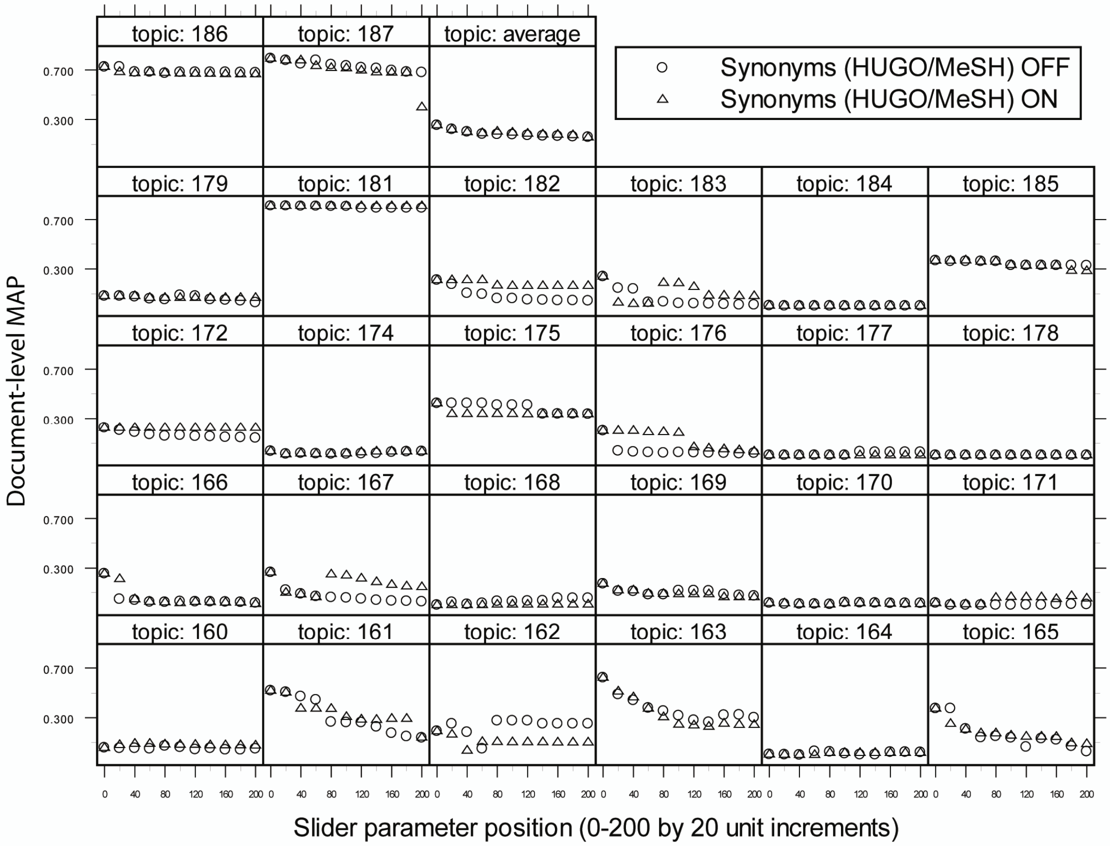

Figure 3. BM25 scorer used with increasing levels of query expansion. The slider parameter controls the level of query expansion. Synonyms obtained from HUGO and MeSH are one source of such expansions. Other sources include morphological word variants and abbreviations discovered for phrases. Performance of topics are plotted separately or as an average across topics. These plots show that BM25 scoring performance declines as more expansion terms are included in the query.

Stemming/Query Expansion before or after indexing. When stemming is performed before indexing, all the terms that share the same stem are represented by their stem in the index. This means that the BM25 scorer will use the frequency of the stem (i.e., the frequency of 'S_1 OR S_2 OR …' over the corpus where S_1, S_2, …, are terms that share the same stem) when scoring documents. We wish to give end-users the control over which words are included in the query. Therefore, in Twease we perform stemming after indexing.

Stemming after indexing requires expanding a two words disjunctive query with two words A and B:

```
A OR B
```

To:

```
(A1 OR A2 OR…) OR (B1 OR B2 OR…)
```

Where the words A1, A2 share the same stem as A and B1, B2 share the same stem as B.

Query expansion done in this way has non-intuitive consequences when scoring with BM25, as the experiments described in the previous section illustrate. Indeed, instead of scoring documents against two terms A and B, with frequencies f(A)= $f(A_1 \mid A_2 \mid ...)$ and f(B)= $f(B_1 \mid B_2 \mid ...)$, the scorer now considers $i+j$ terms with individual frequencies $f(A_1),...f(A_n)$ and $f(B_1),...f(B_n)$. To understand why this may deteriorate retrieval performance, consider what happens with a query such as 'BRCA1 OR ubiquitinating'. Words that share the same stem as "ubiquitinating" (106) include "ubiquitination" (3458), "polyubiquitinated"(273), or "ubiquitinatable" (5) (words are listed with their individual frequency in Medline at time of writing). When stemming is applied before indexing, the frequency of the stem "ubiquitin" is 3458+273+106+5=3842. However, when the query is expanded after indexing, the BM25 scorer sees the query 'BRCA1 OR (ubiquitinating OR ubiquitination OR polyubiquitinated OR ubiquitinatable OR …)'. Since the word "ubiquitinatable" has low frequency, it is weighted more by BM25 than "ubiquitination" or "BRCA1". This can result in ranking documents that contain rare word variants better than documents that contain common word variants. Since the morphological word variant discovery module of Twease can discover very many variants with low frequency, this can be a serious problem. Further, the words that are added to the query are strongly correlated, a property that violates the term independence assumption of the BM25 scoring model [4].

BM25ec: Term Equivalence Classes for BM25 scoring. To counter this effect, we extended the MG4J BM25 scorer to support the definition of term equivalence classes. An equivalence class groups terms so that count and frequency information can be accumulated before scoring (correcting the low frequency problem described above). The equivalence class regroups together terms that are strongly dependent because they are all expanding the same word (so that the counts and frequency provided to the BM25 scorer are independent). Different strategies can be used to accumulate counts and frequencies. Here, we define the count of an equivalence class in a document to be the sum of the counts of the terms that belong to the class in the same document. Ideally, we would define the frequency of the class C as the frequency of the query 'C1 OR C2 OR …' with Ci terms that belong to C, but this is somewhat inefficient because it requires doing an extra query for each class. Instead, and as a first approach, we define the frequency of class C as $Max\{f(C_1), f(C_2),...\}$. (After distribution of the notebook version of this paper, we also tried $Average\{f(C_1), f(C_2),...\}$ and found this alternative to slightly outperform Max on the TREC-gen 2006 task.) Assuming the query described above as an example, the frequency of the class of terms that correspond to the word “ubiquitinating” is taken to be 3,458. The scorer that implements equivalence classes is noted BM25ec in this manuscript.

Equivalence classes fix BM25 scoring. When used on TREC-gen 2005, the BM25ec scorer corrects for the drop in performance observed when varying the level of query expansion. Development and tuning of BM25ec was done on TREC-gen 2005. Here, we report how BM25ec improves performance when used with query expansion on the independent corpus TREC-gen 2006 (see Figures 2, 3 and 4). Runs on TREC-gen 2006 were performed before the genomics track judgments were distributed, and constitute a strong confirmation of our findings.

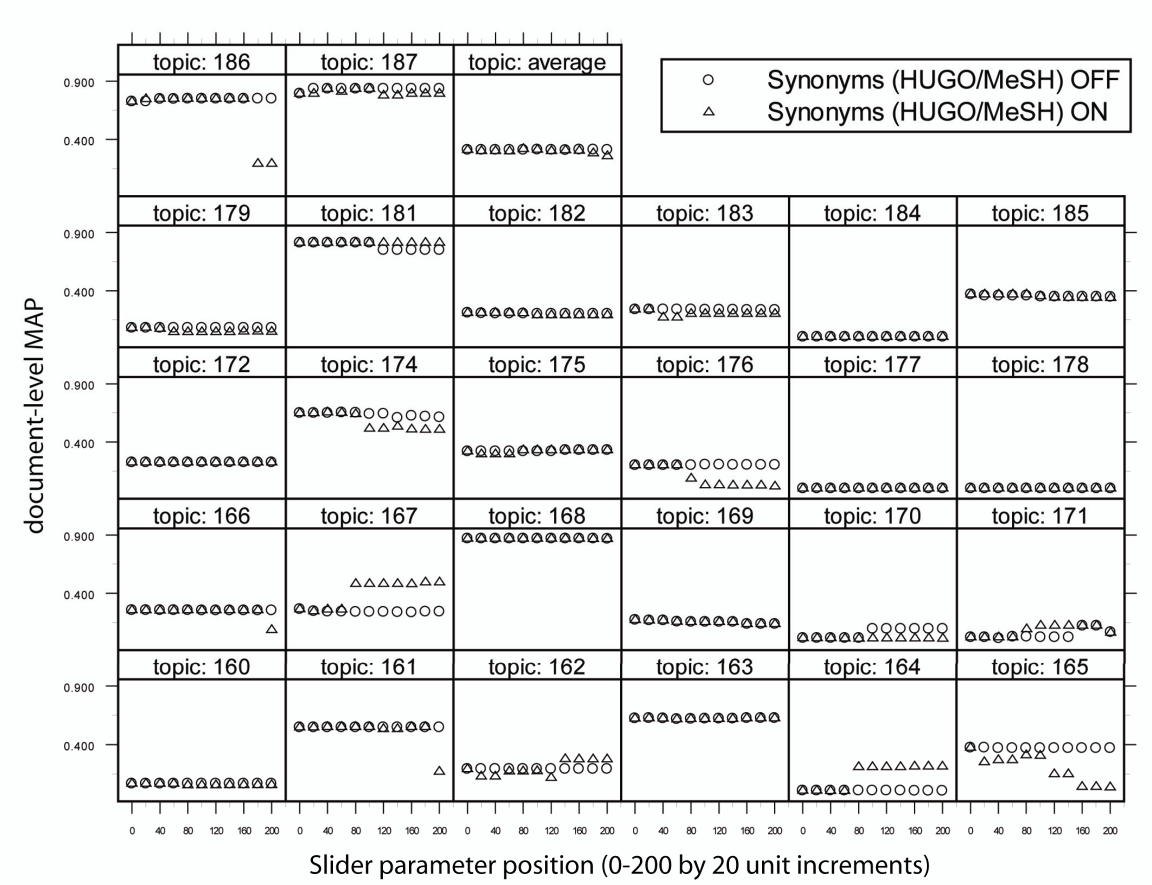

Figure 4. BM25ec scorer used with increasing levels of query expansion. The slider parameter controls the level of query expansion. Synonyms obtained from HUGO and MeSH are one source of such expansions. Other sources include morphological word variants and abbreviations discovered for phrases. Performance of topics are plotted separately or as an average across topics. These plots show how term equivalence classes maintain the performance of BM25ec scoring as more terms are used to expand the queries. Compare the profile of topics 161, 163, 165, 167 with those on the same topics in Figure 1 where standard BM25 is used.

Impact of synonym expansion. Batches of runs batch-1 and batch-2 indicate that synonym expansion (with HUGO and MeSH) is marginally beneficial: MAP increases from 0.311 to 0.318 when synonyms are included in the BM25ec search. This result independently confirms the results reported at TREC 2004 [7, 8].

Minimal interval semantics or BM25ec. Figure 5 presents a comparison of BM25 and minimal interval semantic on TREC-gen 2006. The plots show how much of the relevant documents are identified by each scoring method, at a given result rank. The black points represent an ideal combination of the minimal interval semantic (Vigna, green) and BM25 scorers (red). Topics 163, 165, 169, 172 and 176 could benefit from a combined scorer because the black curve (ideal union of results from Vigna and BM25) is above both the red and green curve. For most topics, however, one scorer is limiting (the black curve overlaps with the red or green), and a combination would not be expected to improve performance significantly. Since the plots are normalized to the total number of relevant document by topic, the position of the plateau and the fact that most topics reach a plateau early indicate that the most promising strategy to improve the performance of our search engine is to identify suitable query expansion words not provided in the topic narratives.

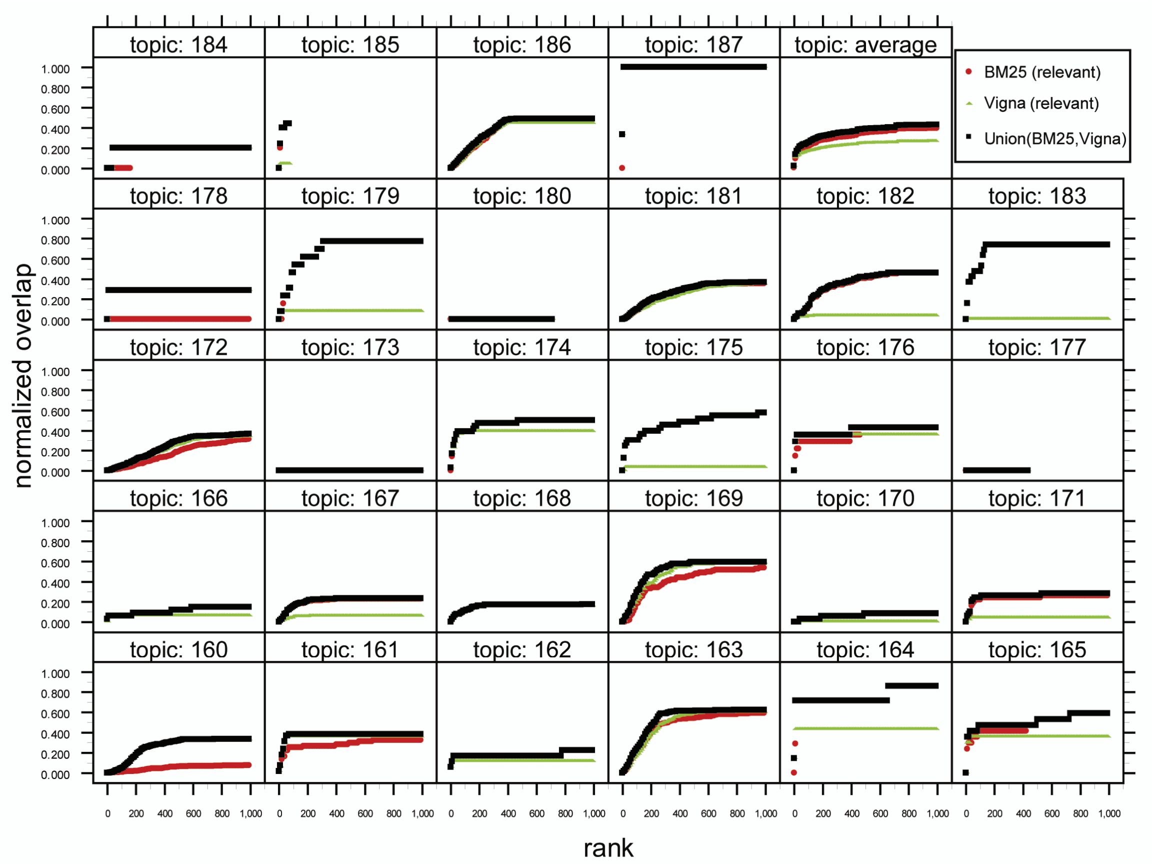

Figure 5. Contributions of BM25 scoring and minimal interval semantic scoring (Vigna scorer) to the relevant documents retrieved by our best runs. (A color version of this Figure appears at http://icb.med.cornell.edu/supplementary_materials/trec_2006)

Preparing TREC-gen passage submissions. We developed the locator tool to map evaluation results with text snippets to byte spans in the corpus input documents (HTML files). The program is distributed at http://chagall.med.cornell.edu/trec-gen/2006/locator.html and has been used by other groups to prepare their final TREC-gen 2006 submissions. The latest version of the locator tool uses dynamic programming to align the words of the text snippets to the HTML content of the input document (we calculate the longest common subsequence between the text snippet and the whole document). This allows mapping text snippets even when the snippets span HTML tags or special characters (e.g., HTML codes Unicode characters with the multi-character &xxx; syntax).

Passage Retrieval. In 2006, TREC-gen evaluated passage and aspect retrieval in addition to document level retrieval. We did not optimize Twease for passage retrieval, but instead used as passages the minimal intervals identified by the Vigna minimal interval semantic scorer (implemented in MG4J). Passages that were evaluated in our official run (icb-1) are thus the same that end-users of Twease can see marked in bold in the text snippets returned by the Twease user interface. Our run icb-1 scored at 0.052 for passage MAP (the median passage MAP for other runs was 0.0277/manual, interactive and 0.0240/automatic), about double the median performance obtained by other runs. This result strongly suggests that minimal interval semantic is a competitive option for passage retrieval. This would be an attractive solution because minimal interval semantics scoring is scalable and fast (as can be seen on Twease.org where the same technique is applied to Medline). The performance of run icb-1 on aspect retrieval was close to the median of other runs, at 0.110 aspect-level MAP. This is not surprising since we did not try to cluster passages by aspect and simply report passages in the order in which the minimal interval scorer ranks them.

Conclusions

Our first participation to TREC helped us study some of the design decisions that guided the development of Twease.org. A key observation made possible by the TREC relevance judgments is that the approaches used in Twease.org obtain performance levels which are marginally above the median of other runs. In other words, while this year Twease.org is not competitive with the best research retrieval engines that entered the competition, it scored better than 50% of the engines evaluated. The detailed analysis of our runs presented in this manuscript suggests strategies for future improvements. These strategies include (1) identify words not in the narratives that relevant documents are likely to contain (implementing techniques such as pseudo relevance feedback) and clustering of words into equivalence classes; (2) improving automated query preparation from topics so that minimal interval semantic can be used in an automatic mode (3) combining BM25 and minimal interval semantic scoring.

Acknowledgments

We thank Sebastiano Vigna and Paolo Boldi for stimulating discussions and for their quality open-source releases of the MG4J indexing software. We thank Marko Srdanovic for help preparing some of the index collections used in the post-submission runs.

References

1. Wood M, Dorff K, Paolo Boldi P, Vigna S, Campagne F: **Twease: Searching Medline, one Sentence at a Time.** . *Manuscript submitted for publication* 2006.
2. Boldi P, Vigna S: **MG4J at TREC 2005**. In: *Text Retrieval Evaluation Conference (TREC): 2005*; 2005.
3. Sparck Jones K, Walker S, Robertson SE: **A probabilistic model of information retrieval: development and comparative experiments, Part 2.** *Information Processing and Management* 2000, **36**:809-840.
4. Sparck Jones K, Walker S, Robertson SE: **A probabilistic model of information retrieval: development and comparative experiments, Part 1.** *Information Processing and Management* 2000, **36**:779-808.
5. Boldi P, Vigna S: **Efficient lazy algorithms for minimal-interval semantics.** : Springer−Verlag; 2006.
6. Paice CD: **Another stemmer**. In: *SIGIR Forum 1990; 24(3), 56--61*; 1990: 56--61.
7. Darwish K, Madkour A: **The GUC goes to TREC 2004: using whole or partial documents for retrieval and classification in hte genomics track.** In: *The thirteen Text REtrieval Conference (TREC 2004): 2004; Gaithersburg*: Notebook paper.; 2004.
8. Pirkola A: **TREC 2004 Genomics Track Experiments at UTA: The effects of Primary keys, bigram phrases and query expansion on retrieval performance.** In: *The thirteen Text REtrieval Conference (TREC 2004): 2004; Gaithersburg*; 2004.