# Task-Specific Query Expansion (MultiText Experiments for TREC 2003)

David L. Yeung  Charles L. A. Clarke  Gordon V. Cormack  Thomas R. Lynam  Egidio L. Terra

School of Computer Science, University of Waterloo, Canada
mt@plg.uwaterloo.ca

## I. Introduction

For TREC 2003 the MultiText Project focused its efforts on the Genomics and Robust tracks. We also submitted passage-retrieval runs for the QA track. For the Genomics Track primary task, we used an amalgamation of retrieval and query expansion techniques, including tiering, term re-writing and pseudo-relevance feedback. For the Robust Track, we examined the impact of pseudo-relevance feedback on retrieval effectiveness under the new robustness measures.

All of our TREC runs were generated by the MultiText System, a collection of tools and techniques for information retrieval, question answering and structured text search. The MultiText Project at the University of Waterloo has been developing this system since 1993 and has participated in TREC annually since TREC-4 in 1995.

In the next section, we briefly review the retrieval methods used in our TREC 2003 runs. Depending on the track, various combinations of these methods were used to generate our runs. The remaining sections describe our activities for the individual tracks, with the bulk of the report covering our Genomics Track results.

## II. Retrieval Methods

The MultiText System implements a variety of retrieval methods, three of which were used in our TREC 2003 experiments:

1) *Shortest Substring Ranking* (SSR), a ranked retrieval method for extended boolean queries [1];
2) Qap, a passage-retrieval technique originally developed for question answering [3], [4];
3) Okapi BM25 [8].

Our Genomics runs use a combination of SSR and Okapi methods, generating and executing a number of different term sets and boolean queries, and merging the results to produce a final ranked document set. Our Robust runs use Qap to generate passages for pseudo-relevance feedback and Okapi to evaluate the expanded queries. For our QA track runs, 250-byte answer passages are selected from passages generated by Qap.

In the next few sections, we provide brief overviews of the retrieval methods. Please consult the associated references if further details are required.

### A. *Shortest Substring Ranking*

The MultiText project has used variants of the SSR algorithm in TREC experiments since TREC-4 in 1995. The SSR algorithm operates by locating passages that satisfy a boolean query. A passage may consist of any substring of any document in the target corpus. The algorithm identifies all document substrings that satisfy the query and do not contain shorter substrings that also satisfy the query. This *shortest substring rule* serves to limit the number of passages that must be considered by the SSR algorithm. For ranking purposes, a document's score is computed from the lengths of the passages contained within it.

For example, along with other queries, our system generates the following query for Genomics topic 23:

"c gamma" ^ ("phospholipase"+"phospholipases")

where the + symbol represents boolean OR and the ^ symbol represents boolean AND. Since the algorithm locates the shortest substrings that satisfy the query, a passage located by the algorithm will begin (or end) with the phrase "c gamma" and end (or begin) with one of the words "phospholipase" or "phospholipases". None of these terms will appear elsewhere in the passage, since otherwise the passage would contain a shorter substring that also satisfies the query.

In some cases, structural constraints are applied to the query. For example, the query:

("<NameOfSubstance>".."</NameOfSubstance>")>"cip1"

identifies instances of the NameOfSubstance field that contain the term "cip1", where the > symbol is used to express the CONTAINS relationship.

Assume that document $d$ contains passages $P_1, P_2, ..., P_n$, sorted by increasing length, with each passage satisfying the query under the shortest substring rule. We compute a score for $d$ that rewards shorter passages and documents that contain more passages. For a passage $P$ define:

$$I(P) = \begin{cases} \frac{\mathcal{K}}{l(P)} & \text{if } l(P) \geq \mathcal{K} \\ 1 & \text{if } l(P) \leq \mathcal{K} \end{cases} \qquad (1)$$

where $l(P)$ is the length of $P$ as measured by the number of alphanumeric tokens it contains. For any passage $P$, we have $0 < I(P) \leq 1$. The score for $d$ is then computed by the formula:

$$\sum_{i=0}^{n} I(P_i)^{\gamma} \quad (2)$$

For our TREC 2003 experiments we use the parameters $\mathcal{K} = 16$ and $\gamma = 0.5$.

SSR is most valuable when a boolean query matches a large number of documents. However, in many cases, a boolean query matches few or no documents. To address this case, we often generate *tiers* of boolean queries, with earlier tiers providing higher precision and later tiers providing higher recall. The tiers are executed in order, with the documents generated by earlier tiers ranked before the documents generated by later tiers. Once a document is generated by a tier, it is eliminated from later tiers. Section III-B describes the tiered boolean queries used in our Genomics Track runs.

A complete discussion and analysis of the SSR algorithm may be found in Clarke and Cormack [1]. That paper also provides an efficient algorithm to implement SSR.

### B. Passage Retrieval

The Qap passage retrieval algorithm is related to the SSR algorithm, in that it may return any substring of any document. The algorithm locates "hotspots" within the corpus where query terms cluster in close proximity. The score of a hotspot is based on its length and the weights of the terms occurring within it. A hotspot is usually less than 50 words in length, but may be longer. It may start or end at any word and is not constrained by sentence or paragraph boundaries. At most one hotspot is selected from a document, since additional hotspots from the same document may not exhibit the independence properties assumed by our feedback and QA methods.

Given a query $Q$, document substring $H$, and a term set $T \subseteq Q$, we compute a score for $H$ as follows:

$$\sum_{t \in T} \log(N/f_t) \ - \ |T| \log(l(H)) \quad (3)$$

where $f_t$ is the total number of times $t$ appears in the corpus and $N$ is the total length of all documents in the corpus. In effect, the Qap algorithm considers every substring of every document in the corpus and locates the $m$ substrings with the highest score. Details of the Qap algorithm, its efficient implementation and its application to question answering may be found in Clarke et al. [4].

### C. Okapi

Our implementation of Okapi BM25 follows the description of Robertson et al. [8] with standard parameters: $k_1 = 1.2$, $b = 0.75$, $k_2 = 0$, $k_3 = \infty$. Specifically, given a term set $Q$, a document $d$ is assigned the score

$$\sum_{t \in Q} w^{(1)} q_t \frac{(k_1 + 1) d_t}{K + d_t} \quad (4)$$

where

$$\begin{aligned} w^{(1)} &= \log\left(\frac{D - D_t + 0.5}{D_t + 0.5}\right) \\ D &= \text{number of documents in the corpus} \\ D_t &= \text{number of documents containing } t \\ q_t &= \text{frequency that } t \text{ occurs in the topic} \\ d_t &= \text{frequency that } t \text{ occurs in } d \\ K &= k_1((1-b) + b \cdot l_d / l_{avg}) \\ l_d &= \text{length of } d \\ l_{avg} &= \text{average document length} \end{aligned}$$

As an extension, our implementation of BM25 allows phrases (and extended boolean queries) to be used as query terms, a facility used in our Genomic runs to allow multi-token gene names and bigrams extracted from gene names to be treated as individual terms.

## III. GENOMICS TRACK - METHODOLOGY

For the Genomics track we experimented with a number of different retrieval, feedback and fusion techniques. The following sections describe the various aspects of the experimental methodology. Analysis of the results obtained from the training data is found in the Section IV. Section V discusses the results of our official TREC test runs.

In Section III-A, we investigate the effects of query formulation using the Okapi retrieval model. Section III-B deals with our experiments on using tiers of boolean queries to match against the metadata fields in the MEDLINE records. Then, in Section III-C, we explore the idea of merging the document sets retrieved by Okapi and the query tiering techniques. We describe the use of query expansion and feedback in Section III-D. In Section IV, we assemble the techniques into complete runs.

### A. Okapi Query Formulation for the Genomics Track

Two important facts were discovered in preliminary experiments which influenced the design of the Okapi experiments. First, the gene name type did not seem to matter. A document discussing a particular gene was as likely to use an official name as an alternate one. Second, spacing and punctuation had a large effect on performance in some cases. The gene name in the original LocusLink-derived query may differ from the gene or protein name as it actually appears in the corpus only by the addition or removal of spaces or dashes. In a model based on term sets, such as Okapi, these slight variations may significantly affect the results.

We investigated the effects of query formulation on IR in the Genomics domain by generating multiple term sets from the original query, and comparing the effects of using these term sets to retrieve documents using the Okapi retrieval model. The three rules used to generate the term sets were:

- *Okapi 1*: Each gene name in the original query, which may consist of multiple alphanumeric tokens, is considered as a phrase and treated as a single term, the only change being the removal of punctuation.
- *Okapi 2*: Heuristics were used to split up gene names containing semi-colons, commas, and brackets. Heuristics were also used to guess "plurals" for some of the terms.
- *Okapi 3*: First, the gene names were separated into two sets, one containing those gene names which were comprised of a single token, and another containing gene names which were comprised of multiple tokens. The Okapi term set was created from these two sets by first concatenating all pairs of single-token gene names together, and adding all the token-bigrams from the multiple-token gene names.

The name of the species was also included in each of the term vectors. The three rules are in decreasing order of strictness. Documents retrieved by Okapi 1 will contain the terms exactly as given in the original query (ignoring punctuation), while those retrieved by Okapi 2 will contain terms which are similar to but not exactly like those in the original query. Documents retrieved by Okapi 3 contain the same bigrams as found in the original query.

Each query formulation has its own advantages and disadvantages. The top documents returned by Okapi 1 are likely to be relevant, since they contain the query exactly, but many relevant documents may be missed because the gene name in the document appears differently than in the query. On the other hand, Okapi 3 retrieves many relevant documents in which the gene name does not appear exactly as in the query. However, it also retrieves many documents that are not relevant. The documents retrieved by Okapi 2 are intermediate between the two.

We found that the document sets retrieved using the term vectors generated by the three rules were quite different. Therefore, it was decided that the document sets produced by Okapi 1, Okapi 2, and Okapi 3 would be fused together. The fusion was accomplished in the following manner:

- *Okapi Fusion*: The document sets retrieved by Okapi 1, Okapi 2, and Okapi 3 are combined by taking the intersection of the three result sets. A document's score is taken to be the product of the three scores. This list is then followed by the remainder of Okapi 3, with the scores appropriately scaled.

The rationale behind the fusion is that a document that scores highly on all three query formulations is very likely to be relevant. Taking the product of the scores allows each of the three document sets to vote on the relative distance between similarity values equally. Since Okapi 3 is the most relaxed of the three query formulations, it retrieves most if not all of the relevant documents retrieved by Okapi 1 and 2. Thus, the intersection of the three document sets likely contains most of the relevant documents in the document sets returned by Okapi 1 and 2, while it might miss relevant documents retrieved by Okapi 3. For that reason, the remainder of the Okapi 3 document set is appended to the end of the combined list.

While there are other fusion techniques, the above seemed to work very well in preliminary trials, and thus was the only technique used in the final completed runs. The performance of Okapi 1 is considered to be the baseline for comparison purposes in the rest of this report.

### B. Boolean Query Formulation for the Genomics Track

Preliminary experiments showed that there was a correlation between some of the metadata fields in the MEDLINE record and the relevance of the document. In particular, there was a strong correspondence between the query terms and the terms that appeared in the RN (registry number) field of the MEDLINE record. The RN field contains a list of the chemicals discussed in the document. Many of these chemical names can be matched to the gene names found in the query. The chemical list is a better indicator of a document's relevance than the document's title, which in turn is a better indicator than the abstract. To capture this hierarchical structure among the metadata fields, we experimented with using a number of query tiers. The final tiering system had the following tiers, in decreasing order of relevance:

1) *Tier 1*: The gene name is found in the chemical list, or it is found in the chemical list preceded or followed by the word "protein", optionally followed by the name or description of the species. Spaces and punctuation are ignored for the purposes of comparison. (From training topic 5, "glycine receptor, alpha 1" is considered to be equivalent to "glycine receptor alpha1".)
2) *Tier 2*: This tier is similar to Tier 1, except that the chemical name is allowed to have additional terms. (From training topic 11, "RAC1" retrieves documents in which "rac1 GTP-Binding Protein" appears in the chemical list.)
3) *Tier 3*: An attempt is made to find the conjunction of the terms from the gene name in the chemical list. If the gene name consists of a class name followed by a sequence of letters and numbers that specifies an object of that class, the name is successively weakened until a match is made. A set of heuristics are also used to recognize plurals. (From training topic 32, "estrogen receptor 1" is weakened until the documents retrieved contain "Receptors, Estrogen" in the chemical list.)
4) *Tier 4*: The query is converted into a boolean expression by turning each gene name into the conjunction of its terms, and taking the disjunction of all gene names. The boolean expression is applied to the title.
5) *Tier 5*: The boolean expression is applied to the chemical list.
6) *Tier 6*: The boolean expression is applied to the abstract.

In addition, the documents are restricted to those in which the name of the species appears in the MeSH (Medical Subject Heading) metadata field. This does not completely eliminate documents which are not relevant to the species, since it is possible for the name of the species to appear in the MeSH field even if the focus of the paper is another species. It is quite common for an article about a gene in one species to mention a homologue in a related species. Nevertheless, if the name of the wanted species does not appear in the MeSH heading, then the article is (almost certainly) not relevant. Thus, using species data in the MeSH metadata field may result in false positives but not (or rarely) in false negatives.

Based on the query tiering model described above, we tested three different ways of retrieving documents:

- *All Tiers*: Retrieve documents from all the tiers. Documents retrieved by each tier are ranked ahead of all documents retrieved by the next tier. A document that is retrieved in more than one tier is counted towards only its highest tier.
- *Best Tier*: Retrieve the documents in the first tier that contains a non-zero number of documents. Subsequent tiers are ignored.
- *Exact*: Retrieve only documents in Tier 1. No documents are retrieved if there are no documents in Tier 1.

Note that for some topics, the above techniques may return zero documents. For that reason, the complete runs described in Section IV supplement the document sets retrieved by the tiering techniques with documents retrieved using the Okapi methods.

While the query tiers have a significant effect on performance, further improvement is possible by using fusion and feedback.

### C. Genomics Track Fusion

Since the Okapi and tiering experiments retrieved different document sets, we explored merging the results of the two techniques. We tried two different methods of combining the two document sets returned from Okapi and the tiering technique:

- *Interweave*: The two document sets are combined by taking one document from each set successively.
- *Rank Fusion*: First, documents which were retrieved by both methods are merged. The score assigned to a document is a weighted sum of its (reverse) rank in each document set. The combined documents are followed by interweaving the remainder of the two document sets.

We also attempted other types of fusion, but these were the only two which were completely implemented and tested due to time constraints.

### D. Genomics Track Feedback

As explained in Subsection III-B, the similarity of the chemical list in the MEDLINE record to the query is a good indicator of a document's relevance. Because a gene name may have many variants, however, it is not always possible to match the gene name to an item in the chemical list even though one of the chemicals may refer to that gene or its product.

One possible solution to this problem is to attempt to recognize these name variants. That is not the approach we took. Instead, we attempted to learn the variant name by using feedback. If the gene name was matched in Tier 1 using the tiering technique, then the chemical list in the top retrieved documents already contains the gene name, and so feedback is unnecessary. Otherwise, we assume the top documents retrieved to be relevant, and find the chemical that has the highest correlation with these documents. The chemical names in the top documents were assigned a score using the formula:

$$w_i = R_i \cdot \left(\log\left(\frac{N}{f_i}\right)\right)^{\alpha}$$

For a chemical $i$, $R_i$ is the number of times the chemical name appears in the chemical list of the top documents, $f_i$ is the number of times it appears in the corpus, $N$ is the total length of all documents in the corpus, and $w_i$ is the score assigned to $i$. For our TREC 2003 experiments, we set $\alpha = 3$.

The highest scoring chemical name is then used to retrieve a set of documents containing that chemical name. The three sets of documents retrieved by the Okapi Fusion, query tiers, and feedback are then merged to produce the final document set. The number of top documents assumed to be relevant and the precise mechanism used to merge the final document sets are discussed in the next subsection.

| Method Used | Rel. & Ret. | Avg. Precision | R-Precision |
|---|---|---|---|
| Okapi 1 | 224 | 0.3273 | 0.3077 |
| Okapi 2 | 245 | 0.3193 | 0.2917 |
| Okapi 3 | 261 | 0.3157 | 0.2700 |
| Okapi Fusion | 261 | 0.3321 | 0.3173 |
| AT | 282 | 0.3819 | 0.3452 |
| ATI | 282 | 0.4394 | 0.3836 |
| ATIF | 289 | 0.4429 | 0.3844 |
| ATR | 284 | 0.4519 | 0.4324 |
| **ATRF** | **291** | 0.4598 | 0.4448 |
| BT | 279 | 0.4003 | 0.3818 |
| BTI | 279 | 0.4528 | 0.4236 |
| BTIF | 286 | 0.4812 | 0.4448 |
| BTR | 279 | 0.4452 | 0.4216 |
| **BTRF** | 286 | **0.4821** | **0.4579** |
| Exact | 277 | 0.3981 | 0.3820 |
| ExactI | 277 | 0.4246 | 0.3959 |

TABLE I

GENOMIC TRACK — SUMMARY OF RESULTS ON TRAINING DATA: 50 TOPICS, 1000 RETRIEVED PER QUERY, 335 TOTAL RELEVANT.

## IV. GENOMICS TRACK - TRAINING RUNS

The parameters of the various runs were optimized for the training data, using the supplied relevance judgments. Thus, the performance of the IR system on the training data is not necessarily reflective of its performance on the test data, especially if the training and test data have different characteristics. In particular, the relative performance of some of the runs that relied on a single retrieval technique may not be necessarily preserved. Nevertheless, the runs involving fusion and feedback do seem to consistently outperform the systems on which they are based. The parameters for these runs were adjusted not only to maximize performance, but to increase stability as well.

Following the TREC standard procedure, 1000 documents were retrieved for each run. We attempted to test a large

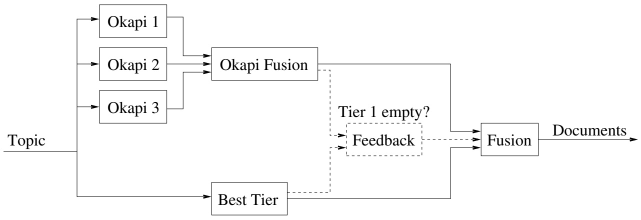


Fig. 1. Flow diagram for the combined system of the BTRF run.

variety of techniques, but unfortunately many tests could not be completed due to time constraints. The results for the runs which we conducted on the training data are shown in Table I. These were:

- *Okapi 1, 2, 3, and Fusion*: These are the document sets retrieved by the procedure described in Section III-A.
- *All Tiers* (AT): This is the set of documents retrieved by using the All Tiers method as described in Section III-B. The documents retrieved by Okapi Fusion are appended to the end.
- *All Tiers Interweave-fusion* (ATI): The set of documents retrieved by All Tiers is interweaved with the document set retrieved by Okapi Fusion.
- *All Tiers Rank-fusion* (ATR): The set of documents retrieved by All Tiers is merged with the Okapi Fusion documents using the weighted rank fusion. It was experimentally determined that good results can be obtained if the Okapi rank was weighted 4 times as heavily as the tiering rank.
- *All Tiers Interweave/Rank-fusion with Feedback* (ATIF, ATRF): These are the same as ATI and ATR, respectively, except that the feedback procedure described in Section III-D and further elaborated below was used if no documents were retrieved in Tier 1.
- *Best Tier* (BT, BTI, BTR, BTIF, BTRF): These are analogous to the above, except that the query tiering subsystem retrieved only documents from the first tier with non-zero documents.
- *Exact*: Instead of all the tiers or the best tier, only Tier 1 was used to retrieve documents. The Okapi Fusion document set was then appended to the end. (If no documents were retrieved in Tier 1 for a topic, then the final set of retrieved documents is just the set retrieved by Okapi Fusion.)
- *ExactI*: The set of documents retrieved by Tier 1 is interweaved with the Okapi Fusion set.

Figure 1 shows the combined system for the BTRF (Best Tier, Rank-fusion, Feedback) run. The topic is sent to both the Okapi and query tiers subsystems, each of which returns a set of documents. If the first tier to retrieve a non-zero number of documents is Tier 1, then the two document sets are fused. Otherwise, a third set of documents is retrieved using feedback, and the three sets of documents are fused. The other runs follow a similar logic flow.

The performance of feedback is dependent on the number of top documents used to determine the most relevant chemical name, and on the type of fusion used to merge the three document sets. These parameters are in turn dependent upon the query tiering technique used. For the All Tiers technique, it was determined that using the top 25–30 documents to determine the most relevant chemical name produced the best performance. (The value of 27 was used in the experiments.) The three document sets are fused using rank fusion with equal weights. For the Best Tier technique, the top 42 documents were used, and the three document sets were merged using weighted rank fusion with a weight of 5 for the query tiers document set, 28 for the feedback document set, and 20 for the Okapi Fusion document set. These numbers were determined experimentally.

The reason for the difference between the feedback parameters of the AT and BT runs is that more of the top documents retrieved by the Best Tier technique are relevant compared to those retrieved by All Tiers. Since feedback is only used when no documents are retrieved in Tier 1, the set of documents retrieved using the top chemical name will be far more likely to be relevant than the documents retrieved by the Best Tier, and slightly more likely to be relevant than those retrieved by Okapi.

As can be seen from Table I, the best average precision belonged to the BTRF run, at 0.4821. This is a 47.3% improvement over the baseline Okapi 1, which had an average precision of 0.3273. The BTIF run had an average precision of 0.4812, a 47.0% improvement, and the ATRF run had an average precision of 0.4598, a 40.5% improvement. The ATRF run retrieved 291 relevant documents, which was the most relevant documents retrieved of all the runs. This is slightly more than the 286 retrieved by BTRF and BTIF, and significantly more than the 224 retrieved by the Okapi 1 run.

Some general trends are discernible from the numbers. Feedback and fusion improved performance in every case, and the systems with the best performance made use of both. It isn't clear which fusion method is better, since ATR outperformed ATI, but BTI did better than BTR. However, when fusion is used with feedback, the rank fusion method outperformed the interweave fusion method in both cases.

There is a high level of correspondence between the metadata fields and the relevance of the documents. This is clear from the fact that retrieval using query tiers based on the information in the metadata fields outperformed the Okapi runs, including the Okapi Fusion run. Before fusion and feedback, the best technique that is based on query tiers is BT, with an average precision of 0.4003, which is a 22% improvement over Okapi 1. The Exact run had an average precision of 0.3981, a 21% improvement, while the AT run had an average precision of 0.3819, which close to 17% over Okapi 1. Note that both Best Tier and Exact had a better average precision than the All Tiers method. It appears that once a match has been found in a tier, it was a better strategy to append the Okapi Fusion list rather than documents from lower tiers. The experimental results suggest that the performance of the Okapi Fusion method was between that of Tier 1 and 2.

| Topic | Number of Documents Retrieved | | | | | | Matches in Best Tier |
|---|---|---|---|---|---|---|---|
| | T1 | T2 | T3 | T4 | T5 | T6 | |
| 1 | **438** | 120 | 0 | 19 | 0 | 482 | “cip1 protein” |
| 2 | **6** | 13 | 38 | 4 | 0 | 28 | “rna dependent atpase”, “protein p68” |
| 3 | **19** | 31 | 0 | 5 | 0 | 43 | “tel protein” |
| 4 | **35** | 2 | 499 | 2 | 0 | 75 | “keratinocyte growth factor”, “fibroblast growth factor 7 precursor”, “fibroblast growth factor 7” |
| 5 | **16** | 0 | 23 | 0 | 0 | 6 | “glycine receptor alpha1” |
| 6 | **93** | 10 | 0 | 2 | 0 | 101 | “hla dqb1” |
| 7 | **56** | 3 | 44 | 0 | 0 | 39 | “janus kinase 2” |
| 8 | – | – | – | **8** | 0 | 50 | ((“luteinizing”^“hormone”^“choriogonadotropin”^“receptor”)+“lhcgr”+“lcgr”+“lhr”+ (“luteinizing”^“hormone”^“receptor”)+(“lutropin”^“choriogonadotropin”^“receptor”)+“lcgrs”+ “lhcgrs”+(“luteinizing”^“choriogonadotropin”^“receptor”)+“lgr2”+“lhrs”+(“lutropin”^“receptor”)+ (“choriogonadotropin”^“receptor”)) |
| 9 | **15** | 1 | 68 | 12 | 0 | 345 | “growth inhibitory factor” |
| 10 | **161** | 360 | 757 | 480 | 0 | 785 | “protein c” |
| 11 | – | **80** | 0 | 0 | 0 | 117 | “rac1” |
| 12 | **3** | 0 | 41 | 0 | 0 | 11 | “tropomyosin 1” |
| 13 | **3** | 0 | 3 | 7 | 0 | 163 | “gpcr protein”, “frizzled 4 protein vertebrate” |
| 14 | – | – | – | **10** | 0 | 408 | ((“tyrosyl”^“trna”^“synthetase”)+“tyrrses”+“ytses”+“yts”+ (“tyrosyl”^“trna”^“ligase”)+“yars”+“tyrrs”+“yarses”+“yrses”+“yrs”) |
| 15 | **11** | 1 | 0 | 13 | 0 | 109 | “major vault protein” |
| 16 | **4** | 0 | 80 | 0 | 0 | 0 | “adrenergic receptor alpha 1d”, “adrenergic receptor alpha 1a” |
| 17 | – | **10** | 0 | 0 | 0 | 0 | “rhob” |
| 18 | **213** | 0 | 205 | 2 | 0 | 73 | “cpp32 protein” |
| 19 | **6** | 0 | 0 | 0 | 0 | 6 | “ctcf protein” |
| 20 | **162** | 0 | 979 | 2 | 0 | 68 | “fasl protein” |
| 21 | – | – | **1** | 2 | 0 | 44 | (((“ig”))) |
| 22 | – | – | – | **4** | 0 | 14 | (“ihhs”+(“indian”^“hedgehog”)+“ihh”) |
| 23 | – | – | **47** | 1 | 0 | 16 | (((“phospholipase”+“phospholipases”))^“c gamma”) |
| 24 | – | – | **3** | 0 | 0 | 0 | (((“seven”+“sevens”)^(“absentia”+“absentias”))) |
| 25 | – | – | – | **3** | 0 | 112 | (“dntts”+“tdt”+“dntt”+(“terminal”^“deoxynucleotidyl”^“transferase”)+ (“deoxynucleotidyltransferase”^“terminal”)+“tdts”) |
| 26 | – | – | – | **1** | 0 | 1 | ((“rho”^“related”^“btb”^“domain”^“containing”^“2”)+“rhobtb2”+“kiaa0717”+“dbc2”) |
| 27 | – | – | – | – | – | **19** | ((“cholinergic”^“receptor”^“muscarinic”^“3”)+“chrm3”) |
| 28 | – | **11** | 0 | 9 | 0 | 57 | “egr1”, “ngfi” |
| 29 | **19** | 1 | 0 | 0 | 0 | 8 | “glucokinase” |
| 30 | **2** | 0 | 40 | 0 | 0 | 1 | “retinoic acid receptor gamma” |
| 31 | **149** | 4 | 460 | 9 | 0 | 93 | “neurokinin a”, “substance p”, “neuropeptide k” |
| 32 | – | – | **186** | 4 | 0 | 75 | (((“estrogen”+“estrogens”)^(“receptor”+“receptors”))) |
| 33 | – | – | **70** | 0 | 0 | 21 | (((“guanylate”+“guanylates”)^(“cyclase”+“cyclases”))) |
| 34 | **20** | 1 | 0 | 0 | 0 | 2 | “cocaine and amphetamine regulated transcript protein” |
| 35 | – | – | – | – | – | – | – |
| 36 | **5** | 0 | 9 | 2 | 0 | 6 | “hop protein” |
| 37 | **1** | 0 | 0 | 0 | 0 | 1 | “slob protein” |
| 38 | **3** | 0 | 0 | 0 | 0 | 0 | “eiger protein drosophila” |
| 39 | **32** | 1 | 7 | 1 | 0 | 15 | “cadherins” |
| 40 | **6** | 0 | 0 | 3 | 0 | 2 | “stat92e protein” |
| 41 | **3** | 0 | 0 | 0 | 0 | 3 | “ebony protein” |
| 42 | **10** | 0 | 0 | 0 | 0 | 5 | “crb protein drosophila” |
| 43 | – | – | **3** | 11 | 0 | 422 | (((“calcineurin”+“calcineurins”))) |
| 44 | **3** | 0 | 4 | 0 | 0 | 0 | “gp73 protein” |
| 45 | **5** | 1 | 3 | 2 | 0 | 5 | “sh3px1 protein”, “wisp protein” |
| 46 | – | **7** | 0 | 5 | 0 | 16 | “hanks”, “ank” |
| 47 | **2** | 0 | 0 | 0 | 0 | 0 | “dda3 protein” |
| 48 | **10** | 0 | 0 | 10 | 0 | 323 | “artemis protein human” |
| 49 | – | – | **1000** | 67 | 0 | 947 | (((“transcription”+“transcriptions”)^(“factor”+“factors”))) |
| 50 | **1** | 0 | 2 | 0 | 0 | 1 | “pax 8 protein” |
| Total | 32 | 4 | 8 | 4 | 0 | 1 | |

TABLE II

GENOMIC TRACK — MATCHES IN THE QUERY TIERS FOR THE TRAINING TOPICS.

| Topic | Query Term/Phrase | Feedback Chem. Name | Ret. | R.&R. | MAP | R-P | MAP Fb. | R-P Fb. | Imp. |
|---|---|---|---|---|---|---|---|---|---|
| 8 | luteinizing hormone/ choriogonadotropin receptor | Receptors, LH | 49 | 7 | 0.2917 | 0.4286 | 0.4305 | 0.4286 | +47% |
| 11 | ras-related C3 botulinum toxin substrate 1 (rho family, small GTP binding protein Rac1) | rac1 GTP-Binding Protein | 80 | 13 | 0.2302 | 0.4118 | 0.1977 | 0.1765 | -14% |
| 14 | tyrosyl-rRNA synthetase | Tyrosine-rRNA Ligase | 10 | 6 | 0.5872 | 0.5000 | 0.8238 | 0.6667 | +40% |
| 17 | ras homolog B (RhoB) | rhoB GTP-Binding Protein | 6 | 2 | 0.3333 | 0.3333 | 0.3889 | 0.6667 | +17% |
| 21 | immunoglobulin heavy chain 6 (heavy chain of IgM) | Immunoglobulins, mu-Chain | 21 | 0 | – | – | – | – | – |
| 22 | Indian hedgehog | hedgehog protein, vertebrate | 69 | 6 | 0.4703 | 0.5000 | 0.6723 | 0.5000 | +43% |
| 23 | phospholipase C, gamma 1 | phospholipase C gamma | 47 | 9 | 0.6503 | 0.5556 | 0.5262 | 0.4444 | -19% |
| 24 | seven in absentia 2 | seven in absentia protein | 3 | 2 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0% |
| 25 | terminal deoxynucleotidyl transferase | DNA Nucleotidylexotransferase | 8 | 2 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0% |
| 26 | Rho-related BTB domain containing 2 | QM protein, Trypanosoma brucei | 0 | 0 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 0% |
| 27 | cholinergic receptor, muscarinic 3 | Receptors, Muscarinic | 153 | 2 | 0.0312 | 0.0000 | 0.0747 | 0.0000 | +139% |
| 28 | Early growth response 1 | Krox-24 protein | 40 | 8 | 0.0258 | 0.1250 | 0.2523 | 0.1250 | +878% |
| 32 | estrogen receptor 1 | Receptors, Estrogen | 163 | 11 | 0.1039 | 0.0909 | 0.1354 | 0.0000 | +30% |
| 33 | guanylate cyclase 1, soluble, beta 3 | Guanylate Cyclase | 70 | 1 | 0.0774 | 0.0000 | 0.0569 | 0.0000 | -26% |
| 35 | CG3599 | Drosophila Proteins | 638 | 0 | – | – | – | – | – |
| 43 | Calcineurin B | Calcineurin | 3 | 1 | 0.5000 | 0.0000 | 1.0000 | 1.0000 | +100% |
| 46 | ankylosis, progressive homolog | ankylosis protein | 5 | 3 | 0.1595 | 0.0000 | 0.7500 | 0.7500 | +370% |
| 49 | transcription factor 23 | Transcription Factors | 1000 | 0 | – | – | – | – | – |

TABLE III

GENOMIC TRACK — ANALYSIS OF THE EFFECTS OF FEEDBACK ON PERFORMANCE.

Table II shows the documents retrieved in each tier for the 50 training topics. The topic number is shown in the first column, followed by six columns showing the number of documents retrieved in each of the six tiers. The last column contains the expression or expressions used in the first tier in which a match was made.

In 32 out of 50 topics, the best tier was Tier 1. Of the remaining topics, Tier 2 was the best tier in 4 topics, Tier 3 was best in 8, and Tier 4 was best in 4. No documents were retrieved at all in Tier 5, and Tier 6 was the best tier for 1 topic. In the final arrangement of the query tiers, it happened that every document retrieved by Tier 5 had already been retrieved in a higher tier.

Because Tier 1 had a better performance on its own than Okapi or even feedback, performance can be improved by recognizing relevant chemical names in the chemical list metadata, even in cases where the name of the gene and the relevant chemical name are different.

Table III shows the chemical names produced by the pseudo-relevance feedback for those topics in which no documents were retrieved in Tier 1, for the BTRF run. The first column gives the topic number, and the second column gives a gene name from the query. The third column shows the chemical name that was found using automatic query expansion. The next four columns show the number of documents retrieved, the number retrieved and relevant, the mean average precision, and the interpolated recall-precision, respectively, for that topic without using feedback. The next two columns give the mean average precision and interpolated recall-precision with feedback, and the last column gives the percentage improvement (or degradation) due to using feedback. It is apparent that most of the chemical names are related in some way to the gene name, and a better way of recognizing the relationship between a gene and a chemical name will clearly improve performance.

For topic 28, the top chemical name "Krox-24 protein" was produced for the "Early growth response 1". In fact, "Krox-24 protein" is another name for "Early growth response 1". By searching on "Krox-24 protein", which does not appear in the original query, the average precision was improved by an incredible 878%. Of course, the original performance for this topic was very poor, but there is clearly a lot of potential for improving performance by recognizing the alternate names of a gene or a substance related to a gene.

In some cases, this is relatively simple. For topic 14, for example, the chemical name "Tyrosine-rRNA Ligase" was generated for the gene name "tyrosyl-rRNA synthetase". A system that understood the relationship between "tyrosine" and "tyrosyl" and "ligase" and "synthetase" can determine that the two expressions refer to the same thing (or closely related things), and even assign a score for the degree of similarity. In other cases, this is complicated by the fact that more than one chemical name generated by the automatic expansion might be relevant to the query. For topic 27, searching on the gene name "cholinergic receptor, muscarinic 3" resulted in the top chemical name "Receptors, Muscarinic". However, the chemical name "muscarinic receptor M3", which is clearly more relevant, was overlooked. Choosing this chemical name instead of the more general "Receptors, Muscarinic" would have resulted in an improvement of 534%.

As the table shows, in most cases the performance was improved by using feedback to find the most relevant chemical, though in some cases there was a degradation in performance. Determining the conditions under which feedback improved or

| Method Used | Rel. & Ret. | Avg. Precision | R-Precision |
|---|---|---|---|
| Okapi 1 | 447 | 0.2060 | 0.1965 |
| Okapi 2 | 473 | 0.2155 | 0.1948 |
| Okapi 3 | 524 | 0.2169 | 0.2095 |
| Okapi Fusion | 524 | 0.2323 | 0.2138 |
| AT | 550 | 0.2542 | 0.1967 |
| ATI | 550 | 0.3334 | 0.2723 |
| ATIF | 559 | 0.3379 | 0.2680 |
| ATR | 552 | 0.3425 | 0.3050 |
| **ATRF** | **562** | 0.3479 | 0.3013 |
| BT | 535 | 0.2443 | 0.2010 |
| BTI | 535 | 0.3066 | 0.2581 |
| BTIF | 556 | 0.3322 | 0.2745 |
| BTR | 535 | 0.3161 | 0.2852 |
| **BTRF** | 556 | **0.3534** | **0.3113** |
| Exact | 528 | 0.2500 | 0.2194 |
| ExactI | 528 | 0.2803 | 0.2449 |

TABLE IV

GENOMIC TRACK — SUMMARY OF RESULTS ON TEST DATA: 50 TOPICS, 1000 RETRIEVED PER QUERY, 566 TOTAL RELEVANT.

degraded performance would allow feedback to be used more effectively.

The two runs chosen for official submission to TREC were the ATRF and BTRF runs. Even though BTIF had a better mean average precision than ATRF, it was too similar to the BTRF run in that it differed only in the fusion method used. It was found that by adjusting the fusion weights, it was always possible for the rank-fusion to outperform the interweave fusion. It was also suspected that the ATRF run might be more stable, in the sense that the performance would not be too adversely affected by an incorrect match in Tier 1. ATRF also had the most number of relevant documents retrieved, and it would be interesting to examine the trade-off between retrieving more relevant documents and having a better precision. In addition to the two official runs, we also performed all the various runs using the test data.

## V. GENOMICS TRACK - RESULTS

The results for various test runs of our system are shown in Table IV. ATRF and BTRF are official runs, submitted to NIST under the run tags "`uwmtg03atrf`" and "`uwmtg03btrf`".

Some similarities and differences between the training and test results may be noted. As with the training data, the BTRF run had the best performance on the test data, with an average precision of 0.3534. This is a 71.5% improvement over the Okapi 1 run, which had an average precision of 0.2060. The ATRF run retrieved the most relevant documents, and had the second best average precision at 0.3479, a 68.9% improvement over the Okapi 1 run. Furthermore, ATRF performed better than BTIF, which had an average precision of 0.3322. The distance between ATRF and BTRF was also smaller. On the training data, BTRF had a 4.8% improvement in average precision over ATRF, but on the test data that difference is only 1.6%. This suggests that with the test data, the gene names in the corpus are less like the queries than with the training data.

This conjecture is also supported by the performance of the Okapi runs. While the Okapi Fusion run performed better than any individual Okapi run, the Okapi 3 run had the highest average precision, followed by Okapi 2, and then Okapi 1. This is the reverse of the order with the training data. Using bigrams rather than the original query resulted in better performance on the test data. A more thorough analysis is needed to determine if this conjecture is correct.

## VI. ROBUST RETRIEVAL TRACK

For the Robust Track, MultiText examined the impact of pseudo-relevance feedback on retrieval effectiveness under the new robustness measures. There are two unusual aspects to our work on this track: 1) the adaptation of techniques from our question answering system to pseudo-relevance feedback, and 2) the expansion of the corpus with a terabyte of Web data for pseudo-relevance feedback. Previous applications of this "collection enrichment" technique have generally used much smaller corpora [6], [9].

### A. Robust Track Feedback

For feedback, we adapted the passage-retrieval and term-extraction methods from our QA system, which we have been developing over the past four years. Query processing proceeds as follows:

1) After stopword elimination and stemming, the terms from the topic field(s) are used by the Qap passage-retrieval algorithm (Section II-B) to locate the top $m$ hotspots.
2) Feedback terms are extracted from the hotspots and the text surrounding them. A score is computed for each extracted term. If two terms stem to the same root, the term with the lowest score is eliminated, since stemming will be applied to the expanded query.
3) The top $k$ feedback terms are added to the original term set.
4) Terms in the expanded query are stemmed. The result is executed using our implementation of Okapi BM25 to return the top 1000 documents.

We treat any non-query term appearing within or near a hotspot as a candidate for feedback. For most of our TREC 2003 runs we extracted only single-word terms for feedback; for one run we we also extracted word bigrams. Our term extraction method assigns a score to each candidate term based on its distance from the hotspot, the number of retrieved passages it which it appears, and its relative frequency within the corpus.

Let $H_1, H_2, ..., H_m$ be the hotspots located by Qap. Let $l(H)$ be the length of hotspot $H$ as measured by the number of alphanumeric tokens it contains. We define a function $L(H,t)$ over hotspots $H$ and terms $t$ that measures the "length" of a passage that contains both the hotspot and the term. If $t$ appears in the hotspot, then $L(H,t) = l(H)$. If $t$ appears outside the hotspot, then $L(H,t)$ is the length in tokens of the shortest passage that contains both $t$ and the entire hotspot $H$. If $t$ does not appear in proximity to the hotspot — if it

| | | Old Topics | | | New Topics | | | Old + New Topics | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Run Tag | Run Type | avgp | norel | bad | avgp | norel | bad | avgp | norel | bad |
| `uwmtCR0` | description only, feedback | 0.150 | 14.0% | 0.011 | 0.403 | 8.0% | 0.052 | 0.276 | 11.0% | 0.018 |
| `uwmtCR1` | description only, no feedback | 0.114 | 18.0% | 0.009 | 0.355 | 6.0% | 0.035 | 0.234 | 12.0% | 0.013 |
| `uwmtCR2` | title only, feedback | 0.168 | 22.0% | 0.006 | 0.370 | 10.0% | 0.053 | 0.269 | 16.0% | 0.015 |
| `uwmtCR3` | title only, no feedback | 0.102 | 16.0% | 0.007 | 0.285 | 8.0% | 0.042 | 0.194 | 12.0% | 0.013 |
| `uwmtCR4` | description only, feedback, bigrams | 0.148 | 20.0% | 0.014 | 0.404 | 8.0% | 0.054 | 0.274 | 14.0% | 0.019 |
| - | title + description, feedback | 0.175 | 16.0% | 0.017 | 0.408 | 8.0% | 0.087 | 0.292 | 12.0% | 0.029 |
| - | title + description, no feedback | 0.133 | 10.0% | 0.018 | 0.369 | 2.0% | 0.066 | 0.251 | 6.0% | 0.026 |
| - | combMNZ (uwmtCR0, uwmtCR2) | 0.174 | 10.0% | 0.020 | 0.411 | 4.0% | 0.085 | 0.292 | 7.0% | 0.033 |
| - | combMNZ (uwmtCR1, uwmtCR3) | 0.130 | 10.0% | 0.016 | 0.360 | 2.0% | 0.066 | 0.245 | 6.0% | 0.024 |

TABLE V
ROBUST TRACK — SUMMARY OF RESULTS

appears outside a large window surrounding the hotspot or if it does not appear inside the document containing the hotspot — then for simplicity we define $L(H,t) = N/f_t$, where $N$ is the total length of all documents in the corpus and $f_t$ is the total number of times $t$ appears in the corpus. We then compute the feedback score for term $t$ as:

$$w_t = \sum_{1 \le i \le m} \log\left(\frac{N}{f_t \cdot L(H_i, t)}\right) \tag{5}$$

We generate an expanded query by combining the top $k$ feedback terms with the original topic terms. We adjust the retrieval weights of the added terms with a scaling factor that takes the feedback score $w_t$ into account, and reflects the fact that terms added through feedback should not be assigned the same importance as the original topic terms.

Let $W$ be the score of the top-ranking feedback term (i.e. the term with the largest feedback score). We define the scaling factor for feedback term $t$ as:

$$S_t = \frac{C \cdot w_t}{W} \tag{6}$$

where $C = 1/3$ in all our experiments. $S_t$ is used to adjust the retrieval weights in the Okapi BM25 formula, modifying Equation 4 to:

$$\sum_{t \in Q} S_t w^{(1)} q_t \frac{(k_1+1)d_t}{K + d_t} \tag{7}$$

For original topic terms $S_t = 1$; for feedback terms $S_t \le 1/3$.

As an example, we examine the end-to-end processing for topic 613, using the description field:

> How were pieces of the Berlin wall disposed of after their removal?

After stopword removal, the Qap algorithm is used to generate a set of passages. A typical passage returned by Qap is:

> Edwina Sandys, whose sculptures are installed at five United Nations centers around the world. One of Winston Churchills 10 grandchildren, her sculpture Breakthrough made of *Berlin Wall pieces* has been called one of the most important monuments constructed on American soil since the Vietnam War Memorial.

The hotspot is in italics. Feedback over these passages generates the expanded term set:

> #1.00 pieces #1.00 berlin #1.00 wall #1.00 disposed #1.00 removal #0.333 war #0.297 freedom #0.293 cold #0.286 dismantling #0.236 souvenirs #0.195 display #0.137 selling #0.110 gift ...

Scaling factors ($S_t$) precede each term. The average precision for the topic increases from 0.212 to 0.487.

### B. Robust Track Experiments

In preparation for the track, we generated a large number of training runs and examined the impact of feedback parameters (e.g. the number of added terms) on average precision and robustness. In general, as feedback parameters are changed, average precision changes slowly in the direction of a single local maxima over a wide range of parameters. In contrast, changes to feedback parameters have small but unpredictable effects on the robustness measures, with many local maxima, making it difficult to tune these parameters specifically for robustness. As a result of our preliminary experiments we used a single set of parameters for all experiments: $m$ = 20 passages and $k = 35$ terms.

Using our passage-retrieval algorithm, we executed the original queries against three collections: 1) the AQUAINT corpus, used for the QA Track; 2) a terabyte collection of Web data; and 3) TREC disks 4 and 5 minus the CR documents, which is the target collection for the Robust Track. We then retrieved the top 20 passages from each collection, extracted the top 300 terms from each set of passages, and merged these terms into a single ranked list. A term was only included in this list if it appeared in list of terms extracted from the target collection. Finally, we added the top 35 terms from the merged list to the original query and executed this expanded query against the target collection using our version of Okapi BM25.

Table V provides a summary of our Robust track results. Each line provides results for a single run over various topic sets. The "Old Topics" were taken from the adhoc tasks of previous TREC evaluations; the "New Topics" were created for TREC 2003. For each set of topics the table reports values for the three measures used in the Robust track: 1) average precision ("avgp"), 2) the percentage of topics with no relevant documents in the top ten ("norel"), 3) and the mean average precision over the 25% of the topic set on which the run exhibited its worst performance ("bad").

The first five lines give the results for our five official runs: The pair `uwmtCR0` and `uwmtCR1` are description-only runs; the pair `uwmtCR2` and `uwmtCR3` are title-only runs. For the fifth run (`uwmtCR4`), we extended the feedback process to extract both single words and word bigrams. The last four lines give results for unofficial runs that use the topic title and description. For the sixth and seventh runs, the title and description were merged into a combined query and evaluated using the procedure above. For the last pair of runs, we fused our official title-only and description-only runs with the CombMNZ algorithm [5], [7].

As expected, feedback has a substantial positive impact on average precision. The impact is greatest on the old topics, were average precision increases by 31-65%. Over the new topics, performance improves by 11-30%. Overall, feedback has a positive impact on the "bad" robustness measure, but unfortunately it often has a negative impact on the "norel" measure.

## VII. Question Answering Track

While we continue to develop our QA system, we did not submit runs for the main task of the QA track. The number of new requirements for the task, and our interest in other tasks, precluded our full participation.

We did submit runs for the passage retrieval task. These runs combine aspects of our TREC 2001 and 2002 QA systems. Using our TREC 2002 system [2], we extracted exact answers from passages returned by the Qap algorithm. We then used techniques from our TREC 2001 system to locate 250-byte fragments that contain both the exact answers and related terms.

## VIII. Conclusion

The MultiText system supports a variety of standard and non-standard IR techniques. Depending on the track, we have combined these techniques in different ways to produce competitive runs. For the Genomics Track we merged the results of structured (boolean) and unstructured (term set) queries. Queries were expanded by re-writing terms and through feedback over the chemical names contained in metadata. For the Robust Track, we utilized a new pseudo-relevance feedback method developed from our existing QA system. Queries were expanded through feedback over an expanded collection that included a terabyte of Web data.

## References


[1] Charles L. A. Clarke and Gordon V. Cormack. Shortest substring retrieval and ranking. *ACM Transactions on Information Systems*, 18(1):44–78, January 2000.

[2] Charles L. A Clarke, Gordon V. Cormack, Graeme Kemkes, Michae l Laszlo, Thomas R. Lynam, Egidio L. Terra, and Philip L. Tilker. Statistical selection of exact answers. In *Eleventh Text REtrieval Conference (TREC-2002)*, Gaithersburg, Maryland, November 2002.

[3] Charles L. A. Clarke, Gordon V. Cormack, and Thomas R. Lynam. Exploiting redundancy in question answering. In *24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 358–365, New Orleans, September 2001.

[4] Charles L. A. Clarke, Gordon V. Cormack, Thomas R. Lynam, and Egidio L. Terra. Question answering by passage selection. In *Advances in Open Domain Question Answering*. Kluwer Academic Publishers, 2003. To appear.

[5] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Second Text REtrieval Conference (TREC-2)*, Gaithersburg, Maryland, November 1994.

[6] K. L. Kwok and M. Chan. Improving two-stage ad-hoc retrieval for short queries. In *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 250–256, Melbourne, Australia, August 1998.

[7] Joon Ho Lee. Analyses of multiple evidence combination. In *20st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–276, Philadelphia, July 1997.

[8] S. E. Robertson, S. Walker, and M. Beaulieu. Okapi at TREC-7. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, Maryland, November 1998.

[9] Amit Singhal, John Choi, Donald Hindle, David D. Lewis, and Fernando Pereira. At&t at TREC-7. In *Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, Maryland, November 1998.