# A Comparison of Techniques for Classification and Ad Hoc Retrieval of Biomedical Documents

*A. M. Cohen, J. Yang, and W.R. Hersh*
Department of Medical Informatics and Clinical Epidemiology, Oregon Health & Science University,
Portland, OR, USA

**ABSTRACT**
Oregon Health & Science University participated in both the classification and ad hoc retrieval tasks of the TREC 2005 Genomics Track. To better understand the text classification techniques that lead to improved performance, we applied a set of general purpose biomedical document classification systems to the four triage tasks, varying one system feature or text processing technique at a time. We found that our best and most consistent system consisted of a voting perceptron classifier, chi-square feature selection on full text articles, binary feature weighting, stemming and stopping, and pre-filtering based on the MeSH term *Mice*. This system approached, but did not surpass, the performance of the best TREC entry for each of the four tasks. Full text provided a substantial benefit over only title plus abstract. Other common techniques such as inverse-document frequency feature weighting, and cosine normalization were ineffective. For the ad hoc retrieval task, we used Zettair search engine. Both of our submissions used Okapi measure with the parameters optimized using the sample topics that were provided. Two different query sets were used in our runs; one with all the words and the other with only the keywords from the topic file. Queries with only keywords consistently outperformed queries with all words from the topic file. Optimization of the Okapi parameters improved our performance.

## 1 INTRODUCTION

The 2005 Text Retrieval Conference (TREC) Genomics Track was divided into two main tasks: categorization, and ad-hoc retrieval. The categorization task included four subtasks that correspond to the document triage that the curators at the Mouse Genome Institute (MGI) perform to annotate genes for alleles of mutant phenotypes, embryologic expression, GO terms, and tumor biology. The ad-hoc retrieval task was composed of 50 topics in five generic topic templates (GTTs). Each GTT had ten instances that represented biologists' information needs. Oregon Health & Science University (OHSU) participated in both categorization and the ad-hoc retrieval tasks.

## 2 CATEGORIZATION TASK

### 2.1 Background

Effective biomedical document classification can be an aid to researchers and curators. However, to provide benefit, appropriate tasks must be identified, and systems with good performance must be created. To create the most effective systems, it is important to understand what algorithmic and system features improve results and which do not. To accomplish this, good training and test collections must be available in order to build and validate the performance of effective document categorization systems.

The 2005 TREC Genomics Track had four different biomedical document classification tasks using the same document training and test corpora. The four subtasks corresponded to the four sets of document triage that the curators at the Mouse Genome Institute (MGI) perform to annotate genes for alleles of mutant phenotypes, embryologic expression, GO terms, and tumor biology. The gold standard for each of the four tasks was created using the actual annotation results of curators at the MGI. This situation of having multiple tasks using the same document set and gold standards created from the same source provided an unprecedented opportunity to study the effectiveness of common text techniques when applied to biomedical text classification.

Shared-task conferences such as TREC typically include a variety of techniques used by different submitting groups Because system implementations and features vary so much between each of the submitting groups, the contributions of individual processing features cannot be easily extracted from the results.

Our group decided to create a set of baseline systems, and then change one processing feature at a time in order to compare the contributions of the individual processing features. Having four subtasks allowed us to compare processing features across tasks, enabling us to draw some general conclusions about the effectives of various techniques commonly used in biomedical text classification.

### 2.2 System and methods

We chose our Voting Perceptron (VP) system (Freund and Schapire, 1999) from last year as our classifier algorithm (Cohen *et al.*, 2004). We have had good results with this system on several tasks in the past, and it allows a simple and effective means of accounting for the small proportion of true positives in the training and test sets.

The evaluation measure for the categorization tasks was the utility measure, computed as:

$$U_{norm} = ((u_r * TP) + (u_{nr} * FP)) / U_{max}$$

where $u_r$ is given for each task in Table 1, $u_{nr}$ is always -1, and $U_{max}$ is a constant for each task.

**Table 1.** Ur for each of the four tasks

| Task | Ur |
|---|---|
| Alleles of Mutant Phenotypes | 17 |
| Embryologic Expression | 64 |
| GO Annotation | 11 |
| Tumor Biology | 231 |

Since this measure was highly asymmetric in terms of the weights for true and false positives, tuning the classifier for this was essential to obtaining good performance. As described last year, some classification systems, such as SVMLight, do not provide adequate means to address highly asymmetric utility functions.

During our initial experiments using cross validation on the training set, we found that setting the false negative learning rate (FNLR) of the VP classifier to be equal to the ratio of the number of positives to the number of negatives in the training set gave the best performance. We used this computed learning rate for all our experiments using the VP classifier. We also tried using a publicly available rules-based classifier, Slipper (Cohen and Singer, 1999), and a single "best" feature classifier. For the latter, we gave the positive examples a weight equal to the same ratio of the number of positives to the number of negatives used for the VP classifier.

While some researchers use a full "bag of words" approach and submit all document tokens into the classifier system, we used a chi-squared based method to select features that were statistically significantly different between the positive and negative documents for each task. This significantly reduces the feature space, from tens or hundreds of thousands of features, to about 5000 features, reducing the noise introduced into the classifier, possibly improving performance when the document collection is of limited size. Chi-square feature selection also reduces classifier processing time substantially.

For the VP and Slipper classifiers studied here, the features analyzed for chi-square collection included the document text, MeSH terms, and MGI gene identifiers found in the abstract using a named entity recognition and normalization (NER+N) system we have previously described (Cohen, 2005). A 2x2 table is then constructed (according to Table 2 for each feature), and the p-value is computed using chi-square. Features with p values less than our chosen alpha of 0.05 are then used as input to the classification algorithm. As the "best" single feature classifier, we simply choose the MeSH *Mice* tag. This was shown to have a dominant effect on the GO classification task last year, and we wanted to assess how it performed on the other tasks as well.

In particular, we were especially interested in determining the value of using full text, and the best combination of algorithmic features to use with a full text representation. For each of the four classifier tasks, we varied several algorithmic features, one at a time. Table 3 shows the different algorithmic features that we varied, and how we compared them to each other.

**Table 2.** 2x2 arrangement for testing feature significance

| | | **Feature is the one under test?** | |
|---|---|---|---|
| | | **Yes** | **No** |
| **Training document is triage positive?** | **Yes** | Number of times feature seen in positive documents | Number of times other features seen in positive documents |
| | **No** | Number of times feature seen in negative documents | Number of times other features seen in negative documents |

**Table 3.** Algorithmic features studied

| **Group** | **Feature** | **Description** |
|---|---|---|
| Text | Title and Abstract | Title and abstract from Medline record. |
| | Full Text | Full text from SGML file including title, abstract, body, and captions. |
| Preprocessing | Stem & Stop | Porter Stemming and stop word list of 300 most common English words. |
| | Mice Prefilter | Only train on documents with MeSH Mice tag, classify others as negative. |
| Weighting | Binary | All features are binary, 1 or 0. |
| | IDF | Use inverse document frequency as feature weight. |
| | TF*IDF | Use standard term frequency times IDF computation as feature weight. |
| | TF*IDF, Cosine | TF*IDF with cosine normalization. |

**Table 4.** Three classifiers with baseline features and best TREC 2005 submission

| Task | Classifier | P | R | F | Un |
|---|---|---|---|---|---|
| Allele | *Mice* | 0.1315 | 0.9880 | 0.2321 | 0.6042 |
| | Slipper | 0.3448 | 0.8765 | 0.4949 | 0.7785 |
| | VP | 0.3556 | 0.8976 | 0.5094 | 0.8019 |
| | **Best** | **0.4669** | **0.9337** | **0.6225** | **0.8710** |
| Expression | *Mice* | 0.0405 | 0.9619 | 0.0777 | 0.6058 |
| | Slipper | 0.0365 | 0.9905 | 0.0705 | 0.5824 |
| | VP | 0.0693 | 0.7429 | 0.1267 | 0.5869 |
| | **Best** | **0.1899** | **0.9333** | **0.3156** | **0.8711** |
| GO | *Mice* | 0.1889 | 0.9093 | 0.3127 | 0.5542 |
| | Slipper | 0.2536 | 0.6429 | 0.3637 | 0.4709 |
| | VP | 0.2308 | 0.7819 | 0.3564 | 0.5449 |
| | **Best** | **0.2122** | **0.8861** | **0.3424** | **0.5870** |
| Tumor | *Mice* | 0.0080 | 1.0000 | 0.0159 | 0.4645 |
| | Slipper | 0.0254 | 0.9000 | 0.0493 | 0.7502 |
| | VP | 0.0237 | 0.9000 | 0.0462 | 0.7394 |
| | **Best** | **0.0709** | **1.0000** | **0.1325** | **0.9433** |

## 2.3 Results

The results of applying the three classifiers - VP, Slipper, and single feature - to the baseline feature set of title, abstract, MeSH terms, and MGI identifiers with Chi-square feature selection are shown in Table 4. This table also includes the best results submitted to TREC for each task for comparison. Notably, the best normalized utility score for the allele, expression, and tumor biology task is much better than any of our baseline results. This is not unexpected, since our results do not include using the full article text. However, the performance differences on the GO task between the MeSH term Mice, the VP classifier, and the best submission are small.

Furthermore, the performance of the VP classifier is consistently good, whereas the performance of Slipper is poorer on the allele and GO tasks as compared to the VP classifier. For the rest of the experiments presented here, the VP classifier will be used, allowing feature-by-feature analysis of the effect of individual classifier system features.

In Table 5, the baseline system is the VP classifier, with FNLR weighting as described above, and chi-square feature selection. Each entry in the table describes the other system features included in that run. The scores for the best TREC 2005 submission, as determined by utility, as well as the one feature classifier based on the MeSH term *Mice,* are included for comparison.

For the allele, expression, and tumor tasks, the highest scoring combination we investigated was consistently the baseline binary feature system plus full text, stemming and stopping, and the MeSH *Mice* term pre-filter. These runs are shown in bold. For the GO task, the baseline system using just the title and abstract, along with stemming and topping, and the *Mice* pre-filter, had the best performance.

The inclusion of full text provided a notable increase in performance for three of the four tasks. However, additional system features were required to bring out the full value of full text. The following three figures compare the performance of full text verses title and abstract using various system features on the four tasks. The height of the bar above the zero line represents the percentage improvement of full text over just title and abstract; bars below the zero line show that title and abstract outperforms full text.

In Figure 1, performance of full text versus title and abstract using no additional features is compared. While the expression task improved about 23%, the allele task was essentially unchanged, and both tumor and especially GO performed much worse with this system and full text. Figure 2 shows the results of adding Porter stemming and stop-list processing to the system. The allele task was unchanged, but the expression and tumor tasks obtained modest improvements, and the performance decrease on the GO task was reduced to one-tenth what it was in Figure 1. Figure 3 added the MeSH *Mice* prefilter to the system. The performance improvements using full text over title and abstract were larger, and the penalty on the GO task was reduced to about 3%.

The preceding figures compared the effect on utility of stemming and stopping and the pre-filter on full text verses title and abstract. Figure 4 compares all systems at once. This figure clearly shows that the combination of full text with stemming and stopping, and the Mice prefilter practically equals or outperforms any title plus abstract based system or full text system without this combination of features.

For none of the four tasks did the inclusion of IDF, TF*IDF, or cosine normalization provide any benefit.

## 2.4 Discussion

As demonstrated in our figures, especially Figure 4, stemming, stopping, and the pre-filter were effective on title plus abstract but were more effective on full text. The combination of these features resulted in a robust system that performed well on each of the four tasks, and outperformed using these features just on the title and abstract. The VP classification algorithm proved to be the most consistent and dependable for these biomedical document classification tasks.

We call the system consisting of full text, stemming and stopping, Mice pre-filter, chi-square binary feature selection and VP classification our standard system. Compared to the submitted runs the standard system would have placed 13th out of 49 for allele, 15th out of 47 for expression, 6th out of 48 for GO, and 18th out of 52 for tumor. While the system does not score above the best systems submitted for each subtask, it does perform consistently well for a generalized approach that has had no specific tuning or customization for the individual tasks. The standard system scored within 0.09 of the top scoring submitted system for all tasks: within 0.04 of the best submitted system for allele, 0.05 for expression, 0.02 for GO, and 0.09 for tumor.

Previous investigators have noted the increased amount of information available in full text verses just the title and abstract (Kostoff *et al.*, 2004). Therefore, it is perhaps not surprising that the inclusion of full text can improve performance over title plus abstract. However we are unaware of any prior research in classifying biomedical text that demonstrates this with comparative experiments such as we have done here. Furthermore, simply including full text is not enough; stemming and stopping are required to fully realize the potential. Again this conclusion was expected but has not previously been simultaneously demonstrated on a variety of biomedical document classification tasks as we have done here. The lack of prior studies demonstrating the benefit of full text may be due to the fact that full biomedical text has only recently been made available to researchers, and there are few full text training and test collections on which to do controlled experiments.

Rather unexpected was the finding that the various term weighting schemes, such as TF*IDF and cosine normalization, did not provide any benefit and in some cases reduced performance. While there are many other potential biomedical document classification tasks, the four tasks studied here consistently lead one to conclude that binary feature weighting is the best general purpose method.

Performance on the allele, expression, and tumor tasks was high enough to appear useful to the MGI curators. The standard system achieved utility measures in the 0.80 and above range for these three tasks. For the tumor tasks, some TREC 2005 submissions were able better these results, achieving utility scores above 0.90. Certainly these three tasks demonstrate that text classification can be useful for biomedical document curation and annotation. Further work is needed to determine the best way to integrate classification systems into the workflow of the MGI curators for these three tasks.

From these results it seems clear that the GO task is somewhat different from the other tasks. The best utility scores that either we or the other TREC 2005 participants were able to achieve were in the 0.50-0.60 range, which were much lower than for the other three tasks. While the data presented here do not shed light on the reasons for that difference, it seems clear that full text provided a large benefit for the three other tasks and no benefit for the GO task. Furthermore, the standard system applied to title-abstract instead of full text performed essentially identically to the top scoring system. Both of these scores are about 0.03 better than simply using the MeSH term Mice, so the classifier systems are finding some additional useful classification features, but their effect is small.

One interesting observation is the $u_r$ factor for the best performing task, tumor biology at 231, was the highest among the tasks, and lowest for the worst performing task, GO, at 11. While a high $u_r$ leads to an increasing preference for high recall over precision, a $u_r$ of 11 is still substantial compared to typical, more balanced classification tasks where the goal is often to optimize F-measure. Furthermore, the $u_r$ for the allele task, at 17 was only a little higher, but the best TREC submission scores were in the high 0.80-0.90 range, close but not quite as good as for the tumor task. Further investigation is needed to understand why the GO task appears more difficult than the other three.

**Figure 1.** Performance comparison of Title+Abstract verses full text

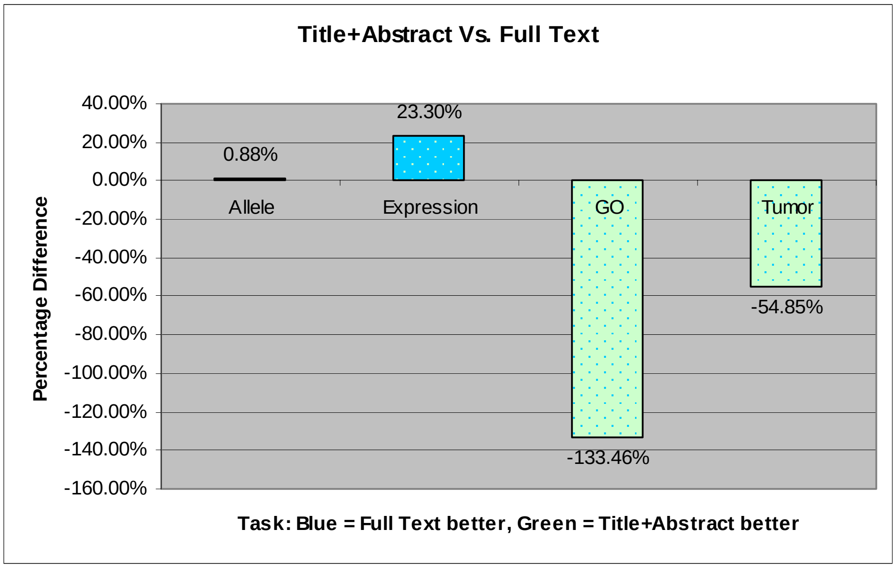

**Figure 2.** Performance comparison of Title+Abstract verses full text using stemming and stop list

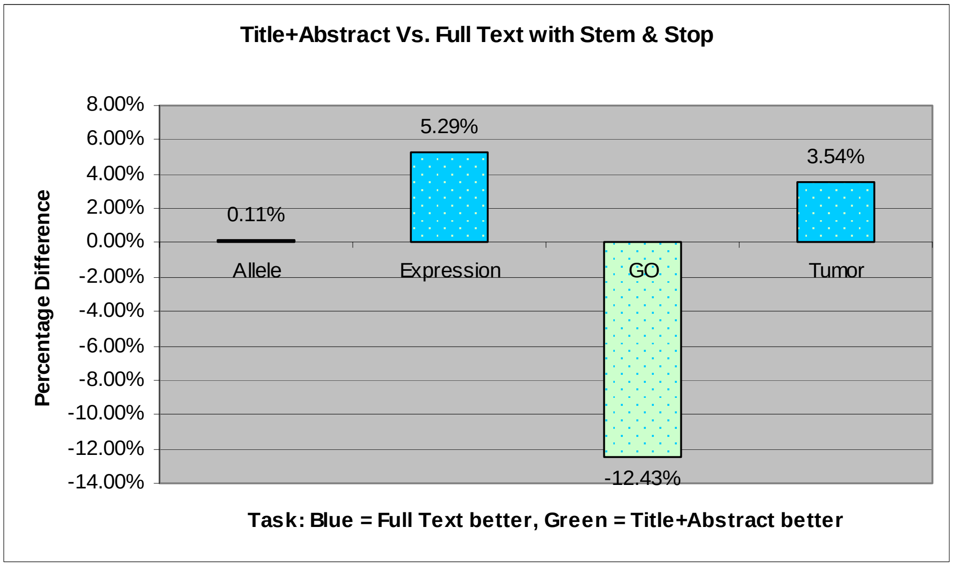

**Figure 3.** Performance comparison of Title+Abstract verses full text using stemming and stop list and MeSH term Mice prefilter

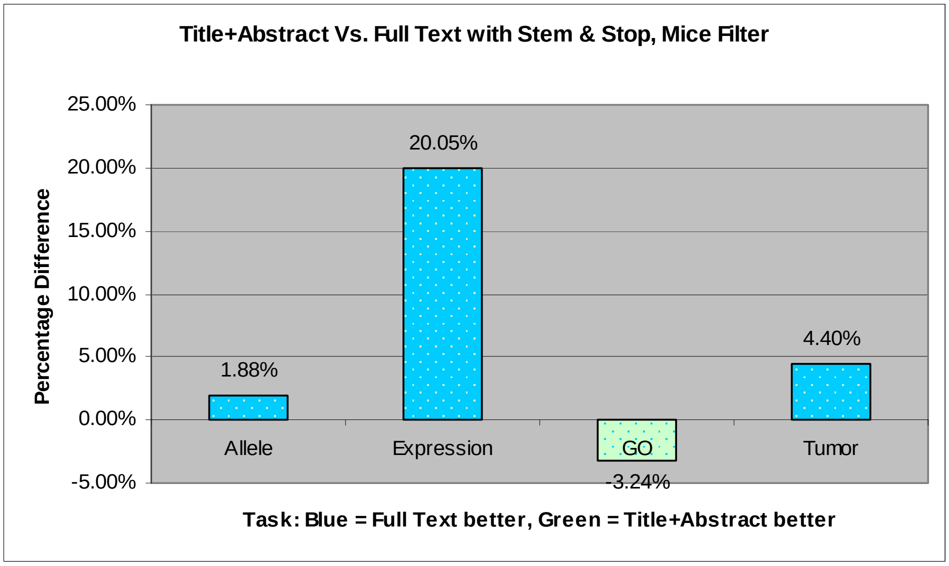

**Figure 4.** Full utility performance comparison of full text verses title+abstract using three sets of system feature.

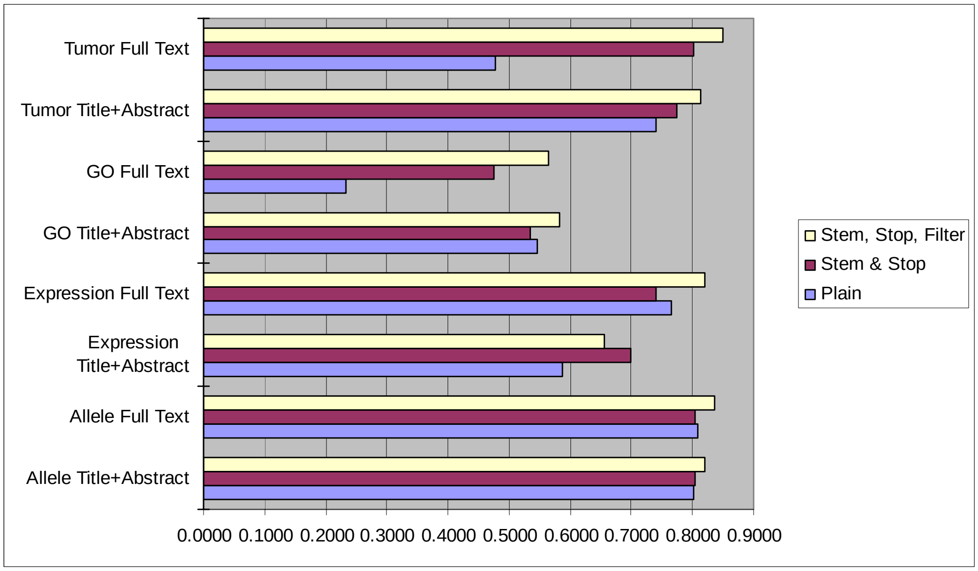

**Table 5.** Classifier system feature comparision

| Task | System Features | P | R | F | Un |
|---|---|---|---|---|---|
| *Allele* | Title & Abstract | 0.3556 | 0.8976 | 0.5094 | 0.8019 |
| | Title & Abstract, Stem & Stop | 0.3258 | 0.9157 | 0.4806 | 0.8042 |
| | Title & Abstract, Stem & Stop, Mice Prefilter | 0.2932 | 0.9548 | 0.4487 | 0.8195 |
| | Full Text | 0.4232 | 0.8795 | 0.5714 | 0.8090 |
| | Full Text, Stem & Stop | 0.4101 | 0.8795 | 0.5594 | 0.8051 |
| | **Full Text, Stem & Stop, Mice Prefilter** | **0.3854** | **0.9217** | **0.5435** | **0.8352** |
| | Full Text, Stem & Stop, Mice Prefilter, IDF | 0.3555 | 0.9036 | 0.5102 | 0.8072 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF | 0.3530 | 0.9187 | 0.5100 | 0.8196 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF & CosNorm | 0.4095 | 0.8795 | 0.5589 | 0.8049 |
| | TREC 2005 Best Un | 0.4669 | 0.9337 | 0.6225 | 0.8710 |
| | One Feature: MESH_MAIN_Mice | 0.1315 | 0.9880 | 0.2321 | 0.6042 |
| | | | | | |
| *Expression* | Title & Abstract | 0.0693 | 0.7429 | 0.1267 | 0.5869 |
| | Title & Abstract, Stem & Stop | 0.0865 | 0.8381 | 0.1569 | 0.6999 |
| | Title & Abstract, Stem & Stop, Mice Prefilter | 0.0758 | 0.8095 | 0.1385 | 0.6552 |
| | Full Text | 0.1522 | 0.8381 | 0.2577 | 0.7652 |
| | Full Text, Stem & Stop | 0.1700 | 0.8000 | 0.2805 | 0.7390 |
| | **Full Text, Stem & Stop, Mice Prefilter** | **0.1422** | **0.9048** | **0.2458** | **0.8195** |
| | Full Text, Stem & Stop, Mice Prefilter, IDF | 0.1521 | 0.8762 | 0.2592 | 0.7999 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF | 0.0897 | 0.9143 | 0.1634 | 0.7693 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF & CosNorm | 0.1176 | 0.8857 | 0.2076 | 0.7818 |
| | TREC 2005 Best Un | 0.1899 | 0.9333 | 0.3156 | 0.8711 |
| | One Feature: MESH_MAIN_Mice | 0.0405 | 0.9619 | 0.0777 | 0.6058 |
| | | | | | |
| *GO annotation* | Title & Abstract | 0.2308 | 0.7819 | 0.3564 | 0.5449 |
| | Title & Abstract, Stem & Stop | 0.2276 | 0.7741 | 0.3518 | 0.5353 |
| | **Title & Abstract, Stem & Stop, Mice Prefilter** | **0.2448** | **0.8108** | **0.3760** | **0.5834** |
| | Full Text | 0.2751 | 0.3069 | 0.2901 | 0.2334 |
| | Full Text, Stem & Stop | 0.2190 | 0.7046 | 0.3341 | 0.4761 |
| | Full Text, Stem & Stop, Mice Prefilter | 0.2337 | 0.8050 | 0.3623 | 0.5651 |
| | Full Text, Stem & Stop, Mice Prefilter, IDF | 0.2444 | 0.7761 | 0.3717 | 0.5579 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF | 0.2422 | 0.7683 | 0.3683 | 0.5498 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF & CosNorm | 0.2300 | 0.8166 | 0.3589 | 0.5681 |
| | TREC 2005 Best Un | 0.2122 | 0.8861 | 0.3424 | 0.5870 |
| | One Feature: MESH_MAIN_Mice | 0.1889 | 0.9093 | 0.3127 | 0.5542 |
| | | | | | |
| *Tumor biology* | Title & Abstract | 0.0237 | 0.9000 | 0.0461 | 0.7394 |
| | Title & Abstract, Stem & Stop | 0.0229 | 0.9500 | 0.0447 | 0.7742 |
| | Title & Abstract, Stem & Stop, Mice Prefilter | 0.0292 | 0.9500 | 0.0566 | 0.8132 |
| | Full Text | 0.0208 | 0.6000 | 0.0401 | 0.4775 |
| | Full Text, Stem & Stop | 0.0385 | 0.9000 | 0.0738 | 0.8026 |
| | **Full Text, Stem & Stop, Mice Prefilter** | **0.0397** | **0.9500** | **0.0763** | **0.8506** |
| | Full Text, Stem & Stop, Mice Prefilter, IDF | 0.0281 | 0.9500 | 0.0546 | 0.8078 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF | 0.0206 | 0.9500 | 0.0403 | 0.7545 |
| | Full Text, Stem & Stop, Mice Prefilter, TF*IDF & CosNorm | 0.0248 | 1.0000 | 0.0484 | 0.8297 |
| | TREC 2005 Best Un | 0.0709 | 1.0000 | 0.1325 | 0.9433 |
| | One Feature: MESH_MAIN_Mice | 0.0080 | 1.0000 | 0.0159 | 0.4645 |

# 3 AD-HOC RETRIEVAL TASK

## 3.1 Background

Searching MEDLINE to answer questions encountered in biomedical research is increasingly important. The TREC Genomics Track ad hoc retrieval task allows investigation of a variety of techniques that can improve the performance of biomedical document information retrieval. This specific domain differs from many others in having a complex terminology and nomenclature system. Previously, both generic and domain-specific techniques have been applied to improve performance with variable success (Hersh *et al.*, 2004).

## 3.2 System and methods

We did not use any domain-specific approaches but instead focused on two basic general IR techniques: Okapi measure parameters optimization and common stop-word exclusion in our runs. We used Zettair 0.6.1 (Billerbeck *et al.*, 2004) out of the box as our search engine and the ten sample topics as training dataset.

Indexing of the ten-year MEDLINE subset was done by first processing the documents to retain the fields we deemed relevant heuristically. These fields included the PubMed identifier (PMID), title, abstract, and MeSH headings. These were then indexed using, Zettair.

We used the zet_trec application to generate our automated queries. Our basic run, OHSUall, included all words in the narrative version of the topic file. The other run, OHSUkey, used only the words in the tables of the tabular version, excluding common words such as of, in, and, the, gene, etc.

Both of our runs used Okapi metric implemented in Zettair. The ranking function was (Billerbeck *et al.*, 2004):

$$bm25(q,d) = \sum_{t \in q} \log\left(\frac{N - f_t + 0.5}{f_t + 0.5}\right) \times \frac{(k_1 + 1) f_{d,t}}{K + f_{d,t}}$$

where terms t appear in query q; N is the number of documents in the collection; term t occurs in $f_t$ documents and in a particular document $f_{d,t}$ times; $K$ is $k_1((1-b)+b*L_d/AL)$; $L_d$ is the length of document $d$ measured in bytes, and $AL$ is the average document length over the collection. The default value of constants $k_1$ and $b$ were set to 1.2 and 0.75 respectively. These two parameters were adjusted using the sample topics provided to optimize the system performance, as measured by average Mean Average Precision (MAP) and mean precision at 10 documents (p10).

After the results of the official runs were released, we re-ran the queries with the default setting of b and $k_1$, in order to verify the effect on the system performance.

**Figure 5.** Adjusting $k_1$ and b of the Okapi metric to optimize MAP and p10.

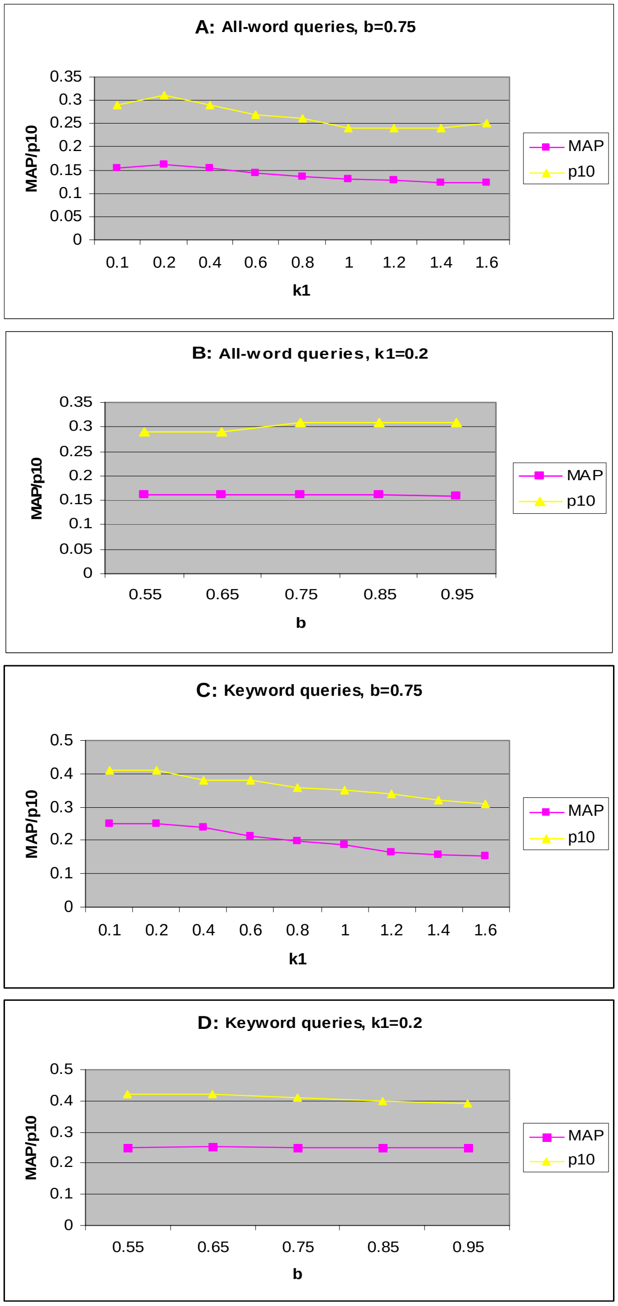

Table 6. Overall performance on 2005 test data

| Run | Average Precision | Precision @ 10 documents |
|---|---|---|
| OHSUall_submission | 0.183 | 0.3286 |
| OHSUkey_submission | 0.2233 | 0.3735 |
| OHSUall_default | 0.1981 | 0.3653 |
| OHSUkey_default | 0.2117 | 0.3755 |

### 3.3 Results

As shown in Figure 5, our training runs with only keywords consistently outperformed runs with all words from the topics. The average difference in MAP was about 0.10. Furthermore, when holding b constant at its default value, MAP and p10 reached a maximum at $k_1$=0.2. On the other hand, holding $k_1$ at 0.2 while changing b provided little improvement in MAP and p10. Therefore, in both of our final runs, k1 was set to 0.2.

Table 6 shows the overall performance on the test data for our official submissions and later baseline default parameter runs. Queries using keywords still outperformed all-word queries, but the gap was narrower when we used the default settings for b and $k_1$. In runs with keywords only, the official submission had higher MAP than run with default $k_1$ and b values, but the default setting faired better if we used all the words in the topics.

### 3.4 Discussion

In this year's ad hoc retrieval task, we used the simple and fast Zettair search engine off the shelf, automatically generated queries from topic descriptions, and manipulated Okapi metric parameters to optimize our performance. Our best system, OHSUkey, was among the median performers. Our runs with keywords indicated that simple exclusion of common words can be very helpful. The experiments with adjusting Okapi metric parameters were inconclusive. The performance improvement in keyword searches due to lowering $k_1$ agreed with the intuition that high term frequency is not as important as having all the key terms in the article. On the other hand, if we used queries with the common words, higher term frequency, especially of the keywords, should have led to higher ranking of documents containing those words. The difference between training and test data may cause the adjusted parameters not as optimized in test as in training.

## 4 CONCLUSIONS

The TREC 2005 Genomics Track enhanced our understanding of biomedical document classification. Full text is essential for high performance classification, but stemming and stopping must be applied to the text before features are extracted. These techniques are also effective on just the title plus abstract, but do not result in the same level of performance. Binary feature weighting is adequate, as the various term weighting schemes such as IDF, TF*IDF, and cosine normalization actually decreased performance. Domain specific filtering, such as the MeSH Mice term pre-filter used here, can also increase performance. Our standard system consisting of a voting perceptron classifier, chi-square feature selection on full text articles, binary feature weighting, stemming and stopping, and pre-filtering based on the MeSH term Mice, approached, but did not surpass, the performance of the best track entry for each of the four tasks. Performance on three of the four tasks, allele, expression, and tumor biology, was high, and likely good enough to provide real-world benefit to MGI's triage process.

In the ad hoc retrieval task, we experimented with adjustment of Okapi metric parameters and contrasting keyword and plain text search. We found that simple elimination of common words helped and, in keyword search, that lowering the weight of term frequency improved performance a modest amount.

## ACKNOWLEDGEMENTS

This work was supported in part by Grant ITR-0325160 from the National Science Foundation.

## REFERENCES

Billerbeck, B., Cannane, A. and *et al.* (2004) RMIT University at TREC 2004. In *Proceedings of the Text Retrieval Conference (TREC) 2004*.

Cohen, A. M., Hersh, W. R. and Bhupatiraju, R. T. (2004) Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage. In *Proceedings of the Text Retrieval Conference (TREC) 2004*.

Cohen, A. M. (2005) Unsupervised gene/protein entity normalization using automatically extracted dictionaries. In *Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics, Proceedings of the BioLINK2005 Workshop*.

Cohen, W. W. and Singer, Y. (1999) A Simple, Fast, and Effictive Rule Learner. In *Proceedings of the Annual Conference of the American Association for Artificial Intelligence (AAAI)*.

Freund, Y. and Schapire, R. E. (1999) Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, **37**, 277-296.

Hersh, W.R. and et al. (2004) TREC 2004 Genomic Track Overview. In *Proceedings of the Text Retrieval Conference (TREC) 2004*.

Kostoff, R. N., Block, J. A., Stump, J. A. and Pfeil, K. M. (2004) Information content in Medline record fields. *Int J Med Inf*, **73**, 515-27.