DIMACS AT THE TREC 2004 GENOMICS TRACK

Aynur Dayanik†, Dmitriy Fradkin†, Alex Genkin†, Paul Kantor†,
David D. Lewis‡, David Madigan†, Vladimir Menkov*
{aynur,dfradkin,agenkin,paul.kantor,dmadigan}@rutgers.edu,
trec2004@daviddlewis.com, vmenkov@cs.indiana.edu
DIMACS, Rutgers University †
David D. Lewis Consulting ‡
Aqsaqal Enterprises, Penticton, BC, Canada*

ABSTRACT

DIMACS participated in the text categorization and ad hoc retrieval tasks of the TREC 2004 Genomics track. For the categorization task, we tackled the triage and annotation hierarchy subtasks.

1. TEXT CATEGORIZATION TASK

The Mouse Genome Informatics (MGI) project of the Jackson Laboratory[1] provides data on the genetics, genomics, and biology of the laboratory mouse. In particular, the Mouse Genome Database (MGD) contains information on the characteristics and functions of genes in the mouse, and on where this information appeared in the scientific literature. Human curators encode this information using controlled vocabulary terms from the Gene Ontology[2] (GO), and provide citations to documents that report each piece of information. GO consists of three structured networks: *Biological Process (BP)*, *Molecular Function (MF)*, and *Cellular Component (CC)*) of terms describing attributes of genes and gene products.

The TREC 2004 Genomics track defined a categorization task with three subtasks based on simplified versions of this curation process. DIMACS participated in two of those subtasks, *triage* and *annotation hierarchy*, but not in the *annotation hierarchy plus evidence* subtask. We discuss our two subtasks below, and full details are available in the track overview paper [4].

1.1 Triage Subtask

To find information on mouse genes, MGI first automatically scans new scientific literature for records containing one or more of the words "mouse", "mice", and "murine". In a triage step, MGI personnel then check each article to see if it contains information appropriate for inclusion in MGD. (The triage step also identifies articles for other purposes, but we can ignore that here.)

The TREC 2004 triage subtask is intended to simulate the problem faced by triage personnel. Full text articles published in 2002 and 2003 by three major journals were obtained. Those articles containing "mouse", "mice", or "murine" were identified and separated into a training set (5837 documents from 2002) and a test set (6043 documents from 2003).

The goal for subtask participants was to identify which of the articles from the test set had, during MGI's operational manual triage process, been chosen for sending to GO curators. (Whether curators had or hadn't actually linked to this document from any MGD entry was not an issue.) We can view this as a binary text classification problem, with articles chosen for curation during the triage process being positive examples, and those rejected during triage being negative examples. Logs from MGI were used to produce relevance judgments for the subtask data. Subtask participants were given the relevance judgments for the training set, which showed that 375 of the training set articles were positive examples (had been selected for curation) and 5462 training articles were negative examples. The test set relevance judgments, revealed after official runs were submitted, showed 420 positive and 5623 negative test examples.

The official effectiveness measure for the triage subtask was this normalized linear utility:

$$T13NU = \frac{T13U}{T13U_{max}}$$

where

$$\begin{aligned} T13U &= 20 * TP - FP \\ T13U_{max} &= 20 * (TP + FN). \end{aligned}$$

TP, FP, and FN are defined in the confusion matrix in Table 1. Table 2 shows the values of T13NU for the boundary cases on the test data set.

1.2 Annotation Hierarchy Subtask

Articles that pass MGI's triage process are examined by GO curators. They identify mouse genes and gene products mentioned in the article, claims that they have certain characteristics, and the type of evidence for those characteristics. These characteristics are recorded using the appropriate GO terms, type of evidence is recorded using other codes, and the document is recorded as the source of the evidence.

The annotation hierarchy subtask is a very simplified version of this curation process. A system is given a pair (D, G), where G is a gene discussed in document D. The system must decide whether D's discussion of G contains information appropriate for coding with GO terms and, if so, in which of the three GO hierarchies those GO terms would fall. Systems are not required to identify the particular GO terms.

The subtask provides a set of 1418 document/gene pairs (representing 504 distinct documents and 1291 distinct genes) for training and 877 pairs for testing (378 documents and 773

[1]http://www.informatics.jax.org
[2]http://www.geneontology.org

	Relevant	Not relevant
Retrieved	True positive (TP)	False positive (FP)
Not retrieved	False negative (FN)	True negative (TN)

Table 1: Confusion table.

Situation	**T13NU - Test**
Completely perfect prediction	1.0
Predict using MeSH term "Mice"	0.64
Best submitted run	0.65
Triage everything	0.33
Triage nothing	0
Completely imperfect prediction	-0.67

Table 2: Boundary cases for T13NU on triage subtask test set.

distinct genes). These sets of pairs were formed by roughly this process:

1. A set of GO records were found that had links to documents from the track data set. One can think of the records as tuples of the form *(G, GO term, evidence, D)*. These tuples were mapped to the form (D, G, XY) by replacing the GO term with the label for the hierarchy it falls in (BP, CC, or MF). Redundant tuples were discarded. This resulted in the records in the files *pgd+train.txt* and *pgd+test.txt*. The presence of a tuple (D, G, XY) in, say, *pgd+train.txt* means that pair (D, G) is a positive example for class XY. If some (D, G, XY) is present in one of these files, but (D, G, WZ) is not present for some $WZ \neq XY$, then (D, G) is a negative example for class WZ.

2. A set of documents from the track data set were found that were selected during triage for purposes other than GO curation. For each such document, and one or more genes identified in that document, a record of the form (D, G) was included in *pg-train.txt* or *pg-test.txt*. Each pair (D, G) in *pg-train.txt* and *pg-test.txt* is viewed as a negative example for all three of BP, CC, and MF.

Note that there are two sources of negative examples, but only one of positive examples. All examples, both positive and negative, are listed in *pgtrain.txt* and *pgtest.txt*. Thus, the goal of a system was to identify for each pair (D, G) in *pgtest.txt*, whether or not it should be assigned each of BP, CC, and MF.

We treated this decision as three separate binary classification problems. This meant we created three copies of the training and test vectors, one for each of GO hierarchy labels. A document/gene pair (D,G) became a positive example for label XY (where XY is BP, CC or MF) if a record of the form (D G XY) is present in *pgd+train.txt* or *pgd+test.txt*, and a negative example for XY otherwise. Table 8 shows the number of positive instances for each topic in training and test data.

The official effectiveness measure for the annotation hierarchy subtask is F1 (F-measure with equal weight on recall and precision) [10, 9, 5] where

$$\begin{aligned} \text{Precision (p)} &= TP/(TP+FP) \\ \text{Recall (r)} &= TP/(TP+FN) \\ F1 &= \frac{2*r*p}{r+p} = \frac{2*TP}{2*TP+FP+FN} \end{aligned}$$

2. BAYESIAN LOGISTIC REGRESSION FOR THE TEXT CATEGORIZATION TASK

Logistic regression models estimate the probability that an example belongs to a class using this formula:

$$p(y_i = +1|\boldsymbol{\beta}, \boldsymbol{x_i}) = \frac{\exp(\boldsymbol{\beta}^T \boldsymbol{x_i})}{1+\exp(\boldsymbol{\beta}^T \boldsymbol{x_i})} = \frac{\exp(\sum_j \beta_j x_{i,j})}{1+\exp(\sum_j \beta_j x_{i,j})}$$

where y_i encodes the class of example i (positive/relevant $= +1$, negative/nonrelevant $= -1$) and $x_{i,j}$ is the value of feature j for example i. The model parameters $\boldsymbol{\beta}$ are chosen by supervised learning, i.e. by optimizing some function defined on a set of examples for which manually judged values of y_i are known.

In our work, we adopt a Bayesian framework and choose the $\boldsymbol{\beta}$ that maximizes the posterior loglikelihood of the data,

$$l(\boldsymbol{\beta}) = (-\sum_{i=1}^{n} \ln(1+\exp(-\boldsymbol{\beta}^T \boldsymbol{x_i} y_i)) + \ln p(\boldsymbol{\beta}),$$

where $p(\boldsymbol{\beta})$ is, for each $\boldsymbol{\beta}$, the prior probability that $\boldsymbol{\beta}$ is the correct parameter vector. The prior $p(\boldsymbol{\beta})$ encodes what we believe are likely values of $\boldsymbol{\beta}$ before seeing the training data.

Our experiments use the BBR (Bayesian Binary Regression) software.[3] BBR supports two forms of priors: a separate Gaussian prior for each β_j or a separate Laplace prior for each β_j. (The overall prior is the product of the individual priors for feature parameters.) The key difference between the two is that Gaussian priors produce dense parameter vectors with many small but nonzero coefficients, while Laplace priors produce sparse feature vectors with most coefficients identically equal to 0.

We describe BBR and Bayesian logistic regression in detail elsewhere [3]. Here we review only a few details necessary to interpreting our results.

[3]http://www.stat.rutgers.edu/∼madigan/BBR/

2.1 Choice of Hyperparameter

The Gaussian and Laplace priors have two hyperparameters for each model parameter β_j: a modal value μ_j (the most likely prior value of β_j), and a regularization hyperparameter (σ_j^2 for Gaussian and λ_j for Laplace) that indicates how close to μ_j we expect β_j to be. For simplicity, our TREC work assumes all μ_j's are 0, and that the regularization hyperparameter is the same for all features. This leaves a single regularization hyperparameter to be chosen for the whole model.

Rather than specifying this regularization hyperparameter manually based on our prior beliefs, we use an empirical Bayes approach [2], and choose it by cross-validation on the training set. We consider a fixed set of hyperparameter values, and choose the one that maximizes the 10-fold cross-validation estimate of mean posterior log-likelihood on the training data. The values considered were

$$1, 2.25, 4, 6.25, 9, 12.25, 16, 20.25, 25, 30.25, 36, 42.25, 49, 56.25, 64, 100, 10000, 1000000, 100000000$$

for Gaussian, and

$$1.41, 0.943, 0.707, 0.566, 0.471, 0.404, 0.354, 0.314, 0.283, 0.257, 0.236, 0.218, 0.202, 0.189, 0.177, 0.141, 0.014, 0.00141, 0.00014$$

for Laplace. Both these sets correspond to this set of prior standard deviations

$$1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 10, 100, 1000, 10000$$

2.2 Threshold Selection

Logistic regression models estimate the probability that the example is a positive/relevant example. We then must convert this probability to a binary class label. The simplest approach is to define a threshold value that the estimated probability must exceed for the test example to be predicted to be relevant.

We tested two approaches to choosing a threshold for a categorization problem:

- MEE (Maximum Expected Effectiveness): Choose the threshold that maximizes the expected value of the subtask effectiveness measure on the test set, under the assumption that the estimated class membership probabilities are correct and independent [5].
- TROT (Training set Optimization of Threshold): Choose the threshold that maximizes the subtask's effectiveness measure on the training set.

Both TROT and MEE were tested by cross-validation on the triage subtask training data. MEE was found consistently better and so was used for all our triage runs. The MEE threshold for the T13NU effectiveness measure is $p(y_i = +1) >= 1/21 = 0.0476$ on a probability scale.

TROT was used for all annotation runs. Computing the MEE threshold for F1 requires processing test examples as a batch [5], something not allowed by the track guidelines.

2.3 Two-Stage Classifiers

The importance of the MeSH term "Mice" in the triage subtask (see Section 4.2) was apparent in our experiments on the training set. Therefore, in addition to one-stage thresholded logistic regression models, we also tested the following two-stage classifier on the triage task:

1. IF a document does NOT contain the MeSH term "Mice" classify it as negative.
2. ELSE classify it using a thresholded logistic regression model.

The logistic regression models used in the two-stage classifier were trained only on training examples containing the MeSH term "Mice". The hope was that this would train the model to focus less on whether the document was about mice, and more on distinguishing whether evidence about gene characteristics was present.

2.4 Upweighting of Positive Examples

The proportion of positive examples in the annotation hierarchy subtask was low, and for that subtask we experimented with upweighting positive training examples relative to negative ones. This was done by making $w-1$ extra copies of each positive training example. The weights tried were: $w = 1$ (no upweighting), $w = 5$, and $w = 6$. The replicated examples were used both when fitting model parameters and when tuning the threshold.

3. TEXT REPRESENTATION FOR TEXT CATEGORIZATION SUBTASKS

The track provided the full text of the journal articles in both SGML and XML form. We used the XML versions from *train.xml.zip* and *test.xml.zip*. We also made use of additional descriptions of each article. The track files *train.crosswalk.txt* and *test.crosswalk.txt* specified the PubMed ID for each article. We used these IDs to obtain the MEDLINE record for each article either from the ad hoc track data, or by downloading from PubMed.[4]

The MEDLINE records for 536 of the training articles and 408 of the test articles contain GenBank accession numbers for genes discussed in the article. While this is done for only a subset of genes mentioned, it is a useful clue when present, because the GenBank entry specifies the organism a gene was studied in. Using the accession number, we downloaded the corresponding GenBank[5] record, and extracted the organism name field.

These materials gave several alternative sources of representations for the training and test articles:

- **Full Text**: The union of text from the title (<*atl*>), subject (<*docsubj*>), abstract (<*abs*>), and body (<*bdy*>) XML elements of the article.
- **Abstract**: The union of text from the subject, title, and abstract of the article.
- **MEDLINE**: The MeSH terms, Medical Subject Headings, from the MEDLINE record (lines starting with *"MH - "* in ASCII text format), plus the union of text from the title (<*ArticleTitle*>) and abstract (<*Abstract*>) elements of that record. MeSH terms were converted to single tokens (Section 3.1) and so were kept distinct from the two text fields.

[4]http://eutils.ncbi.nlm.nih.gov/entrez/query.fcgi
[5]http://www.ncbi.nlm.nih.gov/entrez/batchentrez.cgi?db=Nucleotide

- **MeSH**: Only the MeSH terms from the MEDLINE record.
- **GenBank**: The organism name from the GenBank record, converted to a single token (Section 3.1).

Various combinations of these representations were tried on the training data and a subset were selected for the submitted runs.

3.1 Text processing

For the full text articles, we extracted the contents of the specified XML elements for the particular representation (see above), concatenated those contents, and deleted all the internal XML tags. Text processing was done using the Lemur[6] utility ParseToFile, in combination with the Porter stemmer [6] supplied by Lemur and the SMART [7] stoplist.[7] This parser performed case-folding, replaced punctuation with whitespace, and tokenized text at whitespace boundaries. The Lemur utility BuildBasicIndex was used to construct Lemur index files, which we then converted to document vectors in BBR's format.

MEDLINE records were handled the same way, except that MeSH terms were converted to single tokens (e.g. replacing "Mice, Knockout" with "MHxxxMicexxxKnockout") before Lemur processing to force them to have a separate term ID than words. GenBank organism names were similarly converted to single terms (e.g. "Mus musculus" to "GenBankxxxMusxxxmusculus").

3.2 Term Weighting

BBR requires text to be represented as vectors of numeric feature values. For both annotation and triage subtasks we used TFxIDF (term frequency times inverse document frequency) weighting [8], with IDF weights computed on the training instances only.

We describe our weighting methods using Cornell triple notation [8], i.e. TCN, where

- *T = Term Frequency Component:*
  - b : binary, 1.0 if term is present, 0.0 if not
  - l : "log tf", i.e. $1+\log_e(tf)$ if term is present, 0.0 if not
- *C = Collection Frequency Component:*
  - x : 1 for all terms
  - L = "lookahead IDF": $\log_e \frac{(N+1)}{(n_j+1)}$.
- *N = Normalization Component:*
  - x : no normalization
  - c : Cosine normalization, i.e. the feature vector is normalized to have a Euclidean norm of 1.0.

Here N is the number of documents from which IDF weights are computed (the categorization training set, so $N = 5837$ for the triage subtask and $N = 1418$ for the annotation hierarchy subtask), and n_j is the number of documents containing term j. *Lookahead IDF* (which we indicate by nonstandard symbol "L") is a version of IDF weighting that defines a value even for terms that do not occur in the training corpus. It can be viewed as including a future document being weighted in the set of documents used to define term weights for it, thus the name "lookahead".

Note that different representations of documents or document/gene pairs will produce different IDF weights, and thus different final term weights.

Test set terms that do not occur in the training set introduce a question about how to do cosine normalization. Terms that are unique to the test set can never contribute to a document's score. These test set terms can, however, depress the within-document weights of other terms in test set documents, through their impact on cosine normalization. We therefore tested two variants of cosine normalization:

- Normalize & Project (N&P) : Terms that occur only on the test set are included in the test set vectors during cosine normalization, and then removed (for efficiency).
- Project & Normalize (P&N) : Terms that occur only on the test set are removed from test set vectors before cosine normalization.

3.3 Document Representation for the Annotation Hierarchy Subtask

We have so far discussed representations that capture the entire contents of a document at some level of granularity. Such approaches were used to represent documents in all our triage runs, and to represent document/gene pairs for three of our submitted annotation runs (dimacsAabsw1, dimacsAl3w, and dimacsAg3mh).

For the annotation subtask, we also tested representations of document/gene pairs that take the gene into account. In particular, we attempted to identify portions of the document that discussed the particular gene. Two gene-specific representations were tried:

- **Paragraphs**: We separated the body text (<*bdy*> element) of the article into paragraphs (<*P*> elements). We then retained in the representation of the pair only those paragraphs that contained at least one term from the "gene description" (see below).
- **Windows**: For each term in the gene description, we extract from the document all windows of half-size k (i.e. $2k+1$ terms per window, except at the beginning and end of the document) centered at an occurrence of that term. The document/gene pair is represented by the union of these windows. Note that windows sometimes overlap if multiple terms from a gene description occur near each other. This increases the frequency of words that occur close to many gene terms. In some cases a term can even have a higher frequency in the document/gene description than it has in the full document.

We computed term weights from the resulting representations of document/gene pairs as if each document/gene pair was a document.

Biomedical articles, unfortunately, may refer to a gene using any of several, possibly nonstandard, symbols and/or

[6] http://www-2.cs.cmu.edu/~lemur

[7] ftp://ftp.cs.cornell.edu/pub/smart/english.stop or http://jmlr.csail.mit.edu/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm

names for the gene and/or its products [12]. We therefore tested several approaches to producing gene descriptions:

- **Symbol**: The description consisted solely of the MGI gene symbol which *pgtrain.txt* or *pgtest.txt* lists for the document/gene pair.
- **Name**: The description included the MGI gene name which *gtrain.txt* or *gtest.txt* lists for the gene. The **Name** description is produced by replacing the characters

  `[](),+`

  in those names with whitespace, downcasing the text, and separating the result into terms at whitespace boundaries. No stemming was used.
- **Locuslink**: We downloaded a copy of LocusLink[8], a database linking disparate information on genes, on 20 July 2004. For each gene symbol, we found the corresponding LocusLink record, extracted the contents of the OFFICIAL_GENE_NAME and ALIAS_SYMBOL fields, and separated the contents into terms.

Combinations (e.g. **Symbol + Name**, **Symbol + Name + LocusLink**) of these representations were also tested, with duplications of terms across representations removed.

For example, *pgd+train.txt* contains the record

12213961 Map2k6 BP.

This reflects an MGD record stating document *12213961* presents evidence of one or more biological processes (*BP*) that gene *Map2k6* is relevant to. In our **Symbol** representation, the gene description was thus simply

Map2k6.

In the **Symbol + Name** representation, the gene description was:

Map2k6 mitogen activated protein kinase kinase 6,

and in the **Symbol + Name + LocusLink** representation it was:

Map2k6 mitogen activated protein kinase kinase 6 MEK6 MKK6 Prkmk6 SAPKK3.

4. TRIAGE SUBTASK EXPERIMENTS

After submitting our official triage subtask runs we discovered a few software bugs, and so re-ran each run with the corrected code. The corrected runs also allowed us to to clarify our techniques by omitting CPU-saving shortcuts used in our official runs (e.g. fractional cross-validation and reduced sets of hyperparameter values). We present effectiveness data on both the official and corrected runs. Results were similar, so we give detailed descriptions only of the corrected runs.

Our triage runs used the following techniques:

- dimacsTfl9d : Representation: MEDLINE. Weighting: lLc (N&P). Classifier form: two-stage. Prior: Laplace. Hyperparameter: 0.404.
- dimacsTfl9w : Representation: Full-text. Weighting: lLc (N&P). Classifier form: two-stage. Prior form: Laplace. Hyperparameter: 0.354.
- dimacsTl9md : Representation: MEDLINE. Weighting: lLc (N&P). Classifier form: one-stage. Prior: Laplace. Hyperparameter: 0.354.
- dimacsTl9mhg : Representation: MeSH + GenBank. Weighting: bxx. Classifier form: one-stage. Prior: Laplace. Hyperparameter: 1.41.
- dimacsTl9w : Representation: Full-text. Weighting: lLc (N&P). Classifier form: one-stage. Prior: Laplace. Hyperparameter: 0.404.

All of the triage runs, both submitted and corrected, runs used MEE thresholding (a threshold of 0.0476 on a probability scale). All corrected runs used full 10-fold cross-validation on the training set to choose a hyperparameter (shown above for each run) from the values listed in Section 2.1.

The combinations of techniques submitted were chosen by cross-validation experiments on the training data. Not all combinations were exhaustively tried.

4.1 Results

Our official triage subtask results are summarized in Table 3. Run dimacsTfl9d was our best scoring run, and indeed was the best among all submitted runs (Table 5). Table 4 shows the corrected runs that correspond to each official triage run.

Looking at the above runs, and others we do not have space to include, shows that Laplace priors were consistently more effective than Gaussian priors. This is not surprising, given that a *very* small feature set was able to give high effectiveness (see next Section). MEE thresholding was considerably more effective than TROT thresholding, which suggests a benefit to this approach when the desired tradeoff between false positives and false negatives is extreme. In contrast to the annotation subtask, P&N and N&P cosine normalization gave almost identical effectiveness.

4.2 Data Set Issues

Run dimacsTfl9d, the subtask's best run, uses only the MEDLINE record, not the full text document. This is disturbing, since it suggests participating systems were not successfully making judgments about the presence of experimental evidence in the document text.

The news gets worse. We show in Table 3 a hypothetical run where a test document is classified positive if its MEDLINE record contains the MeSH term "Mice", and negative otherwise. This run would have beaten all runs submitted by other groups! As far as we can tell from the results, no system successfully distinguished documents that discuss mice in general, from documents that contain GO-codable information appropriate for MGD.

On the other hand, the problem might be in the track data. MGD is a database of facts about genes, not facts about documents. Pointers to documents are included to provide citations for these facts, but providing comprehensive access to the scientific literature is not the goal of the database. It seems plausible that, in making the triage decision, MGI personnel may be less likely to designate for

[8]ftp://ftp.ncbi.nlm.nih.gov/refseq/LocusLink

annotation documents that appear to report already well-known facts about mouse genes. This would have little relevance to GO users, but could play havoc with classification experiments. More discussions with MGI personnel, and interindexing consistency studies, would be desirable.

An additional minor problem with the track data, which we and other groups detected only after official submissions, was that 4 of 420 positive test documents were omitted in 6043 test set documents (i.e. in test.crosswalk.txt file) and some documents given as negative documents were found to be positive after the submissions.

5. ANNOTATION HIERARCHY SUBTASK EXPERIMENTS

For each of our annotation hierarchy subtask runs we trained three thresholded logistic regression classifiers, one for each of the BP, CC, and MF hierarchies. As with our triage runs, we found some bugs after submission and so re-ran each run with the corrected code. Our runs were:

- dimacsAabsw1: Representation: Abstract. Weighting: lLc (P&N). Prior: Gaussian. Upweighting of positive examples: no ($w = 1$).
- dimacsAg3mh: Representation: MeSH. Weighting: bxx. Prior: Gaussian. Upweighting of positive examples: no ($w = 1$).
- dimacsAl3w: Representation: Full text. Weighting: lLc (P&N). Prior: Laplace. Upweighting of positive examples: no ($w = 1$).
- dimacsAp5w5: Representation: Paragraphs, selected using Locuslink information. Weighting: lLc (P&N). Prior: Gaussian. Upweighting of positive examples: yes ($w = 5$).
- dimacsAw20w5: Representation: Windows with half-window size 20, selected using LocusLink information. Weighting: lLc (P&N). Prior: Gaussian. Upweighting of positive examples: yes ($w = 5$).

All submitted, and corrected, annotation runs chose a threshold based on optimizing the training set F1 (TROT approach). All corrected runs used full 10-fold cross-validation on the training set to choose hyperparameter values from those listed in Section 2.1.

The results of our 5 official runs are given in Table 6. NIST statistics on all official runs are given in Table 7.

All submitted runs (except the binary representation dimacsAg3mh) used the P&N variant of cosine normalization. Tables 9 and 10 compare corrected runs with the P&N versus the N&P variants. Run dimacsAg3mh is not normalized, and so appears identical in the two tables.

5.1 Discussion

The effectiveness of our annotation submissions varied considerably, with the best (dimacsAl3w) a respectable $F1 = 0.49$. Disappointingly, our runs using gene-specific representations of pairs (dimacsAp5w5 and diamcsAw20w5) scored substantially worse than runs using document-based representations. Gene-specific representations had higher precision than document-based methods, but much lower recall.

One problem with gene-specific representations was that some documents discussing a gene contain few or no terms from the gene description, even with gene descriptions expanded using LocusLink. (The use of LocusLink to expand the gene descriptions did improve effectiveness slightly, as shown in Table 11.) Even with the richest gene descriptions (**Symbol + Name + LocusLink**), there were 54 training document/gene pairs and 80 test document/gene pairs with empty vectors for the paragraph-based representation. Similarly, there were 38 training pairs and 67 test pairs with empty vectors for all window-based representations. (The paragraph and window representations differ because the paragraph representation did not use the title or abstract of the document, while the window representation did.) A weighted combination of the full document and the gene-specific passages might improve the situation.

For weighted representations, the P&N variant of cosine normalization was substantially more effective than the N&P variant. This is somewhat surprising. Cosine normalization is meant to compensate for unequal document lengths, and there seems little reason that it should matter how many terms in a test document also occurred in the training set. We suspect that the rich vocabulary of technical documents, and the relatively small training set, is causing test document vectors to have many novel terms. Our lookahead IDF weighting gives these terms large weights, thus reducing (via cosine normalization) the weights of all other terms under N&P normalization, but not P&N normalization. The benefit for the counterintuitive P&N normalization is likely to disappear if we remove IDF weights from the document representation (where they arguably do not really belong) and instead take them into account in our Bayesian prior.

As for variations on the learning approach, Gaussian priors were almost always more effective than Laplace priors for this task. This is not surprising given the very large vocabulary implied by a full GO hierarchy. Gaussian priors usually gave better precision than Laplace priors, but worse recall, though this may simply be a problem with choosing thresholds for F1. Upweighting positive examples improved effectiveness on document-based representations, but not with gene specific ones. Again, we hope to eliminate the need for this with better thresholding and choice of regularization parameters.

Training data results suggested that the 3 annotation hierarchy classification problems (BP, CC, MF) would have benefited from different machine learning and representation approaches. Due to time and resource constraints we did not take advantage of this in our runs, but doing so would be important in the operational setting.

5.2 Data Set Issues

The test set had a substantially higher proportion of relevant pairs than the training set (Table 8). This increase would not have affected the best threshold for a linear utility effectiveness measure (like T13NU), but does change the best threshold for a nonlinear effectiveness measure such as F1. Our test set results were substantially lower than we expected from cross-validation runs on the training data, and this change may be one reason.

While the annotation subtask does not have a smoking gun analogous to the triage subtask's MeSH "Mice" classifier (Section 4.2), we have similar concerns about the consistency of relevance judgments for the annotation task as well. It is

Run	TP	FP	FN	TN	Precision	Recall	F-score	T13NU
dimacsTfl9d	373	1990	47	3633	0.1579	0.8881	0.2681	0.6512
dimacsTfl9w	371	2018	49	3605	0.1553	0.8833	0.2642	0.6431
dimacsTl9md	334	1597	86	4026	0.1730	0.7952	0.2841	0.6051
dimacsTl9mhg	376	2108	44	3515	0.1514	0.8952	0.2590	0.6443
dimacsTl9w	279	1637	141	3986	0.1456	0.6643	0.2389	0.4694
"Mice" run	375	2121	45	5627	0.1502	0.8929	0.2572	0.6404

Table 3: Our official triage subtask results, plus a hypothetical test set run using only MeSH term "Mice".

Run	TP	FP	FN	TN	Precision	Recall	F-score	T13NU
dimacsTfl9d	373	2072	47	3551	0.1526	0.8881	0.2604	0.6414
dimacsTfl9w	373	2080	47	3543	0.1521	0.8881	0.2597	0.6405
dimacsTl9md	355	1751	65	3872	0.1686	0.8452	0.2811	0.6368
dimacsTl9mhg	359	1798	61	3825	0.1664	0.8548	0.2786	0.6407
dimacsTl9w	314	1974	106	3649	0.1372	0.7476	0.2319	0.5126

Table 4: Test set results from rerunning our triage submissions with corrected software.

	Best	Median	Worst
Precision	0.2309	0.1360	0.0713
Recall	0.9881	0.5571	0.0143
F-score	0.2841	0.1830	0.0267
T13NU	0.6512	0.3425	0.0114

Table 5: NIST-supplied statistics on effectiveness of official triage submissions (59 triage runs, 20 participants).

Run	TP	FP	FN	TN	Precision	Recall	F-score
dimacsAabsw1	113	76	382	501	0.5979	0.2283	0.3304
dimacsAg3mh	225	196	270	381	0.5344	0.4545	0.4913
dimacsAl3w	162	161	333	416	0.5015	0.3273	0.3961
dimacsAp5w5	96	81	399	496	0.5424	0.1939	0.2857
dimacsAw20w5	83	55	412	522	0.6014	0.1677	0.2622

Table 6: Our official annotation hierarchy subtask results.

	Best	Median	Worst
Precision	0.6014	0.4174	0.1692
Recall	1.0000	0.6000	0.1333
F-score	0.5611	0.3584	0.1492
T13NU	0.7842	0.5365	0.1006

Table 7: NIST-supplied official annotation hierarchy results (36 runs, 20 participants).

easy to imagine that GO curators are less likely to include a link to the 10th document mentioning a particular fact about a gene than they are to the first document.

6. AD HOC RETRIEVAL TASK

The ad hoc retrieval task assessed text retrieval systems on information needs of real biomedical researchers. The detailed description of the task is given in the track overview paper [4]. Here we give a brief summary.

Document Collection. The document collection consisted of 10-year subset (from 1994 to 2003) of the MEDLINE database of the biomedical literature. The DCOM field of the MEDLINE records was used to define "date" for selecting this 10-year subset. The collection included 4,591,008 MEDLINE records (about 10 gigabytes in size).

Topics. The track supplied 5 sample topics with incomplete relevance judgments so participants would know what to expect. The test data consisted of 50 topics. All 55 topics (sample and test) were constructed from information needs of the real biomedical researchers. Each topic was represented with a *title*, *need* and *context* field. A sample topic is shown in Table 12.

Relevance Judgements. All relevance judgments were done by two people with backgrounds in biology, but not the creators of the original information needs. A pool of doc-

Topic	Training		Test	
	# Relevant Pairs	% Relevant Pairs	# Relevant Pairs	% Relevant Pairs
BP	228	0.161	170	0.194
CC	163	0.115	131	0.149
MF	198	0.140	194	0.221
Total	589	0.138	495	0.188

Table 8: Number of relevant pairs in the training and test sets for the annotation hierarchy subtask.

Run	TP	FP	FN	Precision	Recall	F-score
dimacsAabsw1	113	93	382	0.5485	0.2283	0.3224
dimacsAg3mh	201	186	294	0.5194	0.4061	0.4558
dimacsAl3w	242	248	253	0.4939	0.4889	0.4914
dimacsAp5w5	92	61	403	0.6013	0.1859	0.2840
dimacsAw20w5	90	58	405	0.6081	0.1818	0.2799

Table 9: Test set results from rerunning our annotation submissions with corrected software. Weighted representations use P&N normalization, as in our submitted runs.

Run	TP	FP	FN	Precision	Recall	F-score
dimacsAabsw1	41	21	454	0.6613	0.0828	0.1472
dimacsAg3mh	201	186	294	0.5194	0.4061	0.4558
dimacsAl3w	157	149	338	0.5131	0.3172	0.3920
dimacsAp5w5	33	31	462	0.5156	0.0667	0.1181
dimacsAw20w5	55	42	440	0.5670	0.1111	0.1858

Table 10: Test set results from rerunning our annotation submissions with corrected software. Weighted representations use N&P normalization, unlike the submitted runs.

TOPIC ID: 52
TITLE: Wnt signaling pathway
NEED: Find information on model organ system where Wnt signaling pathway has been studied.
CONTEXT: Need to retrieve literature for any computer modeled organ system that has studied Wnt.

Table 12: A sample topic.

uments to judge for each topic was built by combining the top 75 documents from one run of each of the 27 groups participating in the track. Duplicates were eliminated leaving an average pool size of 976 documents. Judges did not know which systems submitted each document. Each document in the pool was judged as definitely relevant (DR), possibly relevant (PR), or not relevant (NR) to the topic it belongs. Since the task requires binary relevance judgments, DR and PR labeled documents were considered relevant.

7. TEXT RETRIEVAL FOR AD HOC TASK

We used the ASCII text version of the MEDLINE records, provided to the track participants in five separate files.[9] We uncompressed and concatenated these five files to create a single file for the document collection.

For the ad hoc retrieval task, we employed both the MG text retrieval system[10], version 1.2.1, [11], and the full text capability of MySQL database system[11], version 4.0.16. We were able to create a single MG full text index for the entire collection of MEDLINE records. We used MySQL to create an index from each document ID to the position of the document record in the approximately 10GB concatenated file of records. However, an attempt to build a full text index using MySQL failed due to the large size of the collection.

Our retrieval methods therefore first employed MG to retrieve the top-ranked 5000 documents for each topic, and then did MySQL specific processing on this subset. For the initial MG retrieval, we prepared queries by concatenating title words and nouns from need statements. Nouns from need statements were obtained by running a rule-based part-of-speech tagger [1]. Any word tagged with "NN", "NNP", "NNS" and "CD" were included in the query. Then we issued this query to MG as a ranked query to retrieve the top 5000 documents. MG retrieved at least 5000 documents for all topics except test topic 37, for which only 825 documents were retrieved.

We now describe our two variants on post-processing the top 5000 documents:

Method 1: The MEDLINE abstracts corresponding to the retrieved set of MEDLINE articles (5000 articles) were stored in a table in MySQL (*title, abstract, chemical names and MeSH terms* fields) by creating a full index on all four fields. This process is quite fast; it took less than a second to insert the results into a table and create a full text index. Next a boolean type query, specifically designed for MySQL boolean search, was constructed from the topic statement

[9] 2004_TREC_ASCII_MEDLINE_{A-E}.gz
[10] http://www.cs.mu.oz.au/mg/
[11] http://www.mysql.com

Prior	Weight	Gene-Specific							
		No Domain Knowledge				Locus Link			
		Par	5	10	20	Par	5	10	20
G	1	0.280	0.224	0.178	0.220	0.307	0.204	0.174	0.189
G	5	0.288	0.315	0.290	0.321	0.284	0.326	0.259	0.280
G	6	0.281	0.313	0.305	0.253	0.279	0.331	0.191	0.187
L	1	0.441	0.393	0.410	0.390	0.442	0.434	0.439	0.451
L	5	0.372	0.298	0.298	0.342	0.367	0.335	0.343	0.371
L	6	0.369	0.298	0.305	0.346	0.368	0.336	0.346	0.365

Table 11: Gene-specific representation results (F1 Measure), P&N normalization, on the test set.

and the need statement. Note that MySQL can perform boolean full text searches using the IN BOOLEAN MODE modifier. A '+' sign preceeding a word in a query indicates that this word must be present in every result returned. The > operator increases a word's contribution to the relevance score that is assigned to a result. By default, when no '+' is specified, the word is optional, but the rows that contain it will be scored higher. A phrase that is enclosed within double quote characters matches only rows that contain the phrase. Our MySQL queries were of this form: topic title as a phrase preceeded by ">" to increase the score if topic title appears as a phrase, topic title as a subexperession preceeded by ">"and all words preceeded by "+", all title words each preceeded by ">" and all noun words from the need statement. For instance, for the sample topic 52 given in Table 12, the MySQL query became:

> >"wnt signaling pathway" >(+wnt +signaling +pathway) >wnt >signalling >pathway information model organ system wnt pathway.

The boolean query was executed using MySQL and top 1000 results were obtained. MySQL scores the retrieved documents for a boolean query for relevance ranking, and we used its scores for ranking. MySQL returned 1000 documents for all topics except topic 37. Only 822 documents were returned for topic 37.

Method 2: The second method was based on MG ranking and the use of phrases for topic titles. Our goal was to favor documents that contained the topic title as a phrase. For example, for the sample topic 52, a document having a phrase "wnt signalling pathway" should get a better ranking than a document with only "signalling pathway". We retrieved the MEDLINE abstracts corresponding to the retrieved set of articles (top 5000 results) from the initial retrieval step, using our external index. Then we postprocess these MEDLINE abstracts to find the ones which include the topic title as a phrase by matching (ignoring case). We order the results starting from the documents which contain the topic title as a phrase and then the ones which do not include it. In each case, we ranked the results by MG scores.

8. TEXT REPRESENTATION FOR AD HOC TASK

We extracted the title, abstract, chemical names and MeSH terms from the MEDLINE records. (Note that 1,209,243 (26.3%) of the records had no abstract.) Text from chemical names and MeSH terms were processed the same way text from titles and abstracts were processed. We used MG to parse and build indices. All of the stopwords are indexed by MG, however, we eliminated stop words from the queries. We used the stoplist from SMART system as for the text categorization tasks. We did not use stemming. Document parsing performed case-folding and replaced punctuation with whitespace. Tokenization was done by defining a term as a maximal-length contiguous sequence of up to 15 alphanumeric characters. Query parsing was done identically to document parsing.

9. AD HOC TASK RESULTS

9.1 Approach

We constructed queries using the words in title fields, eliminating stop words, and the "noun" words in need sections of the topics. Brill's rule-based part of speech tagger, version 1.14 obtained as part of KeX protein name tagger tool[12], was used [1]. We eliminated duplicate words and stopwords from the queries. The MG system includes support for ranked queries, where similarity is evaluated using the cosine measure. We used the MG system's default TFx-IDF term weighting and cosine similarity measure. First we issued ranked queries to MG. Then, using the top 5000 results, we applied Method 1 and Method 2 for reranking them to obtain top 1000 results as discussed in Section 7. We submitted one run obtained using Method 1 ranking method: *rutgersGAH1*, and another run using Method 2: *rutgersGAH2*.

9.2 Results

The effectiveness measure for the ad hoc task was mean average precision (MAP). Table 13 shows the MAP results for our official runs computed over 50 test topics. Our rutgersGAH1 run performed better. Partipicants were provided the best, median, and worst average precision results for each topic. On the 50 test topics, compared to 37 automatic runs, our rutgersGAH1 run's average precision score was greater than the median 24 times, was less than the median 26 times, and never achieved the best result.

Run	Mean Average Precision
rutgersGAH1	0.1702
rutgersGAH2	0.1303

Table 13: Summary results of our ad hoc runs.

[12]http://www.hgc.ims.u-tokyo.ac.jp/service/tooldoc/KeX/intro.html

Acknowledgements

The work was partially supported under funds provided by the KD-D group for a project at DIMACS on Monitoring Message Streams, funded through National Science Foundation grant EIA-0087022 to Rutgers University. The views expressed in this article are those of the authors, and do not necessarily represent the views of the sponsoring agency.

10. REFERENCES

[1] E. Brill. Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, WA, 1994.

[2] Bradley P. Carlin and Thomas A. Louis. *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall, London, 1996.

[3] Alexander Genkin, David D. Lewis, and David Madigan. Large-scale bayesian logistic regression for text categorization. Technical report, DIMACS, 2004.

[4] William Hersh. Trec 2004 genomics track overview. In *13th Text Retrieval Conference*, 2004. To appear.

[5] David D. Lewis. Evaluating and optimizing autonomous text classification systems. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–254, New York, 1995. Association for Computing Machinery.

[6] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[7] G. Salton, editor. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[8] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[9] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.

[10] Cornelis Joost van Rijsbergen. *Automatic Information Structuring and Retrieval*. PhD thesis, King's College, Cambridge, July 1972.

[11] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, CA, 2 edition, 1999.

[12] H. Yu and E. Agichtein. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 19:340–349, 2003.