Classifying Biomedical Articles by Making Localized Decisions

Thomas Brow[†*] TEBROW@STANFORD.EDU
Burr Settles[‡*] BSETTLES@CS.WISC.EDU
Mark Craven[*‡] CRAVEN@BIOSTAT.WISC.EDU

[†]Stanford University, Stanford, CA 94305 USA

[*]Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53706 USA

[‡]Department of Computer Sciences, University of Wisconsin, Madison, WI 53706 USA

Abstract

We describe a system developed for the Categorization task of the Text Retrieval Conference (TREC) 2005 Genomics track, and experiments we conducted in the process of developing our system. Our research effort for this task explored the hypothesis that more accurate predictions could be achieved by considering only selected passages in the documents being processed. We investigated methods that involve (i) basing classifications on selected passages from test articles, and (ii) adjusting the classifier training process such that certain putatively relevant passages affect the learned model more than other passages. Whereas the first approach was effective at improving predictive accuracy in our experiments, the latter approach was not.

1. Introduction

There are now more than 700 on-line, publicly available databases focusing on some aspect of molecular biology (Bateman, 2005). Most of these databases require a high degree of continual effort by scientists to curate them. For example, most of the model-organism databases, such as the Mouse Genome Informatics (MGI) databases (Eppig et al., 2005), employ a team of PhD-level biologists to read the scientific literature and then manually enter relevant information into the databases. The Categorization task of the 2005 TREC Genomics track was aimed at investigating methods that might help these human curators filter the scientific literature to identify articles relevant to the curation process. In this paper we describe the approaches we investigated in the course developing a system for the Categorization task.

The Categorization task involves making the following decisions. Given the full text of a scientific article, a system should decide whether the article would support curation in each the following four categories: (1) Gene Ontology annotation (The Gene Ontology Consortium, 2000), (2) the Mouse Tumor Biology Database (3) the Gene Expression Database, and (4) the Alleles and Phenotypes category of the Mouse Genome Database. Since the categories are not mutually exclusive, an article may be classified into any number of categories between zero and four. The training set consists of 5,837 articles from *Journal of Biological Chemistry*, *Journal of Cell Biology*, and *Proceedings of the National Academy of Science*. The test set consists of 6,403 articles from the same journals.

Our research effort for this task is based on the conjecture that, for most of these curation decisions, only a fraction of each given document is relevant to the classification. Therefore, we investigate methods that involve (i) basing classifications on selected passages from test articles, and (ii) adjusting the classifier training process such that certain putatively relevant passages affect the learned model more than other passages. We consider as a baseline a method that makes classifications by considering the entirety of each article, and is trained by equally weighting all parts of all training articles.

2. Making Classifications with Selected Paragraphs

In this section, we present our approach and experiments for categorizing articles by having the classifier base its decisions only on selected passages from a test

Appears in *Proceedings of the 2005 Text Retrieval Conference (TREC).*

article. The hypothesis motivating this line of research is that we can attain more accurate classifications by making such "localized" decisions.

2.1. Methods

Our approach for classifying an article using selected paragraphs involves four steps:

1. The article is segmented into paragraphs using SGML tags and regular expressions.

2. The content of each paragraph is described using a bag-of-words representation.

3. A learned statistical model is used to compute the probability that each paragraph belongs to the *positive* class.

4. From a subset of these paragraph-level probabilities, the probability of the *positive* class for the document as a whole is computed.

After segmenting a given article into paragraphs, each paragraph is processed in the following manner. We strip away all remaining SGML tags and replace Unicode entities by ASCII equivalents or representative strings. The resulting plain text is tokenized using a regular expression that allows words to include hyphens and numeric characters. To reduce the size of our vocabulary, we ignore case and remove stopwords. We then represent each paragraph using a bag-of-words representation.

We train our classification models using a maximum entropy method (Nigam et al., 1999). In these models, the probability of class c for document d is defined as:

$$P(c|d) = \frac{1}{Z_d} \exp(\sum_i \lambda_i f_i(d,c))$$

where Z_d is a normalizing factor over all possible labelings of d (to ensure a proper probability in the range [0,1]), and each λ_i is a real-valued weight associated with feature f_i.

The key idea of maximum entropy methods is that the model should prefer the most uniform distribution that satisfies given constraints. In this case, maximum entropy enforces the constraint that the model has the same expected value for each feature as does the training set:

$$\frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} f_i(d, c_d) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \sum_c P(c|d) f_i(d,c).$$

Here, $\mathcal{D}$ represents the training set, c_d is the class of document d, and $P(c|d)$ represents the model's estimation of the conditional probability of class c given document d. For training, we use a quasi-Newton method called L-BFGS that converges to a global optimum. Since the classes of interest are not mutually exclusive, we train a separate model for each class.

We hypothesize that, in any given document, some paragraphs contain more information pertaining to the correct labeling of the document than do others. We investigate this possibility by considering systems that select the most informative paragraphs from full-text articles and make document-level predictions based on these selections. The metric we use for identifying informative paragraphs is the posterior probability for the *positive* class predicted by our maximum entropy models. In particular, we consider an approach that classifies articles by considering the top-n paragraphs, ranked according to the posterior probability of the *positive* class. The estimated probability that an article belongs to the *positive* class, under this approach, is the average probability across these top-n paragraphs.

Our implementation was written in Java and Perl, and included classes from the MALLET library (McCallum, 2002). MALLET implements the maximum entropy classification model as well as several of the preprocessing pipelines we use.

2.2. Empirical Evaluation

We evaluate our approach using four-fold cross-validation within the training set. For each of the classification tasks, we consider the following three approaches:

- A baseline approach that involves training and classifying articles without any regard for paragraph boundaries. Specifically, documents are not segmented in this approach, and document-level class probabilities are computed in the same manner as paragraph-level class probabilities in the other approaches. We refer to this baseline as the Fulltext approach.

- An approach that trains paragraph-level models and classifies a test article using the Top-n procedure described above. For most experiments reported here, $n = 5$.

- An approach that operates similarly, but that sets $n = L$, where L is the number of paragraphs in a given test document. In other words, these models consider the classification decisions made for all paragraphs in a given test document. We refer to

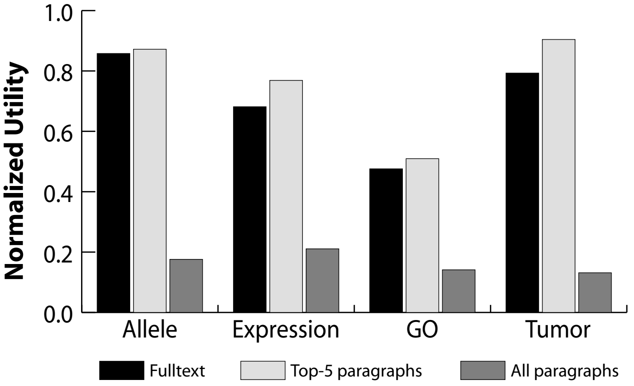

Figure 1. Classifier utilities for models that make classifications based on whole articles (Fulltext and All paragraphs) and selected paragraphs (Top-5 paragraphs). The reported values are average test-set utilities from a cross-validation experiment.

this as the All paragraphs method.

As in the official TREC evaluation, we measure classifier performance by computing *utility*, defined as

$$U_{raw} = (u_r \times TP) + (u_{nr} \times FP),$$

where TP is the count of true positives and FP is the count of false positives. The coefficients u_r and u_{nr} are category-specific weights (or "relative utilities") chosen to account for the varying number of positive instances across categories. These weights are defined as follows:

$$u_{nr} = -1, \qquad u_r = \frac{AN}{AP},$$

where AN and AP are the total counts of actual negative and positive instances, respectively.

Figure 1 shows the measured utility of these three approaches for all four classification tasks. The Top-n classifiers provide better utility than the baseline models for all tasks. This result supports our hypothesis. The results for the All-paragraphs control (Top-n with $n = L$) indicate that success of the Top-n method is due to its focus on a small number of paragraphs, rather than some other aspect of its paragraph-based representation.

3. Making Classifications using Paragraph Distributions

Representing the collection of paragraph-level probabilities by their mean discards information about the distribution of those probabilities. Analysis of the training data suggests that the shape of this distribution might be informative in predicting class labels. Figure 2 compares the average distribution of paragraph-level probabilities in *positive* documents against the average distribution in *negative* documents for the Allele task. The graph on the left illustrates that these average distributions differ in shape as well as mean. The graph on the right shows the plain contrast that appears when we plot distributions for only the top five paragraphs, as considered in the previous section. We hypothesize that using a representation of this entire distribution for a given documents may be more predictive of class labels than the Top-n approach presented in the previous section.

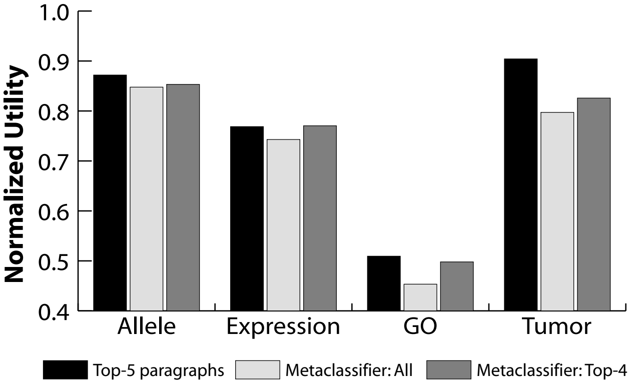

Figure 3. Results of metaclassifier experiments (Metaclassifier:All and Metaclassifier:Top-4) compared to simple mean results from Figure 1 (Top-5 paragraphs).

3.1. Methods

To try to take advantage of this difference in distributions for *positive* and *negative* articles, we train a secondary statistical model to discriminate between the two. The feature vector for this "metaclassifier" is generated by using an integer-valued feature to represent each bin in a discrete representation of the distribution. The value of the feature is the count of paragraphs that have probabilities in the corresponding interval. The metaclassifier models, like the paragraph classification models, are trained using a maximum entropy approach.

3.2. Empirical Evaluation

Figure 3 compares the performance of this strategy, with and without paragraph selection, to those of the baseline and the simple mean approach of the previous section. We consider two variants of the metaclassifier: one that represents the distribution of all paragraphs in a given document, and one that repre-

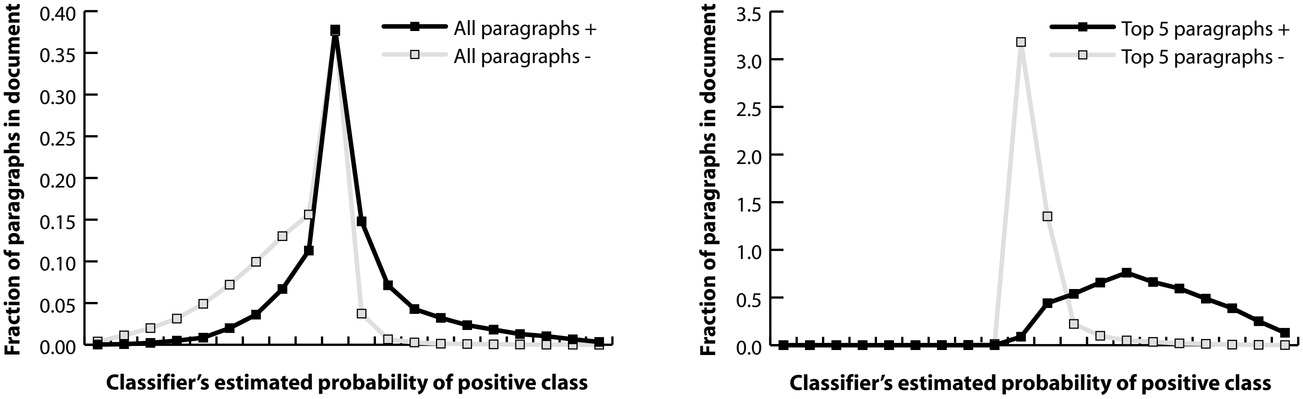

Figure 2. The average distribution of paragraphs in *positive* and *negative* documents for the Allele task, with respect to classifier output. The left side of the figure shows the distributions for all paragraphs and the right side of the figure shows the distributions for Top-5 paragraphs.

sents the distribution of only the top-four paragraphs as determined by a ranking on the predicted posterior probability of the *positive* class. The results in Figure 3 show that the distribution metaclassifiers do not result in higher utilities than the simple Top-n approach for any of the tasks. This result suggests that the metaclassifier models are susceptible to overfitting.

4. Training Classifiers with Selected Paragraphs

So far, we have discussed making localized decisions by focusing the models' attention on important passages during *classification*. Another way of localizing the classifier is to focus the models' attention to important passages during *training*. One way of accomplishing this is by employing an expectation-maximization (EM) algorithm (Dempster et al., 1977), which is an approach to finding likelihood estimates for parameters in probabilistic settings with hidden variables.

4.1. Methods

In our setting, the *hidden variables* represent the extent to which individual paragraphs should be treated as a *positive* instances during training. Our approach employs one hidden variable for each paragraph in a *positive* document. We assume that all paragraphs in a *negative* document really are *negative*, and thus there are no hidden variables for these cases.

In the *E-step*, we use the current model to estimate the probability that each paragraph in a given document is *positive* (i.e., contains text relevant to the document being *positive*). Formally, we compute:

$$z_{ij} = P(c_{ij} = 1 | d_{ij}; \theta^{(t)})$$

where z_{ij} is the hidden variable associated with the jth paragraph in the ith positive document, c_{ij} is the unknown class label of the paragraph (1 for *positive*, 0 for *negative*), d_{ij} represents the text of the paragraph, and $\theta^{(t)}$ represents the model parameters on the tth iteration of the EM procedure.

Occasionally, there may be no paragraphs that appear *positive* for a given training document that is known to be *positive*. To test and correct for this, we sum the model's output for all paragraphs in a document. If the sum is less than some threshold k, we re-normalize weights to sum to k. In other words, we enforce the constraint:

$$\forall_{i \in pos} \sum_j z_{ij} \geq k.$$

The assumption here is that a *positive* document has at least k paragraphs that are relevant to its class. For the experiments reported here, we set $k = 2$.

In the *M-step*, the classifier is re-trained using paragraph instances subject to the newly estimated weights. This entails the following optimization:

$$\theta^{(t+1)} = \arg\max_\theta \sum_{d_i \in pos} \sum_j \big[z_{ij} \log(P(c_{ij} = 1|\theta) P(d_{ij}|c_{ij} = 1; \theta)) + (1 - z_{ij}) \log(P(c_{ij} = 0|\theta) P(d_{ij}|c_{ij} = 0; \theta)) \big] + \sum_{d_i \in neg} \sum_j log(P(c_{ij} = 0|\theta) P(d_{ij}|c_{ij} = 0; \theta)).$$

4.2. Empirical Evaluation

We conduct an experiment in which we evaluate models trained with this EM approach using, as before,

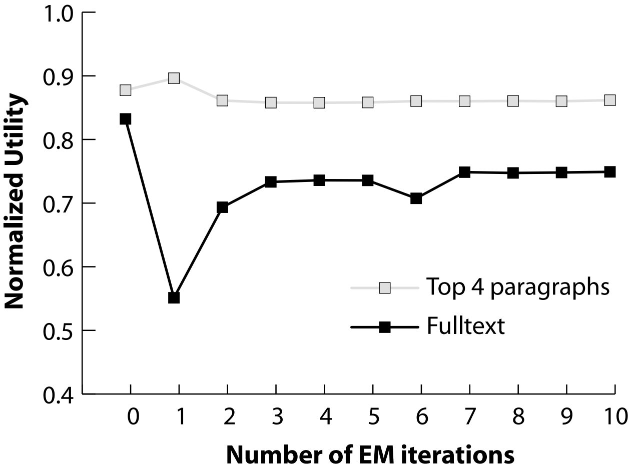

Figure 4. The effects of ten EM iterations on utility measure for the Allele task. The results are similar for the other categories. Classifiers are trained using EM-weighted paragraphs and evaluated against both full-text and Top-n paragraphs.

a four-fold cross-validation methodology. The models are initialized by training with the standard maximum entropy approach described in Section 2. Before each subsequent iteration of EM, the classifiers are applied to the test fold, using both the Fulltext and Top-n classification methods, as described in Section 2. For these experiments, $n = 4$.

Figure 4 shows the classification utility realized by these two classification methods as a function of the number of EM training iterations. Utility generally drops immediately after EM re-weighting begins, and while subsequent iterations show gradual improvement, the models appear to converge before reaching even the initial model's utility. This result indicates that the EM algorithm, as used here, either is not effective at identifying the most relevant paragraphs or that there is no benefit in doing so during training. We also note that the Top-n method of evaluation outperforms the Fulltext method in this context as well.

5. Official TREC Evaluation Results

To generate our final classifications for a given category, we select those documents whose probability for the *positive* label exceeds a fixed threshold. We choose this threshold for each category by averaging the five thresholds that yield the greatest normalized utility from our four-fold cross-validated experiments.

We submitted runs from the Top-n classification approach described in Section 2 as an official run along with the metaclassifier variation described in Section 3 and the baseline Fulltext system. Table 5 shows the results of these methods as well as the minimum, median, and maximum scores from the official task evaluation. These results are consistent with the results of the cross-validation experiments reported in Sections 2 and 3 in that the Top-5 models outperform the Fulltext and Metaclassifier models. The utility achieved by our Top-5 models is close to the median score across all four tasks.

Description	Allele	Expr.	GO	Tumor
Fulltext	0.7434	0.6012	0.4287	0.8160
Metaclassifier	0.7736	0.6548	0.4386	0.7833
Top-5	0.7725	0.7304	0.4572	0.8242
Track min	0.2009	-0.0074	-0.0342	0.0413
Track median	0.7785	0.6548	0.4575	0.7610
Track max	0.8710	0.8711	0.5870	0.9433

Table 1. Normalized utility scores of the three systems for which we submitted runs on official TREC data. Also presented are the minimum, median, and maximum scores for participants in each category.

6. Discussion and Future Work

We investigated three hypotheses in our efforts for the Categorization task of the TREC Genomics track. The first hypothesis, that we could get more accurate classifications by basing classification decisions on selected paragraphs in test articles, was well supported by our experiments. The second hypothesis was that we could achieve more accurate classifications by employing a rich representation of predicted paragraph-level class probabilities. The third hypothesis was that we could learn more accurate models by having the training process put more emphasis on some paragraphs than others. Neither of these latter two hypotheses were supported by our experimental results.

It is encouraging that we were able to successfully select relevant paragraphs using the crude metric of label probabilities assigned by a statistical model, given that the model was trained on un-segmented documents marked only as *positive* or *negative*. In future work we plan to explore an approach in which we consider additional paragraph features, such as location, context and rhetorical role, in deciding on the relevance of the paragraph to the classification task at hand.

We also plan to investigate a *multiple-instance* approach (Dietterich et al., 1997) to the task of training models for this task. The application of this type of approach is motivated by the belief that, many of the paragraphs in *positive* articles should **not** be treated as being representative of the *positive* class. This was the same motivation that prompted our investigation of the EM approach. In contrast the EM method, however, the multiple-instance approach would identify putatively relevant paragraphs in a more supervised manner.

Acknowledgements

This work is supported in part by NIH/NLM grant T15 LM007359 and by NIH/NLM grant R01 LM07050-01.

References

Bateman, A. (2005). Database issue. *Nucleic Acids Research*, *33*.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

Dietterich, T., Lathrop, R., & Lozano-Perez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Eppig, J., Bult, C., Kadin, J., Richardson, J., Blake, J., & the Mouse Genome Database Group (2005). The Mouse Genome Database (MGD): Integrating biology with the genome. *Nucleic Acids Research*, *33*, D471–D475.

McCallum, A. (2002). MALLET: A MAchine Learning for LanguagE Toolkit. http://mallet.cs.umass.edu.

Nigam, K., Lafferty, J., & McCallum, A. (1999). Using maximum entropy for text classification. *Working Notes of the IJCAI Workshop on Machine Learning for Information Filtering*.

The Gene Ontology Consortium (2000). Gene Ontology: Tool for the unification of biology. *Nature Genetics*, *25*, 25–29.