Applying Probabilistic Thematic Clustering for Classification in the TREC 2005 Genomics Track

Z. H. Zheng, S. Brady, A. Garg, H. Shatkay
School of Computing, Queen's University
Kingston, Ontario, Canada
{zhi, 1sb1, 2ag5, shatkay}@cs.queensu.ca

Abstract

Our group participated in the categorization task of the TREC Genomics Track. We introduced and investigated a cluster-based approach for classifying documents. We first clustered the abstracts of the negative training examples based on their term distribution, then built a classifier to distinguish between each cluster and the set of positive examples. The large number of resulting classifiers (a total of 14-19 classifiers per domain) was combined to categorize the test set. We also conducted experiments for cluster-based feature selection; Rather than select features from the whole negative and positive training sets, we selected features from each of the clusters and took the union of these features as the selected features for representing the whole training and test data.

We compared our cluster-based multi-classifier approach against a simple naïve Bayes classification. We also compared the cluster-based feature selection strategy with the commonly used Chi-square-based feature selection.

1. Introduction

Text categorization was one of the two tasks in the TREC 2005 Genomics Track. It was concerned with the classification of articles from four major categories, including *alleles of mutant phenotypes*, *embryologic gene expression*, *tumor biology*, and *gene ontology (GO) annotation*. The task was to identify documents that are relevant to these categories, using a classifier trained on the labeled data.

The full text articles for both training and test set were given, although we used only title, abstract and MeSH terms in our experiments. All articles in the training set were published in 2002, while articles in the test set were published in 2003. Therefore, both the training and test examples are not selected uniformly at random. The text categorization task provides the crosswalk files for both the training and test data as well. The corresponding PubMed ID (PMID) of each article is given in these files.

The evaluation measure for the text categorization task is the normalized utility score U_{norm} for each category, which is defined as follows:

$$U_{norm} = U_{raw} / U_{max} ,$$

where:

- U_{raw} is the raw utility score, defined as the difference between the weighted true positives (TP) and the false positives (FP) :
  $$U_{raw} = (u_r \times TP) - FP ,$$
  where u_r is the relative utility of a relevant document;
- U_{max} is the best possible utility score, defined as the sum of the weighted TP and the false negatives (FN):
  $$U_{\max} = u_r \times (TP + FN) .$$

This evaluation measure penalizes misclassification on the relevant documents (false negatives) u_r times more than misclassification of the irrelevant documents (false positives). It therefore favors high recall and compromises precision.

In our experiments, we considered the categorization task as four separate binary classification tasks. We investigated a new, cluster-based approach for classifying documents into these four categories. The observation underlying our approach is that the distribution of negative vs. positive examples, in both the training and the test data, is biased. The negative examples are abundant and may discuss a variety of topics, while the number of positive examples is small and their topic is usually focused. Hence, we first separated the abstracts in the negative training set into multiple clusters, based on their term distribution, by applying a probabilistic theme generation algorithm [Shatkay and Wilbur 00, Shatkay *et al* 00]. A classifier was then built to distinguish between each cluster and the set of positive examples. The resulting classifiers were combined into a single classifier. We applied the combined classifier to the test set.

We also tried to address the feature selection issue, noted as "conceptual drift" in last year's TREC [Cohen *et al*. 04], using cluster-based feature selection. Our experimental results suggest an improvement by selecting features that distinguish each individual cluster from the positive data set.

The rest of this paper is organized as follows: In the next section we introduce our methods, the theme generation technique, the cluster-based classification and the feature selection approach. Section 3 discusses our experimental results and a brief analysis is given. Finally, we conclude our work and suggest some future directions.

2. Methods

We approached the text categorization task as four separate binary classification tasks, one for each of the four categories. For each task, the irrelevant documents were viewed as belonging to the *irrelevant* category. Our experiments mainly focused on the cluster-based classification approach, which will be introduced in Section 2.2. As a baseline for comparison and as a basic building block in our own categorization we use a naïve Bayes classifier. This is because the naïve Bayes technique is simple and still effective for text categorization. The version we employed is the one introduced by John and Langley [John and Langley, 95], and implemented as WEKA's NaiveBayes utility [Witten and Frank 05]. This version allows both discrete and numerical features. When the classification model is built on the whole training set, without using clusters, we refer to it as the *single-model classification*.

We applied a cost sensitive approach [Breiman *et al*. 84] to reflect the different penalties on different types of misclassification, while using the normalized utility scoring. This was done by re-weighting the positive examples in the training set with a weight factor W_p. That is, each positive example is considered as W_p positive examples when learning the classifier. W_p is calculated using the formula: $W_p = CB \times u_r \times (neg\#/\ pos\#)$, where:

- *CB* is the Cost Bias for adjusting the weight factor, as explained below;
- *neg#* is the number of negative examples in the training data;
- *pos#* is the number of positive examples in the training data.

This formula takes into account the ratio between positive and negative data (*neg#/ pos#*), and the relative utility of a relevant document, u_r. We introduce the variable *CB* into the formula so that the weight factor W_p becomes adjustable. In our experiments, *CB* was chosen from the range [0, 5.0]. We adjust *CB* to find the model that produces the highest utility on the training data. The n-fold cross validation was used for the single-model classification. The number of folds (n) was set to 10 for the *alleles* and *GO* classification, and to 3 for the *embryologic gene expression* and *tumor* classification.

2.1 Feature Generation and Weighting

While the Genomics Track provided the full text for both the training and test set, we generated the features for all documents from their Medline records, including only the titles, abstracts, and MeSH terms. Using the PMIDs provided in the files *train.crosswalk.txt* and *test.crosswalk.txt*, the related Medline records were obtained directly from the Medline database. All XML tags in the Medline records were removed before the feature generation. Unlike the traditional "bag-of-word" model, by which typically only unigrams are extracted as features, we extracted both single words and 2-gram terms as features. Stop-words were removed and Porter stemming was applied to each word [Porter 80]. While a variety of feature weighting schemes exist, we used the simple binary weights: *1* for present and *0* for absent term. This is because our

early preparatory experiments indicated that the binary weighting scheme outperformed the TF*IDF scheme for learning a naïve Bayes classifier.

2.2 Cluster-Based Classification

In the cluster-based approach, we first cluster the negative training examples into subgroups, based on their term distribution. This is done by applying a probabilistic theme generation algorithm, which is discussed in the next subsection. The generated clusters are referred to as *themes*. We then built a classifier to distinguish between each of the themes and the set of positive examples. Finally, we combined the large number of resulting classifiers to categorize the test set.

2.2.1 Probabilistic Theme Generation

The probabilistic theme generation algorithm that we used is based on the one introduced a few years ago for retrieving Medline documents similar to a given a query document [Shatkay and Wilbur 00], and for finding functional relationship among genes [Shatkay *et al* 00]. A theme T is a set of documents that are likely to discuss a common topic. It is characterized by a set of term distributions. Given a database *DB*, the model R for a theme T consists of five components: $R = \{P_d, \{p_i\}, \{q_i\}, \{DB_i\}, \{\lambda_i\}\}$ where:

- P_d is the prior probability of any document $d \in DB$ to be in the theme T, $P_d = \Pr(d \in T)$. It is assumed fixed by the application.
- p_i is the probability that the term t_i occurs in a document d where d is an on-theme document, $\Pr(t_i \in d \mid d \in T)$.
- q_i is the probability that the term t_i occurs in a document d when d is *not* in the theme T, $\Pr(t_i \in d \mid d \notin T)$.
- DB_i is the probability that the term t_i occurs in any document in the database, $\Pr(t_i \in d \mid d \in DB)$. $\{DB_i\}$ can be easily estimated from the documents in the database.
- λ_i is the probability that the term t_i is generated according to the general database distribution DB_i. It is used as a mixture distribution between DB_i and the theme-specific parameters p_i and q_i.

The main tasks in generating a theme, based on an example document d, is to simultaneously estimate the model parameters p_i, q_i and λ_i, while finding other documents in the database that are likely to have been generated by the same model. The latter set of documents is called the *theme*, while the terms with high score $Log(p_i / q_i)$ are viewed as its characteristic terms. An *EM* (Expectation Maximization) algorithm is used to find the most likely model R for a given example document d and a database *DB*. For details see earlier work [Shatkay and Wilbur 00, Shatkay *et al* 00].

2.2.2 Building Clusters and Classifiers

A schematic overview of the cluster-based approach is depicted in Figure 1. This approach originates from the observation that the positive training examples are few and typically focused on one topic, while the negative examples are many and may discuss a wide variety of topics. Taking the embryologic gene expression category as an example, the number of the negative examples in the training data is 5,756, while the number of the positive examples is only 81 (less than 2% of the total data). Looking at the negative examples, we observe that different examples discuss different topics. For instance, the negative example with PMID 12221087 is about the retinoblastoma tumor, while the document with PMID 12052828 discusses the role of serine proteases in erythrocyte invasion by merozoites of the malaria parasite. In this situation, the distribution of the whole negative examples becomes flat and non-specific. Therefore, trying to characterize all the negative examples with one single-model does not provide a clear distinction between the positive examples and the negative ones.

To address this problem, we apply the probabilistic theme generation algorithm to the negative training examples, sub-grouping them into disjoint thematic clusters. We expect that the negative examples within each thematic cluster are similar to each other, and that the term distribution within a cluster is more specific and well-defined than that of the whole negative example set. We therefore expect that it will be easier to determine, given a test example, with respect to each cluster separately (as opposed to with respect

to the whole negative set), whether the example is *inside* the cluster or *outside* the cluster. Accordingly, we train as many classifiers as there are thematic clusters.

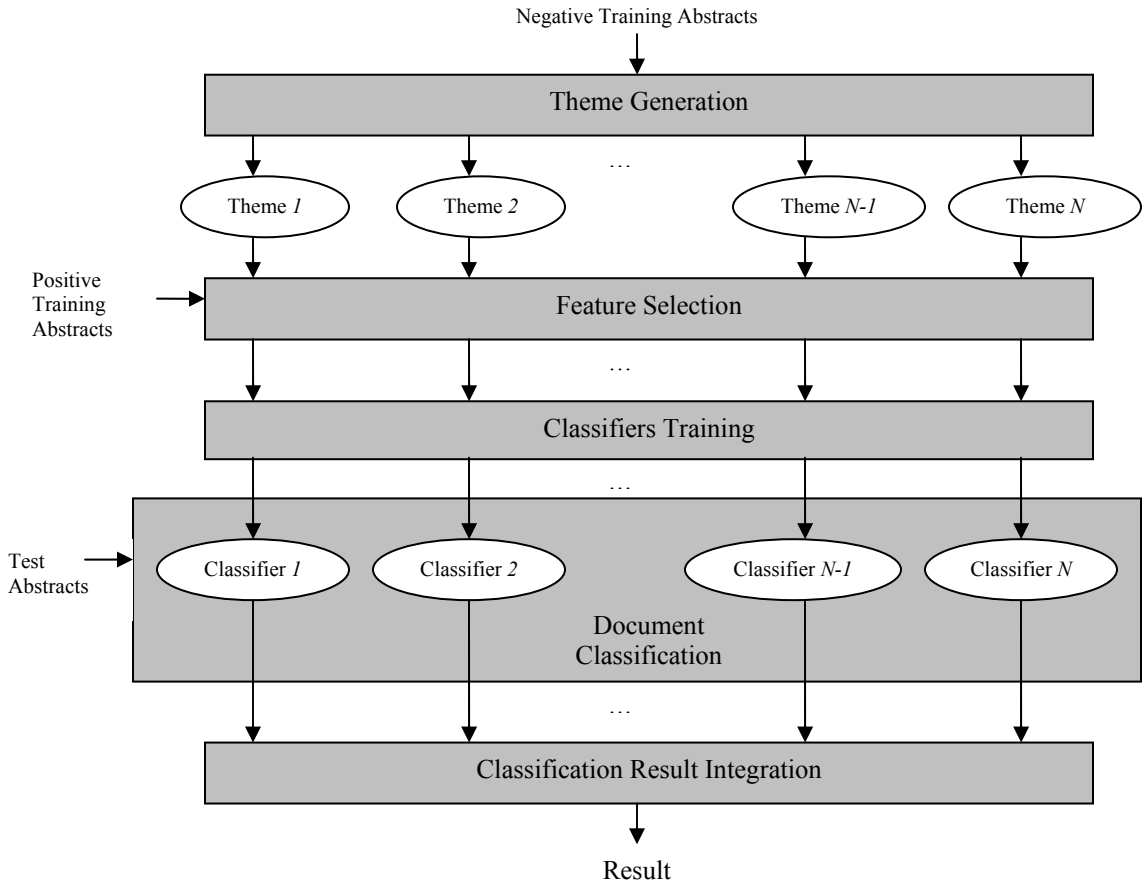

Figure 1. The Cluster-based Classification Process

As indicated in the Section 2.2.1, a theme is built around a single document. In our experiments, we randomly select an abstract from the negative data set as the representative document for a theme. Since we know that none of the positive examples should be a member of any cluster generated from the negative examples, we initialize the off-theme parameter q_i of the model to reflect this bias. Whenever a term t_i occurs in a positive example(s), q_i is initialized by the number of occurrences of term t_i in the *positive* data set divided by the total number of occurrences of any term in the *positive* data set. Moreover, we also impose the restriction that no abstract is associated with more than one theme. This guarantees that all thematic clusters are disjoint, although the original application [Shatkay and Wilbur 00, Shatkay et al 00] allowed overlap among themes.

To train a classifier for a particular cluster, we use the negative examples in a single cluster as the negative training set while *all* the positive examples are used as the positive training set. Note that each classifier is meant to determine whether an unseen example is *inside* or *outside* a specific thematic cluster. An example that is outside the cluster is *not* automatically placed inside the positive category, as it might be a member of another cluster.

Once all the classifiers are built from each thematic cluster, we use them to determine the membership of each example from the test data set. The classification results for an abstract *d* form an N-dimensional vector $\langle T_d^1,...T_d^N \rangle$, where *N* is the number of themes, and

$$T_d^i = \begin{cases} 1 & if\ d \in theme\ i \\ 0 & otherwise \end{cases}$$

We consider an abstract to be inside the positive category *if and only if* it is excluded from *all* thematic clusters. Formally, let $f_c : DB \to \{true, false\}$ be a function mapping a document *d* in the database *DB* to the values, *true* or *false*, depending on whether *d* is a member of the category *c*. Then

$$f_c(d) = \begin{cases} true, & if\ T_d^i = 0\ for\ all\ i, 1 \le i \le N, \\ false, & otherwise \end{cases} .$$

Clearly, our approach takes an alternative way to distinguish the positive documents from the negative ones, compared with the single-model classifier. Rather than directly test for membership of a document in the positive or the whole negative category, we separately test membership in each negative thematic cluster. A document is inferred to be a member of the positive category *if and only if* it is excluded from *all* the negative thematic clusters. Moreover, a document must demonstrate similarity to at least one specific negative cluster in order to be considered negative.

2.3 Cluster-Based Feature Selection

The Chi-square test for feature selection was applied to the training set or its subsets, depending on the chosen classification method. Supervised feature selection is typically performed by considering the *whole* set of negative training samples vs. the whole set of positive ones; analyzing the correlation between the features and the categories, features that have significantly different distributions under different categories are selected. The assumption underlying such feature selection schemes is that the training data is a good representative of the true data, and specifically of the test data. This assumption, as reported by Cohen *et al*. [Cohen *et al*. 04], and Zhang and Lee [Zhang and Lee 04], does not necessarily hold in the biomedical literature used in TREC, where the training data consist of earlier publications than the test data. Cohen *et al*. refer to the change in feature distribution over time as *conceptual drift*. That is, the topics that are discussed in the literature, the relative number of documents discussing these topics, and possibly the jargon used to discuss them, may all vary over time. Therefore, features that are selected from the whole training data may not be as useful for categorizing the "conceptually drifted" test data.

To try and overcome this drift, we have devised a cluster-based feature selection approach. We cluster the negative training set as described in Section 2.2. Then, rather than select features using the Chi-square test applied to the whole negative vs. positive training set, we apply the Chi-square test individually to each negative cluster (vs. the positive set). We then take the union of these features as the selected features and use them to represent the training and test data. The assumption is that the term distribution within each theme-cluster will remain consistent over time, while only the number of samples in each particular theme may change.

3. Experiments

We have applied both the single naïve Bayes classification and the cluster-based classification to all the four categories included in the track. In both approaches, we trained the naïve Bayes classifiers using the cost sensitive scheme. For the cluster-based classification, numerous clusters have been built. Table 1 shows the number of clusters for each category. In addition to the official runs, we describe some of the unofficial runs.

Category	Allele	Embryologic Gene Expression	GO	Tumor
# of clusters	16	15	19	18

Table 1: The number of clusters generated from *the negative examples* in each category

3.1 Official Runs

The results of our official runs are presented in Table 2. In the columns "Classification Method" and "Feature Selection", we use the term *Single Model* to denote that the classification or the feature selection is built on the whole training data, and the term *Cluster-Based* to denote the application of the cluster-based method to the classification or to the feature selection. The best results we achieved in the official runs, in terms of the normalized utility score, for each of the categories *allele*, *embryologic gene expression*, *GO*, and *tumor,* are 0.7760, 0.5563, 0.3763 and 0.7439, respectively. The recall is higher than the corresponding precision in all of our runs. This is partly because we have introduced a high cost to false negatives to reflect the biased penalty applied. We also note that in particular the cluster-based approach is expected to favor recall, as it requires examples to fit into specific negative-cluster models in order to be classified as negative. The highest recall we got is of 1.0 for tumor categorization in the run tQUT10, in which the cluster-based classification is applied. Overfitting is observed in both official and unofficial runs. For instance, the single-model run gQUNB12 resulted in a utility score of 0.696 for the training data, but only 0.346 for the test data. This might be partially caused by the different distributions of the training and test set, especially when the cost sensitive approach is applied.

Category	Run	Classification Method	Terms	Feature Selection	*CB*	Precision	Recall	F-score	Normalized Utility
Allele	aQUNB8	Single Model	4487	Cluster-Based	0.54	0.3182	0.8464	0.4626	0.7397
	aQUT11	Cluster-Based	10533	Single Model	1	0.3785	0.7741	0.5084	0.6993
	aQUT14	Cluster-Based	4487	Cluster-Based	2	0.3582	0.8675	0.5070	0.7760
Embryologic Gene Expression	eQUNB11	Single Model	2228	Cluster-Based	1	0.1086	0.6381	0.1856	0.5563
	eQUNB19	Single Model	5155	Single Model	1.17	0.1132	0.4571	0.1815	0.4012
	eQUT18	Cluster-Based	5155	Cluster-Based	0.10	0.0967	0.5238	0.1632	0.4473
GO	gQUNB12	Single Model	13414	Single Model	1	0.1603	0.6602	0.2580	0.3459
	gQUNB15	Single Model	4872	Cluster-Based	0.55	0.2102	0.5676	0.3067	0.3763
	gQUT22	Cluster-Based	11417	Single Model	4	0.1811	0.6158	0.2799	0.3628
Tumor	tQUNB3	Single Model	3058	Single Model	1	0.0244	0.9000	0.0474	0.7439
	tQUT10	Cluster-Based	3058	Single Model	0.02	0.0132	1.0000	0.0260	0.6758
	tQUT14	Cluster-Based	1500	Cluster-Based	1	0.3095	0.6500	0.4194	0.6437

Table 2: The official runs

Note that the official runs from different classification methods are not directly comparable to each other because of their different feature selection schemes or different parameter settings. We have conducted some unofficial runs for a more complete comparison of different classification methods and different feature selection schemes.

3.2 Unofficial Runs

In our unofficial runs, the best utility scores we got for the *allele*, *embryologic gene expression*, *GO*, and *tumor* categories are 0.7505, 0.6101, 0.4189 and 0.8582, respectively. These results are much better than those obtained in our official runs, except the one for allele categorization, which is 0.02 lower than the utility of our best official run. These results were obtained by either the cluster-based classification or by the single-model classification with the cluster-based feature selection. We will use bold to highlight these results in Table 3 and Table 4.

We performed several runs to directly compare the performance of the single-model classification and the cluster-based classification. Table 3 shows the performance of the two classification methods under the

same parameter setting. We set the same Chi-square threshold for both methods on the same classification task. The thresholds are 7 for *allele* and *GO*, and 5 for *embryologic gene expression* and *tumor*. For each category, both methods – the single-model classification and the cluster-based classification – use the same value for *CB*.

Category	Classification Method	*CB*	*TP*	*FP*	*FN*	Precision	Recall	F-score	Normalized Utility
Allele	Single Model	2	160	100	172	0.6154	0.4819	0.5405	0.4642
	Cluster-Based		288	516	44	0.3582	0.8675	0.5070	0.7760
Embryologic Gene Expression	Single Model	2	35	71	70	0.3302	0.3333	0.3318	0.3228
	Cluster-Based		84	1276	21	0.0618	0.8000	0.1147	**0.6101**
GO	Single Model	4	188	340	330	0.3561	0.3629	0.3595	0.3033
	Cluster-Based		379	1782	139	0.1754	0.7317	0.2829	**0.4189**
Tumor	Single Model	1	6	4	14	0.6000	0.3000	0.4000	0.2991
	Cluster-Based		13	29	7	0.3095	0.6500	0.4194	0.6437

Table 3: The performance of the single-model and of the cluster-based classification

Our results show that, all else being equal, the cluster-based classification outperforms the single-model classification on all categorization tasks in terms of normalized utility and recall. Using the cluster-based classification, there is about 56% improvement in normalized utility. However, the precision resulting from the cluster-based approach is significantly lower than that of the single-model classification on all runs. This is because the cluster-based classification combines multiple cluster-based classifiers, each built on a separate cluster, where all these individual classifiers are biased towards avoiding false negatives. Recall that each individual cluster-based classifier is trained with respect to a *single cluster* of negative examples. Therefore, each such classifier has a narrow negative category associated with it, and all the documents falling outside this category are initially viewed as *positive* with respect to this specific classifier. This means that each cluster-based classifier initially produces a large number of false positives and relatively few false negatives. Taking our experimental result on the *allele* classification as an example, using sixteen clusters, the average number of the false negatives is 4, while the average number of the false positives is 4,875 (note that this is *before* the classification results are combined – this phase is not shown in the tables). As discussed in Section 2.2.2, the results from the individual classifiers are combined such that a document is labeled as positive if and only if it is labeled as positive by *all* of the individual cluster-based classifiers. At this stage, we are left with 288 true positives and 516 false positives. While the number of the false positives has been significantly reduced when the results from the individual classifiers are combined, this number is still larger than the one resulting from the single-model classification, which is 172. Thus, many false positives still remain even after the strict positive selection induced by the combination of the individual classifiers. Trying to improve the precision under the cluster-based framework is our immediate next step.

Category	Feature Selection	Feature Number	Normalized Utility		
			Training Data	Test Data	Difference
Allele	Single Model	4487	0.8446	0.6588	0.1858
	Cluster-based		0.8422	**0.7505**	0.0917
Embryologic Gene Expression	Single Model	2228	0.8937	0.0467	0.8470
	Cluster-based		0.8490	0.5563	0.2927
GO	Single Model	4872	0.6848	0.1909	0.4939
	Cluster-based		0.5085	0.3508	0.1577
Tumor	Single Model	1500	0.9768	0.1496	0.8272
	Cluster-based		0.9030	**0.8582**	0.0448

Table 4: The effect of the cluster-based feature selection.

As discussed in Section 2.3, a side-effect of the cluster-based classification is an alternative method for feature selection. We have run some preliminary experiments to evaluate the performance of this method. We first collected the selected features from all the clusters that were constructed as part of the cluster-based classification. The union of the feature sets was used as features for both the training and test data. For comparison, we applied the Chi-square test using the training data set, with no clustering, to select the same number of features. Keeping all parameters identical, single-models were constructed separately based on these two feature sets. The experimental results are presented in Table 4. All experiments listed in Table 4 have *CB* value of 1. The *Training Data* column and the *Test Data* column show the normalized utility on the training set and the test set, respectively. The *Difference* column shows the difference in normalized utility between the training and test data. The *Difference* is used to evaluate the variability in the model performance between the training and the test sets. The smaller the difference is, the more consistent performance a model provides. We can clearly see that all models have better performance on the training set than on the test set. However, the models that are constructed based on the cluster-based feature selection perform more consistently than the models that are constructed based on the single-model feature selection. Our results suggest that there is less drift between test and training data when the cluster-based feature selection is used.

4. Conclusion

We approached the categorization task in the TREC 2005 Genomics Track as a set of separate binary classification tasks. A common characteristic of both training and test data is their relative abundance of negative examples, which typically leads to low recall. We have investigated a cluster-based classification approach, which aims to distinguish among subsets within the large set of negative examples. A comparison was made between this approach and the single-model approach. Using a basic text processing method, when all else is equal, the cluster-based classification approach outperforms the single-model approach in our experiments. We have also explored the effect of the cluster-based feature selection. Our primary finding is that the utility gap between the training data and the test data decreases when the cluster-based feature selection is applied, which suggests that the cluster-based feature selection may address the issue of "conceptual drift". We recognize the need for further improvement on the classification performance, and plan to experiment with more advanced baseline methods and feature selection in the future.

Acknowledgements

The work was partially supported by Queen's ARC award #380-265, and NSERC Discovery Grant #298292-04, awarded to HS, and NSERC summer student grants awarded to SB and AG.

References

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. 1984, "Classification and Regression Trees", *Wadsworth*, Belmont, CA

Cohen, AM. *et al*. 2004, "Feature generation, feature selection, classifiers, and conceptual drift for biomedical document triage", *Proc. of TREC 2004. Gaithersburg, MD: National Institute of Standards and technology.*

John, G. H. and Langley, P. 1995, "Estimating continuous distributions in Bayesian classifiers", *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. 338-345

Porter, MF. 1980, "An algorithm for suffix stripping", *Program*, 14: 130-137

Shatkay, H., Edwards, S., Wilbur, J. and M. Boguski, 2000, "Genes, Themes and Microarrays", *ISMB 2000,* pp. 317-328.

Shatkay, H. and Wilbur, J. 2000, "Finding Themes in Medline Documents: Statistical Similarity Search", *IEEE Conference on Advances in Digital Libraries 2000,* pp. 183-192.

Witten, I. H. and Frank, E. 2005, "Data Mining: Practical machine learning tools and techniques, 2nd Edition", *Morgan Kaufmann*, San Francisco.

Zhang, D. and Lee, WS. 2004, "Experimence of using SVM for the triage task in TREC2004 genomics track", *Proc. of TREC 2004. Gaithersburg, MD: National Institute of Standards and technology.*