An Evaluation of Statistical Approaches to Text Categorization

Yiming Yang
yiming@cs.cmu.edu

April 10, 1997

CMU-CS-97-127

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper is a comparative study of text categorization methods. Fourteen methods are investigated, based on previously published results and newly obtained results from additional experiments. Corpus biases in commonly used document collections are examined using the performance of three classifiers. Problems in previously published experiments are analyzed, and the results of flawed experiments are excluded from the cross-method evaluation. As a result, eleven out of the fourteen methods are remained. A k-nearest neighbor (kNN) classifier was chosen for the performance baseline on several collections; on each collection, the performance scores of other methods were normalized using the score of kNN. This provides a common basis for a global observation on methods whose results are only available on individual collections. Widrow-Hoff, k-nearest neighbor, neural networks and the Linear Least Squares Fit mapping are the top-performing classifiers, while the Rocchio approaches had relatively poor results compared to the other learning methods. KNN is the only learning method that has scaled to the full domain of MEDLINE categories, showing a graceful behavior when the target space grows from the level of one hundred categories to a level of tens of thousands.

This research was supported in part by NIH grant LM-05714 and by NSF grant IRI9314992.

Keywords: text categorization, statistical learning, comparative study.

1 Introduction

Text categorization is the problem of assigning predefined categories to free text documents. A growing number of statistical learning methods have been applied to this problem in recent years, including regression models[5, 18], nearest neighbor classifiers[3, 19], Bayes belief networks [14, 9], decision trees[5, 9, 11], rule learning algorithms[1, 15, 12], neural networks[15] and inductive learning techniques[2, 8]. With more and more methods available, cross-method evaluation becomes increasingly important. However, without an unified methodology of empirical validation, an objective comparison is difficult.

The most serious problem is the lack of standard data collections. Even when a shared collection is chosen, there are still many ways to introduce inconsistency. For example, the commonly used Reuters newswire corpus[6] has at least four different versions, depending on how the training/test sets were divided, and what categories are included or excluded in the evaluation. Lewis and Ringuette used this corpus to evaluate a decision tree approach and a naive Bayes classifier, where they included a large portion of unlabelled documents (47% in the training set, and 58% in the test set) [9]. It is not clear whether these unlabelled documents are all negative instances of the categories in consideration, or that they are unlabelled simply as an oversight. Apte et al. run a rule learning algorithm, SWAP-1, on the same set of documents after removing the unlabelled documents[1]. They observed an 12-14% improvement of SWAP-1 over the results in Lewis&Ringuette's experiments, and concluded that SWAP-1 can often substantially improve results over decision trees, and that "text classification has a number of characteristics that make optimized rule induction particularly suitable." This would be a significant finding if the same data were used in the two experiments. However, given that 58% of the test documents were removed from the original set, it is questionable whether the observed difference came from the change in the data, or from the difference in the methods. An analysis later in Sections 3 and 5 will further clarify the point: the inclusion or exclusion of unlabelled documents could have a significant impact to the results; ignoring this issue makes an evaluation problematic.

It would be ideal if a universal test collection were shared by all the text categorization researchers, or if a controlled evaluation of a wide range of categorization methods were conducted, similar to the Text Retrieval Conference for document retrieval[4]. The reality, however, is still far from the ideal. Cross-method comparisons have often been attempted but only for two or three methods. The small scale of these experiments could lead to overly general statements based on insufficient observations at one extreme, or the inability to state significant differences at the other extreme. A solution for these problems is to integrate the available results of categorization methods into a global evaluation, by carefully analyzing the test conditions in different experiments, and by establishing a common basis for cross-collection and cross-experiment integration. This paper reports on an effort in this direction.

Section 2 outlines the fourteen methods being investigated. Section 3 analyzes the collection differences in commonly used corpora, using three classifiers to examine to what degree a difference in conditions effects the evaluation of a classifier. Section 4 defines a variety of performance measures in use and addresses the equivalence and comparability between them. Section 5 reports on new evaluations, and compares them with previously published results. The performance of a baseline classifier on multiple data collections is used as a reference point for a cross-collection observation. Section 6 concludes the findings.

2 Categorization Methods

The intention here is to integrate available results from individual experiments into a global evaluation. Two commonly used corpora, the Reuters news story collection[9] and the OHSUMED bibliographical document collection[7] are chosen for this purpose. Fourteen categorization methods are investigated, including eleven methods which were previously evaluated using these corpora, and three methods which were newly evaluated by this author. Not all of the results are directly comparable because different versions or subsets of these corpora were used. These methods are outlined below; the data sets and the result comparability will be analyzed in the next section.

1. CONSTRUE, an expert system consisting of manually developed categorization rules for Reuters news stories[6].

2. Decision tree (DTree) algorithms for classification[9, 11].

3. A naive Bayes model (NaiveBayes) for classification where word independence is assumed in category prediction[9, 10].

Table 1. Data collections examination using WORD, kNN and LLSF in category ranking

Corpus	Set	UniqCate	TrainDoc	TestDos	(labelled)	WORD	kNN	LLSF
Reuters	CONSTRUE*	182	21,450	723	(80%)	.28	.80	-
	CONSTRUE.2	182	14,346	575	(100%)	.35	.85	-
	Lewis*	113	14,704	6,746	(42%)	.10	.84	-
	Lewis.2	113	7,789	3,309	(100%)	.21	.93	.92
	Apte	93	7,789	3,309	(100%)	.21	.93	.92
	PARC	93	9,610	3,662	(100%)	.21	.91	.91
OHSUMED	full range	14,321	183,229	50,216	(100%)	.16	.52	-
	HD big**	49	183,229	50,216	(100%)	-	-	-
	HD small***	28	183,229	50,216	(100%)	-	-	-

* Unlabelled documents are included.

** Heart Diseases (a sub-domain) Categories only, with a training-set category frequency of at least 75.

*** Heart Diseases (a sub-domain) Categories only, with a training-set category frequency between 15 to 74.

4. SWAP-1, an inductive learning algorithm for classification using rules in Disjunctive Normal Form (DNF)[1].
5. A neural network approach (NNets) to classification[15].
6. CHARADE, a DNF rule learning system for classification by I. Moulinier[12].
7. RIPPER, a DNF rule learning system for classification by W. Cohen[2].
8. Rocchio, a vector space model for classification where a training set of documents are used to construct a prototype vector for each category, and category ranking given a document is based on a similarity comparison between the document vector and the category vectors [8].
9. An exponentiated gradient (EG) inductive learning algorithm which approximates a least squares fit [8].
10. The Widrow-Hoff (WH) inductive learning algorithm which approximates a least squares fit[8].
11. Sleeping Experts (EXPERTS), an inductive learning system using n-gram phrases in classification [2].
12. LLSF, a linear least squares fit (LLSF) approach to classification [18]. A single regression model is used for ranking multiple categories given a test document. The input variables in the model are unique terms (words or phrases) in the training documents, and the output variables are unique categories of the training documents.
13. kNN, a k-nearest neighbor classifier[16]. Given an arbitrary input document, the system ranks its nearest neighbors among training documents, and uses the categories of the k top-ranking neighbors to predict the categories of the input document. The similarity score of each neighbor document is used as the weight of its categories, and the sum of category weights over the k nearest neighbors are used for category ranking.
14. A simple, non-learning method which ranks categories for a document based on word matching (WORD) between the document and category names. The conventional Vector Space Model is used for representing documents and category names (each name is treated as a bag of words), and the SMART system [13] is used as the search engine.

3 Collection Analysis

3.1 Two corpora

The Reuters corpus, a collection of newswire stories from 1987 to 1991, is commonly used for text categorization research, starting from an early evaluation of the CONSTRUE expert system [6, 9, 1, 15, 12, 2] [1]. This collection is

[1] A newly refined version named Reuters-21578 is available through Lewis' home page http://www.research.att.com/~lewis.

split into training and test sets when used to evaluate various learning systems. However, the split is not the same in different studies. Also, various choices were made for the inclusion and exclusion of some categories in an evaluation, as described in the next section.

The OHSUMED corpus, developed by William Hersh and colleagues at the Oregon Health Sciences University, is a subset of the documents in the MEDLINE database[2]. It consists of 348,566 references from 270 medical journals from the years 1987 to 1991. All of the references have titles, but only 233,445 of them have abstracts. We refer to the title plus abstract as a document. The documents were manually indexed using subject categories (Medical Subject Headings, or MeSH; about 18,000 categories defined) in the National Library of Medicine. The OHSUMED collection has been used with the full range of categories (14,321 MeSH categories actually occurred) in some experiments[17], or with a subset of categories in the heart disease sub-domain (HD, 119 categories) in other experiments[8].

3.2 Different versions

Table 1 lists the different versions or subsets of Reuters and OHSUMED. Each is referred as a "set" or "collection", and labelled for reference. To examine the collection differences from a text categorization point of view, three classifiers (WORD, kNN and LLSF) were applied to these collections. The assumption is that if two collections are statistically homogeneous, then the results of a classifier on these collections should not differ too much. Inversely, if a dramatic performance change is observed between collections, then this would indicate a need for further analysis. Since the behavior of a single classifier may lead to biased conclusions, the multiple and fundamentally different classifiers were used instead. All the systems produces a ranked list of candidate categories given a document. The conventional 11-point average precision[13] was used to measure the goodness of category ranking. WORD and kNN were tested on all the collections, while LLSF was only tested on the smaller collections due to computational limitations. The HD sets were examined together with the OHSUMED superset instead of being examined separately.

Several observations emerge from Table 1:

1) Homogeneous collections. The Apte set, the PARC set and the Lewis.2 of the Reuters documents are relatively homogeneous, evident from the similar performance of WORD, kNN and LLSF on these sets. The Lewis.2 is derived (by this author) from the original Lewis set by removing the unlabelled documents. The Apte set is obtained by further restricting the categories to have a training set frequency of at least two. In both sets, a continuous chunk of documents (the early ones) are used for training, and the remaining chunk of documents (the later ones) are used for testing. The PARC set is drawn from the CONSTRUE set by eliminating the unlabelled documents and some rare categories[15]. Instead of taking continuous chunks of documents for training and testing, it uses a different partition. The collection is sliced into many subsets using non-overlapping time windows. The odd subsets are used for training, and the even subsets are used for testing. The differences between the PARC set, the Apte set and the Lewis.2 set do not seem to have a significant impact on the performance of the classifiers.

2) An outlier collection. The CONSTRUE collection has an unusual test set. The training set contains all the documents in the Lewis set, Apte set or PARC set, and therefore should be statistically similar. The test set contains only 723 documents which are not included in the other sets. The performance of WORD and kNN on this set are clearly in favor of word matching over statistical learning. Comparing the Apte set to the CONSTRUE set, the relative improvement in WORD is 33% (changing from 21% to 28% in average precision), while the performance change in kNN is -13% (from 92% to 80%). Although we do not know what criteria were used in selecting the test documents, it is clear that using this set for evaluation would lead to inconsistent results, compared to using the other sets. The small size of this test set also makes its results statistically less reliable for evaluation.

3) A harder collection. The categorization task in OHSUMED seems to be more difficult than in Reuters, as evidenced from the significant performance decrease in both WORD and kNN. The category space is two magnitudes larger than Reuters. The number of categories per document is also larger, about 12 to 13 categories on average in OHSUMED while about 1.2 categories in Reuters. This means that the word/category correspondences are more "fuzzy" in OHSUMED. Consequently, the categorization is more difficult to learn. The collections named "HD big" (containing 49 common categories) or "HD small" (containing 28 secondarily common categories) are sub-domains of the heart diseases sub-domain. Since they contains only about 0.2-.3% of the full range of the categories, performance of a classifier on these sets may not be sufficiently representative of its performance over the full domain. This does

[2]OHSUMED is anonymously ftp-able from medir.ohsu.edu in the directory /pub/ohsumed

not invalidate the use of the HD data sets, but it should be taken into consideration in a cross-collection comparison of categorization methods.

4) A problematic collection. The Lewis set of the Reuters corpus seems to be problematic given the large portion of suspiciously unlabelled documents. Note that 58% of the test documents are unlabelled. According to D. Lewis, "it may (or may not) have been a deliberate decision by the indexer" [3]. It is observed by this author that on randomly selected test documents, the categories assigned by kNN appeared to be correct in many cases, but they were counted as failures because these documents were given as unlabelled. This raises a serious question as to whether or not these unlabelled documents should be included in the test set, and treated as negative instances of all categories, as they were handled in the previous experiments[9, 2]. The following analysis addresses this question.

Assume the test set has 58% unlabelled documents, and suppose that all of the unlabelled documents should be assigned categories but are erroneously unlabelled. Let us further assume A to be a perfect classier which assigns a category to a document if and only if they match, and B a trivial classifier which never assigns any category to a document. Now if we use the errorful test set as the gold standard to evaluate the two systems, system A will have an assessed error rate of 58% instead of the true rate of zero percent. System B will have an assessed error rate of 42% instead of the true rate of 100%. Clearly, conclusions based on such a test set can be extremely misleading. In other words, it can make a better method look worse, and a worse method look better. Of course we do not know precisely how many documents in the Lewis set should be labelled with categories, so the argument above is only indicative. Nevertheless, to avoid unnecessary confusion, it would be more sensible to remove the unlabelled documents, or use the Apte set or PARC set instead. This point will be further addressed in Section 5, with a discussion on the problems with the experimental results on the Lewis set.

4 Performance Measures

Classifiers either produce scores, and hence ranked lists of potential category labels, or make binary decisions to assign categories. A classifier that produces a score can be made into a binary classifier by thresholding the score. The inverse process is considerably more difficult. An evaluation method applicable to a scoring classifier may not apply to a binary method. We present evaluations suitable to the two cases and indicate in the following which are used for comparison.

4.1 Evaluation of category ranking

The recall and precision of a category ranking is similar to the corresponding measures used in text retrieval. Given a document as the input to a classifier, and a ranked list of categories as the output, the recall and precision at a particular threshold on this ranked list are defined to be:

$$\text{recall} = \frac{\text{categories found and correct}}{\text{total categories correct}}$$

$$\text{precision} = \frac{\text{categories found and correct}}{\text{total categories found}}$$

where "categories found" means that the categories are above the threshold. For a collection of test documents, the category ranking for each document is evaluated first, then the performance scores are averaged across documents. The conventional 11-point average precision is used to measure the performance of a classifier on a collection of documents[13].

4.2 Evaluation of binary classification

Performance measures in binary classification can be defined using a two-way contingency table (Table 2). The table contains four cells:

- a counts the assigned and correct cases,

[3] Refer to the documentation of the newly refined Reuters-21578 collection.

- b counts the assigned and incorrect cases,
- c counts the not assigned but incorrect cases, and
- d counts the not assigned and correct cases.

Table 2. A contingency table

	YES is correct	No is correct
Assigned YES	a	b
Assigned NO	c	d

The recall (r), precision (p), error (e) and fallout (f) are defined to be:

- $r = a/(a+c)$ if $a+c > 0$, otherwise $r = 1$;
- $p = a/(a+b)$ if $a+b > 0$, otherwise $p = 1$;
- $e = (b+c)/n$ where $n = a+b+c+d > 0$;
- $f = b/(b+d)$ if $b+d > 0$, otherwise $f = 1$.

Given a classifier, the values of r, p, e and f often depend on internal parameter tuning; there is a trade-off between recall and precision in general. A commonly used measure in method comparison [9, 1, 15, 12] is the *break-even point* (BrkEvn) of recall and precision, i.e., when r and p are tuned to be equal. Another common measure[12, 8, 2] is called the F-measure, defined to be:

$$F_\beta(r,p) = \frac{(\beta^2+1)pr}{\beta^2 p + r}$$

where β is the parameter allowing differential weighting of p and r. When the value of β is set to one (denoted as F_1), recall and precision is weighted equally:

$$F_1(r,p) = \frac{2pr}{p+r}$$

When $r = p$, the value of $F_1(r,p)$ is equivalent to the break-even point. Often the break-even point is close to the optimal score of $F_1(r,p)$, but they are not necessarily equivalent. In other words, the optimal score of $F_1(r,p)$ given a system can be higher-valued than the break-even point of this system. Therefore, the break-even point of one system should not be compared directly with the optimal F_1 value of another system.

4.3 Global averaging

There are two ways to measure the average performance of a binary classifier over multiple categories, namely, the *macro-average* and the *the micro-average*. In macro-averaging, one contingency table per category is used, and the local measures are computed first and then averaged over categories. In micro-averaging, the contingency tables of individual categories are merged into a single table where each cell of a, b, c and d is the sum of the corresponding cells in the local tables. The global performance then is computed using the merged table. Macro-averaging gives an equal weight to the performance on every category, regardless how rare or how common a category is. Micro-averaging, on the other hand, gives an equal weight to the performance on every document (category instance), thus favoring the performance on common categories. The micro-average is used in the following evaluation section.

5 Result Analysis

Table 3 summarizes the results of all the categorization methods investigated in this study. The results of kNN, LLSF and WORD are newly obtained. The results of the other methods are either directly from previous publications.

Table 3. Results of different methods in category assignments

	Reuters Apte BrkEvn	Reuters PARC BrkEvn	OHSUMED full range $F(\beta=1)$	OHSUMED HD big $F(\beta=1)$	Reuters Lewis BrkEvn	Reuters CONSTRUE BrkEvn
kNN (N)	.85*	.82*	.51*	.56	.69	-
LLSF (L)	.85*	.81 (−1%)	-	-	-	-
NNets (N)	-	.82*	-	-	-	-
WH (L)	-	-	-	.59* (+5%)	-	-
EG (L)	-	-	-	.54 (−4%)	-	-
RIPPER (N)	.80 (−6%)	-	-	-	.72	-
DTree (N)	[.79]	-	-	-	.67	-
SWAP-1 (N)	.79 (−7%)	-	-	-	-	-
CHARADE (N)	.78 (−8%)	-	-	-	-	-
EXPERTS (N)	.76 (−11%)	-	-	-	.75*	-
Rocchio (L)	.75 (−12%)	-	-	.46 (−18%)	.66	-
NaiveBayes (L)	.71 (−16%)	-	-	-	.65	-
CONSTRUE	-	-	-	-	-	.90*
WORD	.29 (−66%)	.25 (−69%)	.27 (−47%)	.44 (−21%)	.15	-

"L" indicates a linear model, and "N" indicates a non-linear model;
"*" marks the local optimal on a fixed collection;
"(...)" includes the performance improvement relative to kNN;
"[...]" includes a F(1) score; the corresponding break-even point should be the same or slightly lower.

5.1 The new experiments

The KNN, LLSF and WORD experiments used the SMART system for unified preprocessing, including stop word removal, stemming and word weighting. A phrasing option is also available in SMART but not used in these experiments. Several term weighting options (labelled as "ltc", "atc", "lnc" , "bnn" etc. in SMART's notation) were tried, which combine the term frequency (TF) measure and the Inverted Document Frequency (IDF) measure in a variety of ways. The best results (with "ltc" in most cases) are reported in the Table 3.

In kNN and LLSF, aggressive vocabulary reduction based on corpus statistics was also applied as another step of the preprocessing. This is necessary for LLSF which would otherwise be too computationally expensive to apply to large training collections. Computational tractability is not an issue for kNN but vocabulary reduction is still desirable since it improves categorization accuracy. About 1-2% improvements in average precision and break-even point were observed in both kNN and LLSF when an 85% vocabulary reduction was applied. Several word selection criteria were tested, including information gain, mutual information, a χ^2 statistic and document frequency[20]. The best results (using the χ^2 statistic) were included in Table 3. Aggressive vocabulary reduction was not used in WORD because it would reduce the chance of word-based matching between documents and category names.

KNN, LLSF and WORD produces a ranked list of categories first when a test document is given. A threshold on category scores then is applied to obtain binary category assignments to the document. The thresholding on category scores was optimized on training sets (for individual categories) first, and then applied to the test sets.

Other parameters in these systems include:

- k in kNN indicates the number of nearest neighbors used for category prediction, and
- p in LLSF indicates the number of principal components (or singular vectors) used in computing the linear regression.

The performance of kNN is relatively stable for a large range of k, so three values (30, 45 and 65) were tried, and the best results are included in the result table. A satisfactory performance of LLSF depends on whether p is sufficiently large. In the experiments of LLSF on the Reuters sets, the optimal or nearly optimal results were obtained when using about 800 to 1000 singular vectors. A Sun SPARC Ultra-2 Server was used for the experiments. LLSF has not yet applied to the full set of OHSUMED training documents due to computational limitations.

5.2 Cross-experiment comparison

A row-wise comparison in Table 3 allows observation of the performance variance of a method across collections. Unfortunately, most of the rows are sparse except for kNN and WORD. A column-wise comparison allows observation of different methods on a fixed collection. A star marks the best result for each collection.

KNN is chosen to provide the baseline performance on each collection. Several characteristics of this method make it preferable, i.e., efficient to test, easy to scale up, and relatively robust as a learning method. LLSF is equally effective, based on the empirical results obtained so far; however, its training is computationally intensive, and thus has not yet been applied to the full range of the OHSUMED collection. WORD is chosen to provide an secondary reference point in addition to kNN, to enable a quantitative comparison between learning approaches to a simple method that requires no knowledge or training.

The Reuters Apte set has the densest column where the results of eight systems are available. Although the document counts reported by different researchers are somewhat inconsistent[1, 2][4], the differences are relatively small compared to the size of the corpus (i.e., at most 21 miscounted out of over ten thousands training documents, and at most 7 miscounted out of over three thousands of test documents), so the impact of such differences on the evaluation results for this set maybe be considered negligible.

The results on the Lewis set, on the other hand, are more problematic. That is, the inclusion of the 58% "mysteriously" unlabelled documents in the test set makes the results difficult to interpret. For example, most of the methods (kNN, RIPPER, Rocchio and WORD) which were evaluated on both the Apte set and the Lewis set show a significant decrease in their performance scores on the Lewis set, but the scores of EXPERTS are almost insensitive to the inclusion or exclusion of the large amounts of unlabelled documents in the test set. Moreover, EXPERTS has a score near the lower end among all the learning methods evaluated on the Apte set, but the highest score on the Lewis set. Cohen concluded EXPERTS the best performer ever reported on the Lewis set without an explanation on its mysterious insensitivity to the large change in test documents[2]. This is suspicious because the inclusion of a large amounts of incorrectly labelled documents in the test set should decrease the performance of a good classifier, as analyzed in Section 3.

Another example of potential difficulties is the misleading comparison by Apte et al. between SWAP-1 (or rule learning), NaiveBayes and DTree methods (Section 1). They claim an advantage for SWAP-1 based on a score on the Apte set versus scores for the other methods on the Lewis set. To see the perils in such an inference, kNN has a score of 85% on the Apte set, versus the SWAP-1 score of 79% on the same set. On the Lewis set, however, the kNN score is 69%, i.e., 10% lower than Apte SWAP-1 score. Should we then conclude that SWAP-1 is better than kNN, or the opposite? More interestingly, a recent result using a DTree algorithm (via C4.5) due to Moulinier scores 79% on the Apte set[11], which is exactly the same as the SWAP-1 result. How should this be interpreted? To make the point clear, the Lewis set should not be used for text categorization evaluation unless the status of the unlabelled documents is resolved. Results obtained on this set can be seriously misleading, and therefore should not be used for a comparison or to draw any conclusions. Inferences based on the CONSTRUE set should also be questioned because the test set is much smaller than the other sets, contains 20% mysteriously unlabelled documents, and may possibly be a biased selection (Section 3).

Finally, it may worth mentioning that the cross-method comparisons here are not necessarily precise, because some experimental parameters might contribute to a difference in the results but are not available. For instance, different choices could be made in stemming, term selection, term weighting, sampling strategies for training data, thresholding for binary decisions, and so on. Without detailed information, we cannot be sure that a one or two percent difference in break-even point or F-measure is an indication of the theoretical strength or weakness of a learning method. It is also unclear how a significance test should be designed, given that the performance of a method is compressed into a single number, e.g., to the break-even point of averaged recall and precision. A variance analysis would be difficult given that the necessary input data is not generally published. Further research is needed on this issue. Nonetheless, missing detailed information should not prohibit the good use of available information. As long as the related issues are carefully addressed, as shown above, an integrated view across methods and experiments is possible, especially for significant variations in results on a fully-labelled common test set.

[4] Inconsistent numbers about the documents in the Apte set were found in previous papers and the corpus documentation, presumably due to counting errors or processing errors by the individuals. The numbers included in Table 1 are those agreed by at least two research sites. Details are available through yiming@cs.cmu.edu.

6 Discussions

Despite the imperfectness of the comparison across collections and experiments, the integrated results are clearly informative, enabling a global observation which is not possible otherwise. Several points in the results appear to be interesting regarding the analysis of classification models.

The impressive performance of kNN is rather surprising given that the method is quite simple and computationally efficient. It has the best performance, together with LLSF, on the Apte set, and is equally effective as NNets on the PARC set. On the OHSUMED set, it is the only learning method evaluated on the full domain, i.e., a category space which is more than one hundred times larger than those used in the evaluations of most learning algorithms. When extending the target space from the sub-domain of 49 "HD big" categories to the full domain of 14,321 categories, the performance decline of kNN is only 5% in absolute value, or a 9% decrease relative. In contrast, the performance of WORD declined from 44% to 27%, or a 39% decrease relatively. This suggests that kNN is more powerful than WORD in making fine distinctions between categories. Or, it "failed" more gracefully when the category space grows by several orders of magnitude.

The good performance of WH on "HD big" calls for deeper analysis. WH is an incremental learning algorithm trained based on an least squares fit criterion. Its optimal performance therefore should be bounded by or close to a least squares fit solution obtained in a batch-mode training, such as LLSF. It would be interesting in future research to compare the empirical results of LLSF with WH. It is also worth asking whether there is something else, beyond the core theory, which contributed to the good performance. In the WH experiment on "HD big", Lewis used a "pocketing" strategy to select a subset of training instances from a large pool[8]. This is similar or equivalent to a sampling strategy which divides available training instances into small chunks, examines one chunk at a time using a validation set, and adds a new chunk to the selected ones only if it improves the performance on the validation set. This strategy would be particularly effective when the training data are highly noisy, such as OHSUMED documents. Nevertheless, the sampling strategy is not a part of the WH algorithm, and can be used in any other classifiers. It would be interesting to examine the effect of the pocketing strategy in kNN on OHSUMED in feature research, for example.

Rocchio has a relatively poor performance compared to the other learning methods, and is almost as poor as WORD on the "HD big" subset, surprisingly. This suggests that Rocchio may not be a good choice (although commonly used) for the baseline in evaluating learning methods, because it is inferior to most methods and thus would be not very informative especially when the comparison includes only one or two other learning methods. In other words, Rocchio is a straw man rather than a challenging standard. KNN would be a better alternative, for instance.

The mixture of the linear (L) and non-linear (N) classifiers among the top-ranking performers (WH, NNets, kNN and LLSF) suggests that no general conclusion can be fetched regarding reliable improvement of non-linear approaches over linear approaches, or vice versa. It is also hard to draw a conclusion about the advantage of a multiple-category classification model (kNN or LLSF) over unary classification models (WH, NNets, EG, RIPPER etc.) Either the category independence assumption in the latter type of methods is reasonable, or an improvement in kNN and LLSF is needed in the handling of the dependence or mutual exclusiveness among categories. Resolving this issue requires future research.

The rule induction algorithms (SWAP-1, RIPPER and CHARADE) have a similar performance, but below the local optimum of kNN on the Apte set, and also below some other classifiers (WH, NNets) based on an indirect comparison across collections via kNN as the baseline. This observation raises a question with respect to a claim about the particular advantage of rule learning in text categorization. The claim was based on context-sensitivity, i.e., the power in capturing term combinations[1, 2]. It seems that the methods which do not explicitly identify term combinations but use the context implicitly (such as in WH, NNets, kNN and LLSF) performed at least as well.

It may be worth mentioning that a classifier can have a degree of context-sensitivity without explicitly identifying term combinations or phrases. The classification function in LLSF, for instance, is sensitive to weighted linear combinations of words that co-occur in training documents. This does not makes it equivalent to a non-linear model, but makes a fundamental distinction from the methods based on a term independence assumption, such as naive Bayes models. This may be a reason for the impressive performance of kNN and LLSF. It would be interesting to compare them with NaiveBayes if the latter were tested on the Apte set, for example.

7 Conclusions

The following conclusions are reached from this study:

1. The performance of a classifier depends strongly on the choice of data used for evaluation. Using a seriously problematic collection[8], comparing categorization methods without analyzing collection differences[1], and drawing conclusion based on the results of flawed experiments[2] raise questions about the validity of some published evaluations. These problems need to be addressed to clarify of the confusions among researchers, and to prevent the repetition of similar mistakes. Providing information and analysis on these problems is a major effort in this study.

2. Integrating results from different evaluations into a global comparison across methods is possible, as shown in this paper, by evaluating one or more baseline classifiers on multiple collections, by normalizing the performance of other classifiers using a common baseline classifier, and by analyzing collection biases based on performance variations of several baseline classifiers. Such an integration allows insights on methods and collections which are rarely apparent in comparisons involving two or three classifiers. It also shows an evaluation methodology which is complementary to the effort to standardize collections and unify evaluations.

3. WH, kNN, NNets and LLSF are the top performers among the learning methods whose results were empirically validated in this study. Rocchio had a relatively poor performance, on the other hand. All the learning methods outperformed WORD, the non-learning method. However, the differences between some learning methods are not as large as previously claimed[1, 2]. It is not evident in the collected results that non-linear models are better than linear models, or that more sophisticated methods outperform simpler ones. Conclusive statements on the strengths and weaknesses of different models requires further research.

4. Scalability of a classifier when the problem size grows by several magnitudes, or when the category space becomes a hundred times denser, has been rarely examined in text categorization evaluations. KNN is the only learning method evaluated on the full set of the OHSUMED categories. Its robustness in scaling up and dealing with harder problems, and its computational efficiency make it the method of choice for approaching very large and noisy categorization problems.

8 Acknowledgement

I would like to thank Jan Pedersen at Verity, David Lewis and William Cohen at AT&T, and Isabelle Moulinier at University of Paris VI for providing the information of their experiments. I would also like to thank Jaime Carbonell at Carnegie Mellon University for suggesting an improvement in binary decision making, Yibing Geng and Danny Lee for the programming support, and Chris Buckley at Cornell for making the SMART system available.

References

[1] C. Apte, F. Damerau, and S. Weiss. Towards language independent automated learning of text categorization models. In *Proceedings of the 17th Annual ACM/SIGIR conference*, 1994.

[2] William W. Cohen and Yoram Singer. Context-sensitive learning metods for text categorization. In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996. 307-315.

[3] R.H. Creecy, B.M. Masand, S.J. Smith, and D.L. Waltz. Trading mips and memory for knowledge engineering: classifying census returns on the connection machine. *Comm. ACM*, 35:48–63, 1992.

[4] Ed. DK Harman. *Overview of the Third Text REtrieval Conference (TREC-3).* US Government Printing Office, Washington, DC, 1995.

[5] N. Fuhr, S. Hartmanna, G. Lustig, M. Schwantner, and K. Tzeras. Air/x - a rule-based multistage indexing systems for large subject fields. In 606-623, editor, *Proceedings of RIAO'91*, 1991.

[6] P.J. Hayes and S. P. Weinstein. Construe/tis: a system for content-based indexing of a database of new stories. In *Second Annual Conference on Innovative Applications of Artificial Intelligence*, 1990.

[7] W. Hersh, C. Buckley, T.J. Leone, and D. Hickman. Ohsumed: an interactive retrieval evaluation and new large text collection for research. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 192–201, 1994.

[8] David D. Lewis, Robert E. Schapire, James P. Callan, and Ron Papka. Training algorithms for linear text classifiers. In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1996. 298-306.

[9] D.D. Lewis and M. Ringuette. Comparison of two learning algorithms for text categorization. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

[10] I. Moulinier. Une approche de la catégorisation de textes par lápprentissage symbolique. In *PhD thesis, Université Pierre et Marie Curie (Paris 6)*, 1996.

[11] I. Moulinier. Is learning bias an issue on the text categorization problem? In *Technical report, LAFORIA-LIP6, Universite Paris VI*, page (to appear), 1997.

[12] I. Moulinier, G. Raskinis, and J. Ganascia. Text categorization: a symbolic approach. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval*, 1996.

[13] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Pennsylvania, 1989.

[14] K. Tzeras and S. Hartman. Automatic indexing based on bayesian inference networks. In *Proc 16th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 22–34, 1993.

[15] E. Wiener, J.O. Pedersen, and A.S. Weigend. A neural network approach to topic spotting. In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

[16] Y. Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *17th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, pages 13–22, 1994.

[17] Y. Yang. An evaluation of a statistical approaches to medline indexing. In *Proceedings of the 1996 Annual Full Symposium of the American Medical Informatics Association (1996 AMIA)*, pages 358–362, 1996.

[18] Y. Yang and C.G. Chute. A linear least squares fit mapping method for information retrieval from natural language texts. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 92)*, pages 447–453, 1992.

[19] Y. Yang and C.G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transaction on Information Systems (TOIS)*, pages 253–277, 1994.

[20] Y. Yang and J.P. Pedersen. Feature selection in statistical learning of text categorization. In *The Fourteenth International Conference on Machine Learning*, page (to apprear), 1997.