Techniques for Improving the Performance of Naive Bayes for Text Classification

Karl-Michael Schneider

University of Passau, Department of General Linguistics
Innstr. 40, 94032 Passau, Germany
schneide@phil.uni-passau.de
WWW home page: http://www.phil.uni-passau.de/linguistik/schneider/

Abstract. Naive Bayes is often used in text classification applications and experiments because of its simplicity and effectiveness. However, its performance is often degraded because it does not model text well, and by inappropriate feature selection and the lack of reliable confidence scores. We address these problems and show that they can be solved by some simple corrections. We demonstrate that our simple modifications are able to improve the performance of Naive Bayes for text classification significantly.

1 Introduction

Text classification is the assignment of predefined categories to text documents. Text classification has many applications in natural language processing tasks such as E-mail filtering [1, 2], news filtering [3], prediction of user preferences [4] and organization of documents [5]. Because of the variety of languages, applications and domains, machine learning techniques are commonly applied to infer a classification model from example documents with known class labels. The inferred model can then be used to classify new documents. A variety of machine learning paradigms have been applied to text classification, including rule induction [6], Naive Bayes [7], memory based learning [8], decision tree induction [9] and support vector machines [10].

This paper is concerned with the Naive Bayes classifier. Naive Bayes uses a simple probabilistic model that allows to infer the most likely class of an unknown document using Bayes' rule. Because of its simplicity, Naive Bayes is widely used for text classification [4, 5, 1, 2, 11].

The Naive Bayes model makes strong assumptions about the data: it assumes that words in a document are independent. This assumption is clearly violated in natural language text: there are various types of dependences between words induced by the syntactic, semantic, pragmatic and conversational structure of a text. Also, the particular form of the probabilistic model makes assumptions about the distribution of words in documents that are violated in practice [12]. Nonetheless, Naive Bayes performs quite well in practice, often comparable to more sophisticated learning methods [13, 14].

One could suspect that the performance of Naive Bayes can be further improved if the data and the classifier better fit together. There are two possible approaches: (i) modify the data, (ii) modify the classifier (or the probabilistic model).

Many researchers have proposed modifications to the way documents are represented, to better fit the assumptions made by Naive Bayes. This includes extracting more complex features, such as syntactic or statistical phrases [15], and exploiting semantic relations using lexical resources [16]. These attempts have been largely unsuccessful. Another way to improve the document representation is to extract features by word clustering [17] or by transforming the feature space [18]. These methods did show some improvement of classification accuracy.

Instead of changing the document representation by using other features than words, it is also possible to manipulate the text directly, e.g. by altering the occurrence frequencies of words in documents [19]. This can help the data to better fit the distribution assumed by the model.

The most important way to better fit the classifier to the data is to choose an appropriate probabilistic model (see Sect. 2). Some researchers have also tried to improve performance by altering the way the model parameters are estimated from training data [20].

In this paper we review and explain a number of very simple techniques that can help to improve the accuracy of a Naive Bayesian text classifier dramatically. Some of them have been proposed before or are simplifications of existing methods. Many of these techniques appear to be counterintuitive but can be explained by the particular (statistical) properties of natural language text documents.

2 Naive Bayes

Bayesian text classification uses a parametric mixture model to model the generation of documents [7]. The model has the following form:

$$p(d) = \sum_{j=1}^{|C|} p(c_j) p(d|c_j)$$

where c_j are the mixture components (that correspond to the possible classes) and $p(c_j)$ are prior probabilities. Using Bayes' rule, the model can be inverted to get the posterior probability that d was generated by the mixture component c_j:

$$p(c_j|d) = \frac{p(c_j)p(d|c_j)}{p(d)}$$

To classify a document, the classifier selects the class with maximum posterior probability, given the document, where $p(d)$ is constant and can be ignored:

$$c^*(d) = \operatorname*{argmax}_j p(c_j)p(d|c_j) \qquad (1)$$

The prior probabilities $p(c_j)$ are estimated from a training corpus by counting the number of training documents in each class c_j.

The distribution of documents in each class, $p(d|c_j)$, cannot be estimated directly. Rather, it is assumed that documents are composed from smaller units, usually words or word stems. To make the estimation of parameters tractable, we make the Naive Bayes assumption: that the basic units are distributed independently.

There are several Naive Bayes models that make different assumptions about how documents are composed from the basic units. The most common models are: the binary independence model (a.k.a. multi-variate Bernoulli model), the Poisson Naive Bayes model, and the multinomial model (the latter is equivalent to the Poisson model under the assumption that the class of a document is marginally independent of its length) [7, 21]. The most apparent difference between these models is that the Poisson model and the multinomial model use word occurrence frequencies, while the binary independence model uses binary word occurrences. In this paper we consider the multinomial Naive Bayes model because it is generally superior to the binary independence model for text classification [7, 21].

In the multinomial model, a document d is modeled as the outcome of $|d|$ independent trials on a single random variable W that takes on values $w_t \in V$ with probabilities $p(w_t|c_j)$ and $\sum_{t=1}^{|V|} p(w_t|c_j) = 1$. Each trial with outcome w_t yields an independent occurrence of w_t in d. Thus a document is represented as a vector of word counts $d = \langle x_t \rangle_{t=1\ldots|V|}$ where each x_t is the number of trials with outcome w_t, i.e. the number of times w_t occurs in d. The probability of d is given by the multinomial distribution:

$$p(d|c_j) = p(|d|)|d|! \prod_{t=1}^{|V|} \frac{p(w_t|c_j)^{x_t}}{x_t!}$$

Here we assume that the length of a document is chosen according to some length distribution, independently of the class. Plugging this into (1) we get the following form (omitting parts that do not depend on the class):

$$c^*(d) = \operatorname*{argmax}_{c_j} p(c_j) \prod_{t=1}^{|V|} p(w_t|c_j)^{x_t} \tag{2}$$

The parameters $p(w_t|c_j)$ are estimated by counting the occurrences of w_t in all training documents in c_j, using a Laplacean prior:

$$p(w_t|c_j) = \frac{1 + n(c_j, w_t)}{|V| + n(c_j)}$$

where $n(c_j, w_t)$ is the number of occurrences of w_t in the training documents in c_j and $n(c_j)$ is the total number of word occurrences in c_j.

3 Word Frequency Information

It is usually claimed that the multinomial model gives higher classification accuracy than the binary independence model on text documents because it models word occurrence frequencies [7, 21]. Contrary to this belief, we show that word frequency hurts more than it helps, and that ignoring word frequency information can improve performance dramatically.

The multinomial Naive Bayes model treats each occurrence of a word in a document independently of any other occurrence of the same word. In reality, however, multiple occurrences of the same word in a document are not independent. When a word occurs once, it is likely to occur again, i.e. the probability of the second occurrence is much higher than that of the first occurrence. This is called *burstiness* [12]. The multinomial model does not account for this phenomenon. This results in a large underestimation of the probability of documents with multiple occurrences of the same word.

In [19] a transformation of the form $x'_t = \log(1+x_t)$ was applied to the word frequency counts in a document in order to better fit the data to the probabilistic model. This does not eliminate word frequencies but has the effect of pushing down larger counts. An even simpler, yet less accurate method is to remove word frequency information altogether using the transform $x'_t = \min\{x_t, 1\}$. This can be thought of as discarding all additional occurrences of words in a document. Instead of transforming the word counts, we can change the classification rule as in (3):

$$c^*(d) = \operatorname*{argmax}_{c_j} p(c_j) \prod_{t=1}^{|V|} p(w_t|c_j)^{\min\{x_t,1\}} \tag{3}$$

and the parameter estimation as in (4), where $d(c_j, w_t)$ is the number of documents containing w_t in c_j:

$$p(w_t|c_j) = \frac{1 + d(c_j, w_t)}{|V| + \sum_{s=1}^{|V|} d(c_j, w_s)} \tag{4}$$

We compare classification accuracy with and without word frequency information on two datasets: 20 Newsgroups[1] [3] and WebKB[2] [11]. We remove all headers from the newsgroup articles. We use only the four most populous classes *course*, *faculty*, *project* and *student* of the WebKB corpus and remove all HTML markup. All non-alphanumeric characters are removed, and letters are converted to lower case. In addition, a 100 word stoplist is applied to the newsgroups articles and all numbers are mapped to a special token. Following [7] we perform feature selection using Mutual Information and vary the number of selected features between 20 and 20,000.

Figure 1 shows the results. On 20 Newsgroups, removing word frequency information improves classification accuracy on average by 7.5 percentage points (23% relative improvement). On WebKB, accuracy is improved on average by 3.8 percentage points (an average error reduction of 21%).

4 Class Prior Probabilities

In (2) one can see that for longer documents the classification scores are dominated by the word probabilities, and the prior probabilities hardly affect the classification. However, in situations where documents are usually very short *and* the class distribution is skewed, the prior probabilities may affect the classification negatively. In such cases,

[1] http://people.csail.mit.edu/people/jrennie/20Newsgroups/

[2] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/

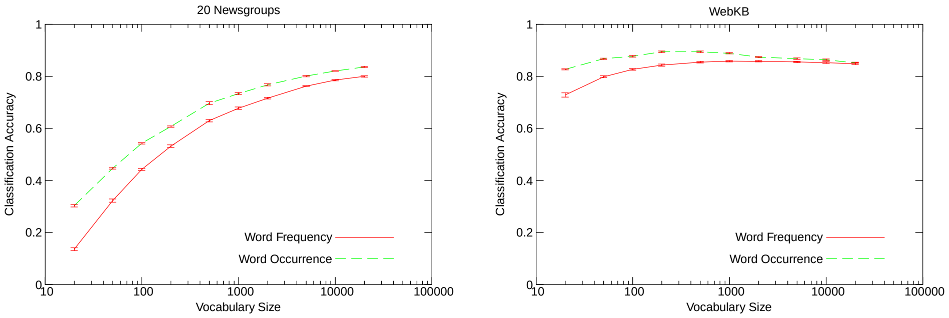

Fig. 1. Classification accuracy of multinomial Naive Bayes with and without word frequency information, on 20 Newsgroups (left) and WebKB (right). Results are averaged over five, respectively ten, cross-validation trials, with small error bars shown. The confidence level using a two-tailed paired t-test is above 99.99% for 20 Newsgroups and above 99% for up to 5,000 words on WebKB.

ignoring prior probabilities altogether (or equivalently, assuming uniform priors) can improve classification accuracy.

To examine the impact of prior probabilities on the classification of short documents, we use a corpus of 4,353 hypertext links from the Linguistics Links Database.[3] There are 16 categories with 60 documents in the smallest category and 1,583 documents in the largest category. We use only the link texts. We convert all letters to lower case, remove tokens on a stop list and replace location names, dates and numerical expressions by special tokens. The average document length after preprocessing is 9 tokens. 505 documents have less than 4 tokens, and 47 documents have only one token.

We produce 20 train/test splits using stratified random sampling with 70% of the data (3,048 documents) for training and 30% (1,305 documents) for testing. Experiments are done using 20 trials over these splits and averaging the results. Table 1 compares classification accuracy with prior probabilities estimated from the training corpus (as described in Sect. 2) and uniform priors, with various vocabulary sizes. Using uniform priors improves performance by more than 5 percentage points when all or almost all words are used. Also, with uniform priors classification accuracy depends less on the number of selected words, whereas with estimated prior probabilities there is a drop in performance when more words are used.

5 Feature Selection

Feature selection is commonly regarded as a necessary step in text classification, due to the high dimensionality of text (i.e. the large vocabulary size). Feature selection increases the efficiency of the classifier because less parameters need to be estimated and less probabilities need to be multiplied. Often, it also improves the accuracy of the classifier provided that the size of the feature subset is correctly chosen.

[3] http://www.phil.uni-passau.de/linguistik/linguistik_urls/

Table 1. Comparison of classification accuracy using prior probabilities estimated from training data and uniform prior probabilities. Words are selected according to their mutual information with the class variable. Without feature selection, the average vocabulary size is 2,742 words. The confidence level using a two-tailed paired t-test is above 99.98%.

Prior Probabilities	500 Words	1,000 Words	2,000 Words	Full Vocabulary
Estimated	0.7120	0.6897	0.6635	0.6544
Uniform	0.7243	0.7278	0.7146	0.7095

Feature selection for text classification uses a greedy filtering approach: A scoring function is used to assign a score to each feature independently, and the highest scored features are selected. The feature set size can be specified directly or by applying a threshold to the feature scores. The main question is how to determine the optimal number of selected features (or the optimal threshold). A common strategy is by testing the classifier with different values on a validation set.

Most feature scoring functions compute some statistics of the training data or some information theoretic measure. The following functions are very common in text categorization: Chi-square [22] uses the χ^2 statistic to estimate the dependence between a feature and the class of a document. Mutual Information (MI) [7] is an information theoretic measure that measures the amount of information a feature gives about the class. Bi-Normal Separation (BNS) [23] assumes that the occurrence of a word is modeled by the event of a random normal variable exceeding a certain threshold, and measures the separation between the thresholds for a class and its complement (i.e. the union of the other classes).

The above scoring functions are not directly related to the probabilistic framework of Naive Bayes. In the following we present a novel feature scoring function called CRQ (Cluster Representation Quality) that is derived directly from the probabilistic Naive Bayes model, rather than from some independent statistics. CRQ is based on ideas from distributional clustering [17]. Distributional clustering aims at finding a clustering of a set of elements that minimizes the distance (in some information theoretic sense) between the elements in the same cluster (tight clusters) and simultaneously maximizes the distance between the elements in different clusters (distant clusters).

In our approach to feature selection, the elements are the training documents and the clusters are the classes in the training set. The important difference is that we do not change the clustering of the training documents (i.e. their classification) but rather seek to improve the clustering quality by removing certain words from the vocabulary.

5.1 Cluster Representation Quality

First, note that a document d can be regarded as a probability distribution over words with $p(w_t|d) = n(w_t,d)/|d|$, where $n(w_t,d)$ is the number of times w_t occurs in d and $|d|$ denotes the length of d. To measure the distance of one probability distribution, p_1, from another distribution, p_2, we use Kullback-Leibler (KL) divergence [24], defined by $D(p_1\|p_2) = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)}$. In [17] the distance between a finite set of probability distributions is measured using generalized Jensen-Shannon (JS) divergence. It can be shown that JS divergence is the weighted sum of the KL divergences of the distributions from the (weighted) mean distribution.

For any document d_i, let $p_i = p(W|d_i)$ denote the probability distribution induced by d_i, as above. The class-conditional distributions $p_j(W) = p(W|c_j)$ can be expressed as the weighted sum of the p_i (for $d_i \in c_j$): $p_j = \sum_{d_i \in c_j} \pi_i p_i$, where the weights $\pi_i = |d_i| / \sum_{d_i \in c_j} |d_i|$ are the normalized document lengths. Instead of the weighted sum of KL divergences, we measure the *within-cluster divergence* using the unweighted mean of the distributions p_i (for $d_i \in c_j$) from p_j. We do this because the standard definition of classification accuracy (which we would like to optimize) gives each document the same weight, regardless of its length. In contrast, the weighted sum of the KL divergences would give longer documents a higher weight, which corresponds to a misclassification cost that is proportional to the document length.

Similarly, the *total divergence* is defined as the unweighted mean of the distributions p_i (for all d_i in the training corpus) from the mean distribution $p = \sum_j \sum_{d_i \in c_j} \pi'_i p_i$, where $\pi'_i = |d_i| / \sum_j \sum_{d_i \in c_j} |d_i|$. Then the cluster representation quality of the training corpus is defined as the difference between total divergence and within-cluster divergence (i.e. the reduction in divergence from the mean due to clustering the training documents into their classes):

$$\begin{aligned} & \frac{1}{N} \sum_{j=1}^{|C|} \sum_{d_i \in c_j} \Big[D(p_i \| p) - D(p_i \| p_j) \Big] \\ = \quad & \frac{1}{N} \sum_{j=1}^{|C|} \sum_{d_i \in c_j} \sum_{t=1}^{|V|} p_i(w_t) \Big[\log p_j(w_t) - \log p(w_t) \Big] \end{aligned} \tag{5}$$

Note that when the total divergence is large, the documents are scattered over a large space in the document space. If, on the other hand, within-cluster divergence is small (tight clusters), the clusters are (on average) far apart.

5.2 An Information Theoretic Analysis of Naive Bayes

The connection to Naive Bayes is established via an information theoretic interpretation of the Naive Bayes classifier. (2) can be written in the following form by taking logarithms, dividing by the length of d and adding the entropy of $p(W|d)$, $H(p(W|d)) = -\sum_t p(w_t|d) \log p(w_t|d)$:

$$\begin{aligned} c^*(d) &= \operatorname*{argmax}_{c_j} \frac{1}{|d|} \log p(c_j) - \sum_{t=1}^{|V|} p(w_t|d) \log \frac{p(w_t|d)}{p(w_t|c_j)} \\ &= \operatorname*{argmin}_{c_j} D(p(W|d) \| p(W|c_j)) - \frac{1}{|d|} \log p(c_j) \end{aligned} \tag{6}$$

Note that the modifications in (6) do not change the classification of documents. (6) shows that, ignoring prior probabilities, Naive Bayes selects the class with minimum KL-divergence from the document. Therefore, classification accuracy is improved if each document is more similar to its true class than to other classes. This is achieved by maximizing the distance between the classes, i.e. (5).

5.3 Cluster Quality Based Feature Scores

Note that (5) can be written as a sum over words. Our new feature scoring function, called CRQ, is derived from (5) by using the value of the term in the sum that corresponds to the word:

$$CRQ(w_t) = \frac{1}{N} \sum_{j=1}^{|C|} \sum_{d_i \in c_j} p_i(w_t) \Big[\log p_j(w_t) - \log p(w_t) \Big] \tag{7}$$

Note that $CRQ(w_t)$ can be negative. (5) is maximized when all words w_t for which $CRQ(w_t)$ is negative are discarded.[4] Therefore, a natural threshold for CRQ is 0. In contrast, Chi-square, MI and BNS do not have natural thresholds. To find a good threshold one could guess some reasonable values and then use a validation set. However, this would have to be done for every dataset since the best threshold may depend on the data, whereas the (theoretically) optimal threshold 0 for CRQ is data independent.

We compare three different strategies for determining the number of selected words, using the four feature scoring functions:

- use a fixed threshold that depends on the scoring function but not on the dataset. To find reasonable values, we experimented with some thresholds. In addition, for CRQ we used the theoretically optimal threshold 0.
- specify the number of features directly, for each scoring function and each dataset. We tried some reasonable values and used the best one.
- use the full vocabulary (i.e. no feature selection).

In addition to 20 Newsgroups and WebKB, we use two other datasets: Ling-Spam[5] [2] and Reuters-21578.[6] Ling-Spam consists of 83.4% legitimate E-mails and 16.6% spam E-mails. For the Reuters-21578 setup, see Sect. 6.

Table 2 shows the results. We use the full vocabulary and the best number of features as a baseline against which we compare the performance of the thresholding strategy.

Chi-square with threshold 0.1 outperforms the baseline on three out of four datasets, while CRQ with threshold 0 performs better than the baseline on half of the datasets. However, the baseline for CRQ is generally higher than for Chi-square, and CRQ generally gives better performance than Chi-square. Moreover, on Reuters and Ling-Spam, CRQ with threshold 0 selects considerably less words than Chi-square with threshold 0.1. In general, CRQ is more sensitive to the complexity of the datasets (one notable exception is WebKB).

Note that all strategies except using the full vocabulary and CRQ with threshold 0 require experimentation to find good values. Often it is difficult to guess reasonable values. For example, the performance on 20 Newsgroups is best with a very large vocabulary, much larger than what one would guess. Therefore, if feature selection is required, CRQ is a priori a better scoring method because of its theoretically optimal

[4] However, removing words from the vocabulary changes the distribution of the remaining words.

[5] http://www.iit.demokritos.gr/skel/i-config/downloads/

[6] http://www.daviddlewis.com/resources/testcollections/reuters21578/

Table 2. Classifier performance and number of selected words for different feature selection strategies: thresholding on the feature score (with the same threshold for all datasets), individual selection of the vocabulary size for each dataset (best), no feature selection (full). Results printed in bold outperform both the full vocabulary and the best vocabulary size.

	20 Newsgroups		WebKB		Ling-Spam		Reuters-21578	
	Words	Acc	Words	Acc	Words	F(spam)	Words	Recall
Chi^2=1	22,809	0.8012	9,671	0.8539	13,504	0.9474	12,839	0.8149
Chi^2=0.1	76,797	**0.8070**	32,712	0.8479	44,498	**0.9503**	18,861	**0.8172**
Chi^2 best	20,000	0.7999	1,000	0.8589	5,000	0.9456	20,000	0.8172
MI=10^{-6}	25,926	0.8033	11,904	0.8523	6,845	0.9445	6,802	0.8150
MI=10^{-7}	88,932	**0.8078**	32,776	0.8479	17,149	0.9455	18,014	0.8172
MI best	20,000	0.7998	1,000	0.8582	5,000	0.9434	1,000	0.8210
BNS=0.1	19,684	0.7994	20,877	0.8517	24,964	0.9516	17,449	0.8176
BNS=0.05	49,274	**0.8076**	32,550	0.8478	45,372	0.9503	20,086	0.8176
BNS best	20,000	0.8001	5,000	0.8660	700	0.9753	2,000	0.8323
CRQ=10^{-6}	38,521	**0.8231**	14,715	0.8549	7,759	0.9478	3,755	0.8294
CRQ=10^{-7}	71,493	**0.8249**	29,794	0.8504	18,300	**0.9507**	6,750	0.8252
CRQ=0	94,616	**0.8213**	32,090	0.8500	23,089	**0.9507**	7,617	0.8247
CRQ best	20,000	0.8164	1,000	0.8708	4,000	0.9429	5,000	0.8301
Full	100,874	0.8063	32,873	0.8480	56,247	0.9503	22,430	0.8161

threshold. Without experimenting to find a good vocabulary size, CRQ provides a simple way to determine the number of features by simply setting the threshold to 0. With this strategy, in most cases the classifier performance is comparable to or better than any of the other methods, and in all cases is higher than using the full vocabulary.

6 Confidence Scores

Sometimes it is desirable to have the classifier produce classification scores that reflect the confidence of the classifier that a document belongs to a class. For example, in binary classification problems where one class (the target class) contains examples that are relevant to some query, a document could be assigned to the target class only if its confidence score exceeds some threshold. In multi-label classification tasks (where each document can belong to zero, one or more classes), a document can be assigned to all classes for which the confidence is above the threshold. Such confidence scores must be independent of document length and complexity.

The posterior probabilities $p(c_j|d)$ computed by Naive Bayes are inappropriate as confidence scores because they are usually completely wrong and tend to go to zero or one exponentially with document length [25]. This is a consequence of the Naive Bayes independence assumption and the fact that the words in a document are not really independent. Note that the classification decision of Naive Bayes is not affected as long as the ranking of the classes is not changed (in fact it has been argued that the large bias can reduce classification error [14]).

We follow the approach in [11] to get better confidence scores for Naive Bayes. First, we replace the posterior scores with the KL-divergence scores in (6):

$$score(c_j|d) = \frac{1}{|d|} \log p(c_j) - \sum_{t=1}^{|V|} p(w_t|d) \log \frac{p(w_t|d)}{p(w_t|c_j)}$$

This has two effects. By taking logarithms and dividing by the length of a document, instead of multiplying conditional probabilities (as in Eq. 2) we calculate their geometric mean and thus account for the impact of wrong independence assumptions under varying document lengths. Furthermore, by adding the entropy of (the probability distribution induced by) the document, we account for varying document complexities.

Finally, to make the scores comparable across different documents, we normalize the scores such that they form a probability distribution over classes (i.e. the scores for all classes sum to one):

$$conf(c_j|d) = \frac{score(c_j|d)}{\sum_{k=1}^{|C|} score(c_k|d)}$$

We compare the posterior scores and the confidence scores on the Reuters-21578 dataset, using the ModApte split with only the 10 largest topics [26]. We remove all non-alphabetic characters and convert all letters to lower case. In addition, we map all numbers to a special token. For each topic, we build a binary classifier using all documents in that topic as relevant examples and all other documents as non-relevant examples. The threshold is set for each classifier individually such that recall equals precision (precision/recall break-even point).

Figure 2 shows the results. Microaveraged recall is on average 2 percentage points higher using confidence scores, but macroaveraged recall is improved on average by 4.2 percentage points. This indicates that the confidence scores improve performance especially on the smaller topics (e.g. with 5,000 features the relative improvement is up to 28% on individual topics).

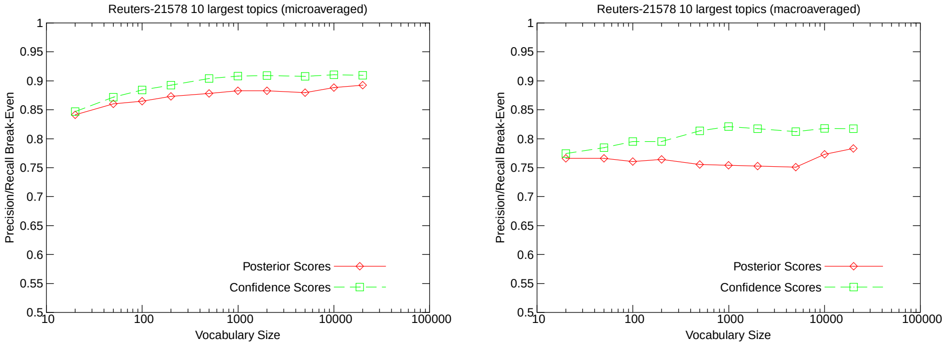

Fig. 2. Comparison of posterior scores and smoothed confidence scores on the Reuters-21578 dataset, using microaveraged (left) and macroaveraged (right) precision/recall break-even.

7 Conclusions

This paper has described some simple modifications of the Naive Bayes text classifier that address problems resulting from wrong independence assumptions. Some of the modifications have been proposed before, some are simplifications of existing methods, and some are new. In particular, we have used a simple transformation that effectively removes duplicate words in a document to account for burstiness phenomena in text; we have proposed to use uniform priors to avoid problems with skewed class distributions when the documents are very short; we have used a different but equivalent form of the Naive Bayes classification rule in an information theoretic framework to obtain more reliable confidence scores; and by viewing a training corpus as a clustering of the training documents and feature selection as a way to improve that clustering, we have obtained a new feature scoring function for Naive Bayes.

The main contribution of this paper is our novel feature scoring function, which is able to distinguish features that improve the clustering of the training documents (and thus are useful for classification) from features that degrade the clustering quality (and thus should be removed). The threshold that separates these two sets of features is data-independent. This is a big advantage over other feature scoring functions because (in theory) optimizing the threshold on a validation set becomes unnecessary. We have, however, noted that removing features may change the scores of the remaining features. Future work will have to examine the implications of that.

The modifications of Naive Bayes proposed in this paper have been applied in isolation. Future work will also consider combinations of them.

References

1. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: Learning for Text Categorization: Papers from the AAAI Workshop, Madison Wisconsin, AAAI Press (1998) 55–62 Technical Report WS-98-05.
2. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C.D., Stamatopoulos, P.: Learning to filter spam e-mail: A comparison of a Naive Bayesian and a memory-based approach. In Zaragoza, H., Gallinari, P., Rajman, M., eds.: Proc. Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), Lyon, France (2000) 1–13
3. Lang, K.: NewsWeeder: Learning to filter netnews. In: Proc. 12th International Conference on Machine Learning (ICML-95), Morgan Kaufmann (1995) 331–339
4. Pazzani, M., Billsus, D.: Learning and revising user profiles: The identification of interesting web sites. Machine Learning **27** (1997) 313–331
5. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. In: Proc. 14th International Conference on Machine Learning (ICML-97). (1997) 170–178
6. Cohen, W.W., Singer, Y.: Context-sensitive learning methods for text categorization. ACM Transactions on Information Systems **17** (1999) 141–173
7. McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Learning for Text Categorization: Papers from the AAAI Workshop, AAAI Press (1998) 41–48 Technical Report WS-98-05.

8. Yang, Y., Liu, X.: A re-examination of text categorization methods. In: Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99). (1999) 42–49
9. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
10. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proc. 10th European Conference on Machine Learning (ECML98). Volume 1398 of Lecture Notes in Computer Science., Heidelberg, Springer (1998) 137–142
11. Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., Slattery, S.: Learning to construct knowledge bases from the World Wide Web. Artificial Intelligence **118** (2000) 69–113
12. Katz, S.M.: Distribution of content words and phrases in text and language modelling. Natural Language Engineering **2** (1996) 15–59
13. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning **29** (1997) 103–130
14. Friedman, J.H.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery **1** (1997) 55–77
15. Mladenić, D., Grobelnik, M.: Word sequences as features in text-learning. In: Proc. 17th Electrotechnical and Computer Science Conference (ERK98), Ljubljana, Slovenia (1998)
16. Gómez-Hidalgo, J.M., de Buenaga Rodríguez, M.: Integrating a lexical database and a training collection for text categorization. In: ACL/EACL-97 Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications. (1997) 39–44
17. Dhillon, I.S., Mallela, S., Kumar, R.: A divisive information-theoretic feature clustering algorithm for text classification. Journal of Machine Learning Research **3** (2003) 1265–1287
18. Torkkola, K.: Linear discriminant analysis in document classification. In: IEEE ICDM-2001 Workshop on Text Mining (TextDM'2001), San Jose, CA (2001)
19. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.: Tackling the poor assumptions of Naive Bayes text classifiers. In Fawcett, T., Mishra, N., eds.: Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington, D.C., AAAI Press (2003) 616–623
20. Kim, S.B., Rim, H.C., Yook, D., Lim, H.S.: Effective methods for improving Naive Bayes text classifiers. In Ishizuka, M., Sattar, A., eds.: Proc. 7th Pacific Rim International Conference on Artificial Intelligence. Volume 2417 of Lecture Notes in Artificial Intelligence., Heidelberg, Springer (2002) 414–423
21. Eyheramendy, S., Lewis, D.D., Madigan, D.: On the Naive Bayes model for text categorization. In Bishop, C.M., Frey, B.J., eds.: AI & Statistics 2003: Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics. (2003) 332–339
22. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Proc. 14th International Conference on Machine Learning (ICML-97). (1997) 412–420
23. Forman, G.: An extensive empirical study of feature selection metrics for text classification. Journal of Machine Learning Research **3** (2003) 1289–1305
24. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley, New York (1991)
25. Bennett, P.N.: Assessing the calibration of Naive Bayes' posterior estimates. Technical Report CMU-CS-00-155, School of Computer Science, Carnegie Mellon University (2000)
26. Apté, C., Damerau, F., Weiss, S.M.: Towards language independent automated learning of text categorization models. In: Proc. 17th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94). (1994) 23–30