Submitted to SIGIR 98:

# Training Context-Insensitive versus Context-Sensitive Text Classifiers using Small Data Sets

Ran Bachrach Ran El-Yaniv Noam Slonim
Naftali Tishby

Institute of Computer Science
The Hebrew University, Jerusalem 91904, Israel
email: {ranb,ranni,noamm,tishby}@cs.huji.ac.il

January, 1998

**Abstract**

Recent studies of supervised learning algorithms for text classifiers employ (very) large data sets for training and evaluation. Nevertheless, there are many important situations that cannot be faithfully represented by such large data sets and are characterized mainly by inherently small training sets.

In this paper we focus on three such "small" binary classification tasks: Topical Web pages classification, internet newsgroup classification and identification of authorship. For these tasks we study and compare two important families of learning algorithms: several variants of the classical Bayesian *Probability Ratio Test (PRT)* method and several variants of the recent *sleeping experts (SE)* algorithm. Among the SE variants are *context-sensitive* algorithms that attempt to model a text using higher order statistics of the words.

Our results indicate that for these three test cases our *context-insensitive* algorithms perform sufficiently well for practical applications. Moreover, these algorithms seem to be superior to the context-sensitive algorithms.

A possible explanation is that the small training sets in our tests do not allow for sufficient sampling needed to properly train the context-sensitive algorithms. In addition, it seems that the SE algorithm is more suitable for classification according to subject matter than according to writing style.

# 1 Introduction

Algorithms for learning text binary classifiers are of fundamental importance for many information retrieval (IR) tasks such as text categorization, retrieval, routing and filtering [23]. Some of these required tasks are of critical importance for controlling the flow of data in the modern user's environment [13]. A fundamental topic of interest in IR is the experimental evaluation of supervised algorithms for learning text classifiers. Such evaluations necessarily include data sets that should faithfully represent the IR task in question.

Much of the recent experimental research on supervised learning for IR tasks utilize "large" data sets, in the sense that they contain large number (e.g. $10^4 - 10^5$) of documents in the training set. Such data sets are intended to be used for predicting the performance of IR systems coping with very large data bases, such as MEDLINE.

On the other hand, there are numerous practical text classification tasks that are associated with inherently limited data sets. Such tasks cannot be appropriately evaluated on large data sets.[1] For example, when designing a system for filtering relevant information from the Web on some arbitrary topic of interest, one must accommodate the constraint that a user cannot provide feedback for a large number of documents. This means that the number of documents in the training set is limited (e.g. order of hundreds). Another example is the *author identification problem* [15]. Here, the number of documents in the training set is bounded by the number of documents produced by the authors in question, which in many cases is small.

In this paper we focus on three such, relatively small, data sets that we believe represent binary classification tasks of practical importance. In particular we consider (i) Topical Web pages classification - representing Web pages filtering suited for specific information needs; (ii) Internet newsgroup classification - resembling routing of topical email messages; and (iii) A synthetic authorship determination problem instance, as a test case for the general author identification problem.

For these particular data sets we selected two methods that, in a sense, represent two extremes (although both can be viewed as linear classifiers). The classical Bayesian *Probability Ratio Test (PRT)* [35, 28] and the recently proposed *sleeping experts (SE)* algorithm [6, 11]. The PRT is a well-known and well explored statistical method that is often referred and used as a "benchmark" method, both because it yields reasonable results on various data sets [28, 33] and because it yields stable algorithms that can be used for testing and evaluating other learning issues, such as sampling techniques [24] and feature selection [21]. The SE algorithm represents a new powerful and promising approach for learning linear classifiers. SE algorithms combine the "advice" of a large number of "experts" each of which is typically a simple predictor. The special choice of experts defines the particular SE algorithm version. In a recent work by Cohen and Singer [6] a particular kind of experts was developed that allows for a natural selection of "contextual features" which capture higher order word statistics for the representation of a document. This work shows that these context-sensitive variants of SE improve the performance for large corpora classification tasks. On the other hand, the PRT method is typically applied on single word-based (context-insensitive) representations of the document. Previous attempts to extend the method to higher order word-statistics did not succeed to achieve significant improvements [22, 5]. Notice that both the PRT and SE methods have good theoretical understanding [9, 11] as well as experimental support [21, 24, 33, 4, 6].

The effectiveness of using higher order statistics critically depends on sufficient sampling. That is, when the number of parameters to estimate is of higher orders of magnitude we need larger corpora for their estimation. Therefore, it is interesting to check context-sensitive algorithms, such as the SE variant mentioned above also on smaller corpora, such as those we use.

[1] Hull writes: *Previous versions of the TREC filtering task have been viewed as unrealistic because they provide too much training data* [13].

With this goal in mind we implemented and examined the context-sensitive SE algorithm versus the following context-insensitive algorithms: (i) several variants of the Bayesian PRT method; (ii) a context-insensitive SE variant. Since the SE algorithm is in general a parameterized method, to obtain more conclusive results for our data sets we examined several variants of this algorithm, differing by the various assignments of the free parameters.

Our main results suggest that for smaller corpora, context-sensitive algorithms can suffer from under sampling and thus achieve inferior performance to simpler, context-insensitive methods. In all our tests, the context-insensitive SE variant outperformed the context-sensitive SE variant. Also, we found that the PRT algorithm was significantly more accurate than all SE variants on the author identification test, and slightly more accurate on the newsgroup test. On the Web pages test the context-insensitive SE variant achieved the best results. A possible explanation of these results, considering the nature of our data sets, is that the SE algorithm is more suitable for classification of "subject matters" whereas PRT is better for classification according to "writing style."

In addition, we are proposing an online implementation of the PRT method which uses only a prefix of the document without reducing the overall performance. This variant is highly relevant for applications that require fast response time such as IR applications for data on the Web.

### 1.1 Relation to Other work

Our PRT algorithm is similar to the algorithm used by Maron [28]. The main difference is that Maron used a small number of features, manually selected, while we use the full document vocabulary. Other variants of this method were used in [12, 21, 24, 33].

The main SE algorithm we examine was recently introduced by Cohen and Singer [6]. In addition, we introduce a novel context-sensitive variant of the algorithm and a feature reduction mechanism.

The datasets we used represent generic text classification problems. Web pages filtering is relatively new but currently intensively explored problem. For example, Pazzani *et al.* [33] address this problem via learning user profiles. Armstrong *et al.* [2] developed WebWatcher, a system assisting users seeking information on the Web (see also [32]). Newsgroup related problems are investigated in [30] that proposes a method for Automatic E-mail classification. Krulwich and Burkey [17] address newsgroup message categorization and Lang [18] proposes to learn user interests and filter news messages accordingly. Other authors write about similar problems (see e.g. [27])

The author identification problem was previously addressed by a variety of statistical techniques with various document representations. Those include: character $n$-grams frequencies (e.g. in [19, 16] $n < 4$), frequencies of special sets of "function" words (e.g., *but, by, for,* etc.) [29], various stylistic features (e.g., sentence length) [10], and so on. Using such representations a number of standard techniques can be used, such as principle component analysis [19], cluster analysis [10] and neural networks [29, 31, 16]. As far as we know no attempt have been made to use the frequencies of all distinct words, for this problem, as we do (PRT). The application of the SE method to this problem is also new.

Our study also attempts to provide a better understanding of the differences between these problem classes and to identify the appropriate statistical approach to each task.

## 2 The two learning algorithms

In this section we define our (supervised) learning algorithms: word-based Bayesian PRT and Sleeping Experts. We consider each *document* (or a *sample*) $d$ as a sequence of "words" $d = w_1, w_2, \ldots, w_n$ where each *word* is defined as a maximal sequence of non-blank characters. Let $\mathcal{D}$ be the set of all possible documents. A learning algorithm receives as input a *training set* $T = \{(d_1, \ell_1), (d_2, \ell_2), \ldots, (d_N, \ell_N)\}$ of

$N$ *labeled* documents where $\ell_i = \ell_i(d_i) \in \{0,1\}$ is the label of the $i$th document $d_i$. The learning algorithm then construct a mapping from the set of documents $\mathcal{D}$ to $\{0,1\}$ (i.e. a binary document classifier).

## 2.1 The Bayesian PRT classifier

This algorithm is a straightforward application of the Wald's *probability ratio test (PRT)* [35]. Given a training set $T$, for each word $w$ we can estimate its occurrence probability, $p(w|c)$, for each of the categories $c \in \{0,1\}$ as follows [28],

$$p(w|c) = \frac{\frac{1}{2} + \sum_{d\,:\,d\in T, \ell(d)=c} n(w|d)}{\sum_{d\,:\,d\in T, \ell(d)=c} |d|}$$

where $n(w|d)$ is the number of occurrences of the word $w$ in the document $d$ and $|d|$ is total number of words in $d$. Notice that an alternative approach (see Lewis [23]) is to take

$$p(w|c) = \frac{n_c(d|w)}{\sum_w n_c(d|w)}$$

where $n_c(d|w)$ is the number of documents $d$ with $\ell(d) = c$ each containing at least one occurrence of the word $w$. The underlying assumption when using this alternative method is that the generating source produces blocks of words (i.e. the documents) whereas in the first method the underlying assumption is that the source generates a stream of words.

By Bayes rule $p(c|w)$ is proportional to $p(w|c) \cdot P_0(c)$ and we take

$$P_0(c) = \frac{\sum_w \sum_{d\,:\,d\in T, \ell(d)=c} n(w|d)}{\sum_w \sum_{d\,:\,d\in T} n(w|d)}.$$

Here again one can consider an alternative document-based definition of the prior, $P_0(c) = \frac{|\{d\,:\,d\in T, \ell(d)=c\}|}{|T|}$. Preliminary comparison between the word-stream (first) and the document-based (second) methods significantly favored the first, for which the segmentation into documents has no statistical meaning. This was therefore the method we used.

The learning algorithm works as follows. During the training phase the probabilities $P_0(c)$ and $p(w|c)$, for each word $w$ and category $c$, are estimated. Then, in order to classify out-of-sample (new) document $d$ the algorithm computes

$$R(d) = \log \frac{P_0(1)}{P_0(0)} + \sum_{w\in d} \log \frac{p(w|1)}{p(w|0)}, \tag{1}$$

and decides

$$\ell(d) = \begin{cases} 1, & \text{if } R(d) \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Under the assumptions that the words are independent observations of some probability distributions (i.i.d.), and that the training set is large enough for a proper estimation of these distributions, the PRT algorithm is an optimal classifier [9]. Although this independence assumption is obviously violated in natural language text (see discussions in Cooper [8] and Lewis [23]), variants of this algorithm have been applied successfully in variety of IR tasks (see e.g. [28, 12, 21, 24, 33]).

In addition to the basic algorithm (*pure PRT*) we considered its following variations.

• *Sequential application.* In the classical PRT hypothesis testing procedure [35, 9], one specifies a significance level parameter $\varepsilon$ which prescribes two thresholds $U(\varepsilon) = \log \frac{\varepsilon}{1-\varepsilon}$ and $L(\varepsilon) = \log \frac{1-\varepsilon}{\varepsilon}$.[2]

[2]It is possible to use two *different* error parameters, $\varepsilon_1$ and $\varepsilon_2$, to control for the *false reject* and *false acceptance* levels for each category [35, 9]. In our experiments we use *equal error-rate test*, $\varepsilon = \varepsilon_1 = \varepsilon_2$.

Given a document $d = w_1, w_2, \ldots, w_n$, one computes the summation (1) over only the prefix of the document, $w_1, w_2, \ldots, w_i$, such that $i$ is the minimal index with $R(d)$ larger than $U(\varepsilon)$ or lower than $L(\varepsilon)$. The advantage of this method can be substantial for classification tasks in domains such as the internet, (suffering from network delays) to save communication time.

• *Feature selection.* Applying a feature selection mechanism may be effective for increasing the accuracy via noise reduction, avoiding overfitting and/or for improving time and space complexity (via smaller dictionaries) [22, 34]. We use the following feature selection mechanism. Define the *dictionary of category* 1 to be the set of ordered pairs $\{(p(w|1), p(w|0)) \ : \ p(w|1) > p(w|0)\}$. The dictionary of category 0 is the complementary set of pairs.

Based on Lewis and Gale [24] we consider the following feature quality measure for each word $w$ in the dictionary of category 1, $p(w|1) \cdot \log \frac{p(w|1)}{p(w|0)}$. We sort the dictionary of category 1 w.r.t. this quantity. The dictionary of category 0 is similarly sorted. We reduce each dictionary by dropping a fraction $\gamma$ of the lowest quality pairs.

• *Threshold optimization.* It is often hard to estimate priors correctly. Instead of estimating the priors $P_0(c)$ and using a threshold $t = 0$ in equation (2), we also consider an alternative decision method where the threshold $t$ is estimated on the training set and the value of $R(d)$ in equation (1) is computed without the first term. In order to achieve a desirable objective such as accuracy, precision, break-even, etc., we can calculate $R(d)$ for every document $d$ in the training set, then sort and store these weights in a table $T_R$. The value of the effectiveness measure with respect to the training set is the same for any threshold $t$ in the range $[T_R(i), T_R(i+1)]$, $i = 1, 2, \ldots, N-1$. Therefore it is sufficient to check thresholds of the form $(T_R(i) + T_R(i+1))/2$, $i = 1, 2, \ldots, N-1$, plus the two boundary values $T_R(1)$ and $T_R(N)$. This way we evaluate and compare between only $N+1$ possible thresholds. Notice that it is easy to compute the error levels for a threshold $(T_R(i) + T_R(i+1))/2$ given the levels for the threshold $(T_R(i-1) + T_R(i))/2$.

## 2.2 Classifying with Sleeping Experts

Given a training set $T$, a set $E_T$ of *experts* is constructed. Each expert $e$ is a boolean function from documents to $\{0, 1\}$. We define a mapping $E$ from documents to $2^{E_T}$ so that for each document $d \in \mathcal{D}$, $E(d)$ is a subset of experts, $E(d) \subseteq E_T$. The experts in $E(d)$ are said to be *awake for* $d$ and the rest of the experts (i.e. those in $E_T - E(d)$) are *sleeping for* $d$. Given a document $d$, the set of awaked experts $E(d)$ is partitioned into two complementary subsets: $E_1(d)$, containing all *positive experts* $e$ with $e(d) = 1$ and $E_0(d)$, containing all *negative experts* $e$ with $e(d) = 0$.

We associate with each expert $e \in E_T$ a weight $g(e)$. Based on the experts' advice (weights) the algorithm computes for each document $d$ a weight $G(d)$,

$$G(d) = \frac{\sum_{e \in E_1(d)} g(e)}{\sum_{e \in E(d)} g(e)}.$$

and decides

$$\ell(d) = \begin{cases} 1, & \text{if } G(d) \geq t \\ 0, & \text{otherwise.} \end{cases}$$

where $t$ is some predefined threshold.

The SE algorithm is based on a multiplicative weight update technique [26, 14]. Using competitive analysis it is possible to show that the number of errors made by the SE algorithm is within a constant factor of the optimal linear combination of experts [11].

Of course, in order to apply this scheme one must determine and fix the free parameters which are: (i) The set of experts $E_T$; (ii) The weights assigned to the experts; and (iii) The threshold $t$.

• *The experts.* Cohen and Singer [6] consider the following *sparse k-phrase* experts where one constructs an expert for each phrase of $k$ consecutive words, with all possible "don't care" gaps. For example, The

phrase "sleeping experts algorithm" appearing in a document yields the following seven sparse 3-phrases (in which the symbol $*$ stands for "don't care"):

```
[sleeping experts algorithm]
[   *     experts algorithm]
[sleeping    *    algorithm]
[sleeping experts     *    ]
[   *        *    algorithm]
[sleeping    *        *    ]
[   *     experts     *    ]
```

Note that the set of all 1-phrases of a document is simply the set of its words.

The entire set $E_T$ of experts is extracted from all sparse $k$-phrases seen in the training set $T$ ([6] consider $k = 1, 2, 3, 4$). For each $k$-phrase feature $f$ in a document $d$, two experts, $e_{1,f}$ and $e_{0,f}$, are generated so that $e_{1,f}$ is a positive expert for $d$ with $e_{1,f}(d) = 1$, and $e_{0,f}$ is a negative one, $e_{0,f}(d) = 0$.

Intuitively, with larger values of $k$, the experts constructed from sparse $k$-phrases capture higher order word-based statistics of the document. On the other hand, computational considerations only allow for small values of $k$. Specifically, in the worst case there are $O(\frac{2^k}{k}|d|)$ experts for each document $d$ (any $k$-gram yields up to $2^k$ sparse $k$-phrases).

We consider two variants of the sleeping experts with 1-phrases and sparse 4-phrases. In addition we consider a novel type of experts called *k-proximity pairs* where we generate two (positive and negative) experts for each pair of words located in proximity of $k$ consecutive words in the document. When using this method there are, in the worst case, $O(k|d|)$ experts for a document $d$. Hence, by limiting ourselves to such ordered pairs (instead of all sparse $k$-phrases) we allow for the consideration of large contexts $k$ while the number of experts is not exponential (and in fact linear) in $k$. For this type of experts we only have preliminary experimental results.

• *Weight training.* The goal of the sleeping experts algorithm is to appropriately train the weights associated with experts. We experiment with two weight training methods. The first is the well-known Winnow algorithm (see Blum [3, 4]. Initially, all experts are given a weight of 1. For every sample $d$ in the training set, if $d$ is positive (i.e $\ell(d) = 1$) then each weight of an expert $e \in E_0(d)$ is reduced by multiplying it by a constant factor $\beta < 1$ called the *learning rate*. Similarly, when $d$ is a negative sample, each weight of an expert $e \in E_1(d)$ is reduced by multiplying it by $\beta$.

In the second learning method one uses an additional normalization [6]. The purpose of the normalization is to keep $\sum_{e \in E(d)} g(e)$ constant before and after "punishing" the experts that predicted wrongly. Thus, the weight training algorithm is:

- $N := \sum_{e \in E(d)} g(e)$
- If $d$ is positive (i.e. $\ell(d) = 1$), for every $e \in E_0(d)$, $g(e) := \beta \cdot g(e)$
- Otherwise ($d$ is negative) for every $e \in E_1(d)$, $g(e) = \beta \cdot g(e)$
- $N' := \sum_{e \in E(d)} g(e)$
- For every $e \in E(d)$, $g(e) := \frac{N}{N'} \cdot g(e)$

• *Threshold determination.* In [6] the threshold was taken to be 1/2. When the priors of the two categories is not 1/2 this method can lead to low performance. Therefore we applied the same threshold optimization method as defined for the PRT algorithm. That is, we calculate $G(d)$ for every document $d$ in the training set, then sort these values and find the optimal threshold as previously discussed.

• *Feature selection.* We also performed initial experiments with the following feature selection method. For each feature $f$ (i.e. $f$ is a sparse $k$-phrase or a $k$-proximity pair) we have two weights $g_0(f)$ and $g_1(f)$ associated with the negative and positive experts of $f$, respectively. The *score* $s(f)$ of feature $f$ is then defined to be $s(f) = |g_0(f) - g_1(f)|$. Thus, high scores reflect the fact that one but not both of

the experts associated with $f$ was "punished" many times. We sort the features according to their scores and then discard a fraction $\gamma$ of the least scoring features.
• *Representation.* Since our experts are constructed from phrases, here again (as it is for PRT) there are two representation approaches. In the first, document-based approach, an expert is activated (awaked) only on the first occurrence of its corresponding phrase in the document. In the second, stream-based approach, an expert is activated for each occurrence of its corresponding phrase in the document. Based on preliminary tests we used the document-based approach.

# 3 Three datasets

We considered three datasets consisting of Web pages, newsgroup messages, and English literature. Our tests with the Web and newsgroup data sets are intended to test the adequacy of the learning algorithms with respect to practical retrieval tasks such as Web pages filtering and news (and mail) message routing. For these particular tasks one cannot expect a user to train the learning algorithm by classifying a large number of samples. Therefore, our tests focus on performance with respect to relatively small number of samples in the training sets. This is in contrast to other works that use standard large data sets such as the TREC evaluation data sets (see e.g. [1, 13]) or Reuters-21578 [20].

The third data set, English literature, tests another important classification task, namely author identification [15]. The main difference between the first two tests and the third one is that the first two are generally concerned with discriminating between subject matters (e.g. distinguishing between newsgroups), while the third test is concerned with discriminating between writing style (and not necessarily between subject matters).

All data were preprocessed as follows. The texts were lower cased, each digit character was converted to "$" and each other non-alphanumeric symbol was converted to "*".

## 3.1 Web pages test

We applied the following query to several search engines on the Web.

```
(conference OR workshop OR symposium) AND comp AND paper
```

¿From the returning URLs we selected the first 641 pages with average length of 1164 words. Each page was manually categorized to one of the following two categories: URLs pointing to calls for computer science conferences with paper submission (27% of the data set) and its complement subset of URLs. We filtered out HTML commands.[3]

Since our algorithms were found to be order sensitive we followed Lewis *et al.* [25] and selected 10 random permutations of the entire list of URLs thus creating 10 random training sets (each of 450 samples) and 10 testing sets (each of 191 samples). In the experiments performed we averaged the results over these 10 instances.

## 3.2 Newsgroup test

We selected all 1684 Usenet messages (on July 26th 1997) from the newsgroups `sci.med.nutrition` (1050 messages) and `sci.med.pharmacy` (634 messages). The average message length was 297 words. We deleted the identifying header from each message. As in the Web pages data set we created 10 random

[3]We found that the performance can be minorly improved (by approximately 1% in accuracy) using one of the following alternative representations that increased the space complexity: (i) using *flatenned* Web pages in which each additional Web page addressed by a link in the original page is inserted into the original page; (ii) retaining the HTML tags.

instances each having 1200 messages for training and 485 for testing. Note that this collection of messages contains quite a few non-related "noisy" messages.

### 3.3 Authorship test

Clearly, when designing an author identification test, one is limited by the fact that each author has relatively small number of documents, while some of the documents may be relatively long. Therefore, in order to obtain a sufficiently large number of documents in the training and testing sets we "synthetically" split long documents (books) into smaller, fixed sized segments. Such artificial splitting of the documents cannot be done when the question is not "stylistic" as in the Web pages data set.

We selected two authors: R. Kipling and E.R. Burroughs. Then designed the following three datasets (and tests) to check the influence of (i) using a variety of original documents (books) for each author; and (ii) using different (synthetic) document sizes.

**Test (a):** As a training set we used 500 samples of 500 words each, randomly selected from four books by R. Kipling.[4] 500 samples of 500 words each, randomly selected from four books by E.R. Burroughs.[5] As a test set we used: 190 samples of 500 words each, randomly selected from two books by R. Kipling.[6] 190 samples of 500 words each, randomly selected from two books by E.R. Burroughs.[7]

**Test (b):** 250 samples, 500-words each (randomly selected) from *Kim* (Kipling) and *The Beasts of Tarzan* (Burroughs) as training set and 200 500-words samples from *The Jungle Book* (Kipling) and *The Gods of Mars* (Burroughs) for testing.

**Test (c):** Same books as in (b) but 100-word blocks (1200 for training and 1000 for testing).

Note that in tests (b) and (c) we selected books roughly dealing with the same subject matter (*The Beasts of Tarzan* by Burroughs, and *The Jungle Book* by Kipling) using the first of these "jungle" books for the Burroughs training set, and the second for the Kipling test set. Thus, this test requires discrimination between "writing style" and not only between "subject matters". Also, we insured, for each author, that there will be no trivial similarity between the training and test sets (e.g. same character names etc.)

## 4 Experimental results

Our main goal was to compare the PRT algorithm to the SE algorithm with respect to the above data sets. We compared several variants for each method, trying to understand the influence of different parameters on the overall performance. We are concerned with classification tasks where each document belongs to one of two mutually exclusive categories. Hence, we find it informative to measure performance by total *accuracy*; that is, the ratio of number of test set documents successfully classified over the total number of test set documents. Other effectiveness measures such as precision, recall and F-measure [25] did not qualitatively affect our results. We experimented with successive values of sample sizes and measured the average *accuracy* of each algorithm for each sample size. These measurements allow for "close-up" examination of the learning quality for varied training set sizes, in particular for very small training sets.

In the following subsections we first present the results obtained for each of the algorithm families individually and then compare the two methods.

[4] *Kim*, *Puck of Pook's Hill*, *American Notes* and *Rewards & Fairies*.
[5] *The Beasts of Tarzan*, *The People That Time Forgot*, *At the Earth's Core*, and *The Mucker*.
[6] *The Jungle Book* and *Verses 1889-1896*.
[7] *The Oakdale Affair* and *The Gods of Mars*.

### 4.1 The PRT algorithm

As mentioned above, we checked the following variants of the PRT algorithm: (1) sequential PRT (2) feature selection (3) threshold optimization (4) pure PRT. In general, all these variants achieved similar accuracy performance with pure PRT slightly outperforming all other variants in all tests.

• *Sequential application.* The sequential PRT achieved almost identical performance to pure PRT across all tests (its accuracy is deteriorated by no more than 1%). The main advantage of this algorithm is that it can classify a document by examining only its portion. This fact is depicted in Figure 1. This graph plots the percentage of the document used for classification as a function of the training set size. For larger training sets the algorithm can classify a document $d$ while using a shorter prefix of $d$ (Figure 1) and achieving better overall accuracy (Figure 2). For example, it is possible to achieve 91% accuracy after training with 400 documents while consuming less than 20% of each test document. The results for all other tests were qualitatively similar. Note that there is a tradeoff between the total accuracy and the computation speed. This tradeoff is controlled by the value of tolerance parameter $\varepsilon$ (smaller $\varepsilon$ implies higher accuracy but slower speed and vice versa).

Figure 1: Sequential PRT (Web pages data set): portion of document used for classification as a function of training set size (averaged over 10 independent instances). The tolerance parameter (level of significance) is $\varepsilon = 10^{-15}$. The original average file size is 1164 words

• *Feature selection.* One may expect that applying the feature selection mechanism would be effective for increasing the accuracy (via noise reduction) and for higher time and space efficiency (via smaller dictionaries). We tested several values of the cut-off parameter $\gamma$ ranging in $[0.1, 0.9]$. and found that the accuracy performance does not improve for all tested values in all the tests. However, we found that we can use significantly smaller fractions of the dictionaries (e.g. $\gamma = 0.8$) while not decreasing the accuracy performance by more than 1% on average.

• *Threshold optimization.* This variant achieved in general inferior average performance and exhibited higher variance, compared to the pure PRT. This could mean that our estimation of priors were relatively accurate. Note that for other effectiveness measures (such as precision) it is unclear how to use the priors

in equation (1) while the threshold optimization method is a natural solution.

### 4.2 The Sleeping experts algorithm

In general, we have tested the SE algorithm with (sparse) 1-phrases (called SE-1), and with sparse 4-phrases (called SE-4). SE-1 outperforms SE-4 in all tests. A possible explanation is that the larger number of free parameters in the SE-4 requires larger number of samples to train. This explanation is supported by the learning curves of Figure 2 where the SE-4 attains the same accuracy as SE-1, only for training sets of sizes larger than 400 documents. Similarly, in the newsgroup data set, one can see a "convergence" in the learning curves of SE-4 to SE-1 (see Figure 3).

• *Proximity pairs.* We have also tested an SE variant using 10-proximity pairs together with sparse 4-phrases experts (called SE-10). Due to the enormous number of experts we have limited the number of experts and took just the first million. On the Web pages data set this algorithm achieved the best results that were marginally superior than the SE-1. It is important to note that SE-1 used about 5% of the experts' number used by SE-10. On the other data sets SE-10 was significantly inferior to SE-1.

• *Feature selection.* For SE-10 we also checked our feature selection method (on the Web pages test only). We found that we can leave only 400 features, yet retaining similar accuracy (with deterioration of less then 1% on average). Note that this feature reduction is of 4 orders of magnitude. Interestingly, among the 400 top scoring experts over 300 are 10-proximity pairs.[8]

• *Representation.* For the Web pages test the document-based representation yields superior results for all SE variants so we used it for all experiments. Yet, in the authorship test, where we "artificially" split the documents (books), it seems that the stream-based representation is more natural (because we are looking for "style" in which every occurrence of the same word can be relevant). Since PRT wins on this data set and since its representation is stream-based, we tested the stream-based representation for SE-1 also on the the authorship test. The results where similar to the document-based representation.

• *Threshold optimization.* Testing the influence of the threshold optimization over the Web pages and newsgroup data sets shows considerable improvement over constant threshold of 0.5. The difference is both in the overall performance (sometimes up to 10% gain in accuracy) and in reducing the sensitivity of the algorithm to different training sets. On the Authorship test however the constant threshold $t = 0.5$ performed better. (In all authorship tests, in contrast to the other tests, the categories are balanced).

• *Learning rates.* Following [6] we tested learning rates $\beta \in [0.1, 0.5]$ (only over Web pages test). We found that smaller values of $\beta$ improve the performance on small training sets (i.e. learn faster) but on large training sets, larger (more conservative) learning rates perform better. The difference in accuracy, in both sides of the the range $[0.1, 0.5]$ could reach 2-3% on average. Also, when using small values of $\beta$, the variation of the results on different random instances of the training sets tends to be larger. In all reported results $\beta = 0.4$.

• *The Winnow algorithm.* We performed a preliminary study of the Winnow algorithm only on the Web pages test. We found that Winnow did not perform as well as the other SE variants and exhibited an almost flat learning curve of 85% accuracy on most training-set sizes.

### 4.3 Comparison of the methods

In all data sets in our experiment, context-insensitive algorithms were the best choice between the algorithms tested. In the Authorship tests and Newsgroup test, the PRT achieved the best accuracy. In the Web pages test, although SE-10 achieved best accuracy, it was only slightly (+0.5%) better then the

[8] By inspection of these top scoring features it seems that their conjunction could be served as a basis for a sharper query (to search engines) for the concept being learned (cf. Cohen and Singer [7]).

results for SE-1. On the other hand, SE-1 is much better from efficiency considerations (in speed and memory).

Our results indicate that the data sets we used are qualitatively different. It seems that the authorship tests versus the Web pages test, represent essentially different tasks, while the Newsgroup test have something from both. On the authorship tests the PRT algorithm is significantly better then all the SE variants. In the Newsgroup test PRT is slightly better then SE variants, and in the Web pages test SE-1 is a favorite. This results support our initial assumption on the different nature of the data sets, i.e. authorship tests dealing with "writing style", Web pages test dealing with "subject matters", and Newsgroup test dealing with both questions.

#### 4.3.1 The Web pages test

Figure 2: Web pages data set. Average learning curves over 10 instances

SE-1 is the best performer on this data set. In particular, within 100 samples it achieves 91% accuracy. This result suggests that SE-1 can be effective in practical Web pages filtering tasks. PRT algorithm is not far behind, and SE-4 learns much slower but equalize with SE-1 (and winning PRT) after exhausting the training set.

#### 4.3.2 Newsgroups test

Compared to the Web pages and authorship tests (a) and (b), the newsgroup test is harder in the sense that all algorithms achieve lower performance on it. One plausible explanation is that this data set contains many noisy messages (such as "spam" messages), as opposed to the others. At the outset it may appear that this test is a subject-matter discrimination test. Nevertheless, the subjects in both

Figure 3: Newsgroups data set. Average learning curves over 10 instances

these newsgroups are not significantly distinct (pharmacy versus nutrition). Perhaps more important is the fact that each newsgroup is also affected by writing style, as a result of "Re: Re: Re: Re:" chains, development of community (newsgroup subscribers) "slang", and so on. This observations supports the finding that PRT is the winner in this test, but with only minor advantage.

#### 4.3.3 Authorship tests

First note that with the appropriate training set the performance we achieve (say, by PRT) are very impressive. For example, only 250 "documents" each of 500 words (overall number of 60,000 words, taken from four books per author) are sufficient for the PRT algorithm to achieve 97% accuracy on the test set containing two (different) books for each author. This finding suggests that the PRT algorithm is a very interesting tool for practical authorship research.

All algorithms perform significantly better on test (a) versus test (b), even when considering the same training set sizes (for example, consider 200 samples in both figures 4 and 5). This could reflect the fact, that the "confusion" between "subject matters" is much less dominant in test (a). Further more, the writing style is probably better sampled when training on several books for each author.

SE algorithms exhibits decreasing learning curves in authorship test (c) (see Figure 6), and flat learning curves in test (b) (Figure 5). Note that these tests were designed to confuse algorithms that learn "subject matter" and not "style". Therefore we suggest that the SE algorithms tested are "subject oriented". Another support for this assumption is that comparing the appearance probability of frequent words (such as "the," "a,") is a common discriminating method between writers [29]. In fact, we found that for PRT such words helped in the classification. On the other hand, for the SE algorithms, because

Figure 4: Learning curves for author identification test (a): long test samples from several books for each author. Average over 5 instances. SE with document-based representation

of the multiplicative update mechanism, such frequent words vanished (they are assigned very small and insignificant weights).

Tests (b) and (c) differ only in sample sizes. Although PRT shows significantly inferior results in test (c), even when the total number of words in the training sets is equal. this difference is due to the shorter test documents in test (c), containing less information in the decision making stage. Note that the training stage is similar in both tests for the PRT using stream-based representation. The SE variants suffered from the same effect, and even more due to effect of the multiplicative updates, correlative to the number of documents in the training set.

## 5 Conclusions

Application of statistical methods often rely on worst case and asymptotic theoretical analysis which may provide little information about the typical small-sample performance of those techniques. In this paper we experimentally investigated two families of text categorization methods, the first is based on the classical Bayesian sequential probability ratio test (PRT) and the second on the more recent ideas of sleeping experts (SE) with multiplicative update rules (Winnow). While for both methods there is ample of asymptotic theoretical understanding, their actual performance on realistic tasks with limited training data is very difficult to predict.

Another goal of this paper is to address another related issue, namely, the role of context sensitive information for text classification. While it is clear that such information, beyond single-word statistics, can asymptotically provide much more powerful discrimination, the performance of context sensitive

Figure 5: Learning curves for author identification test (b): long test samples from one book of each author. Average over 5 instances. SE with document-based representation

methods for small amounts of training data is again hard to be estimated. The main reason for this is essentially old "bias-variance dilemma" and the unreliable estimation of the many more context sensitive parameters.

The main result of our study is that for at least three simple generic, small-training set, text classification tasks that we considered, the context-insensitive methods outperform the context sensitive ones. Moreover, for two out of our three tests, newsgroups classification and author-identification, the simple Bayesian PRT significantly outperforms all variants of the sleeping-experts algorithm that we examined.

Various interesting directions are left for future research:

- This paper suggests that, for small corpora, context-insensitive learning algorithms are superior. Yet, we are familiar with other interesting methods to exploit context, which have not been thoroughly tested. For instance, our initial experiments with the $k$-proximity pairs experts (SE-10) yield promising results. Also, we have not tested sparse 2- and 3-phrases that could have benefited from context yet suffer less than sparse 4-phrases from under sampling.

- Feature selection is of special importance to context-sensitive algorithms in which the size of the feature set can be very large. Our preliminary experiments applied to the SE-10 variant are very promising (reduction of 4 orders of magnitude while retaining similar accuracy), but yet not conclusive.

- Although there is a variety of linguistic and statistical methods for author identification, for the purpose of literary research, there is always the need for stronger, well-founded methods [29]. The

Figure 6: Learning curves for author identification test (c): short test samples from one book of each author. Average over 5 instances. SE with document-based representation

application of the PRT in this domain is novel, and based on our experiments, could serve as a powerful, well-founded and simple to implement tool.

## Acknowledgments

We sincerely thank Dana Atzil, Efrat Haskin, Debby Hess, Yakov Kogan, Shuki Sagiv, and Orya Tishby for their support, helpful comments and more.

## References

[1] The Text REtrieval Conference (TREC). Home page: `http://trec.nist.gov/`.

[2] R. Armstrong, D. Joachims, and T. Mitchell. WebWatcher: A Learning Apprentice for the World Wide Web. In *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, Stanford*, March 1995. Available online: `http://www.cs.cmu.edu/afs/cs/project/theo-6/web-agent/www/webagentsplus.ps.Z`.

[3] A. Blum. On-line algorithms in machine learning. Available online: `http://www.cs.cmu.edu/afs/cs.cmu.edu/user/avrim/www/Papers/pubs.html`.

[4] A. Blum. Empirical support for winnow and weighted majority algorithms: results on a calendar scheduling domain. In *Machine Learning: Proceedings of the 12th International Conference*, 1995.

Available online:
`http://www.cs.cmu.edu/afs/cs.cmu.edu/user/avrim/www/Papers/pubs.html`.

[5] K.W. Church. One term or two? In *ACM SIGIR 95*, pages 310–318, 1995.

[6] W.W. Cohen and Y. Singer. Context-sensitive learning methods for text categorization. In *ACM SIGIR 96*, pages 307–315, 1996. Available online:
`http://www.research.att.com/ singer/publications.html`.

[7] W.W. Cohen and Y. Singer. Learning to query the web. In *AAAI Workshop on Internet-Based Information Systems*, 1996. Available online:
`http://www.research.att.com/ singer/publications.html`.

[8] W.S. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. In *ACM SIGIR 91*, pages 57–61, 1991.

[9] T.M. Cover and J.A. Thomas. *Elements of Information Theory.* John Wiley & Sons, Inc., 1991.

[10] P. Dixon and D. Mannion. Goldsmith's periodical essays: A statistical analysis of eleven doubtful cases. *Literary and Linguistic Computing*, 8(1):1–19, 1993.

[11] Y. Freund, R.E. Schapire, Y. Singer, and M.K. Warmuth. Using and combining predictors that specialize. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, 1997.

[12] N. Fuhr. Models for retrieval with probabilistic indexing. *Information Processing & Management*, 25(1):55–72, 1989.

[13] D.A. Hull. The trec-6 filtering track: Description and analysis. In *The 6th Text REtrieval Conference (TREC-6)*, 1997. Available online:
`http://trec.nist.gov/pubs/trec6`.

[14] J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient decent for linear predictors. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 209–218, 1995.

[15] B. Kjell. Authorship determination using letter pair frequency features with neural network classifiers. *Literary and Linguistic Computing*, 9(2):119–124, 1994.

[16] B. Kjell, W.A. Woods, and O. Frieder. Discrimination of authorship using visualization. *Information Processing & Management*, 30(1):141–150, 1994.

[17] B. Krulwich and C. Burkey. The ContactFinder agent: Answering buletin board questions with referrals. In *Proceedings of the Thirteenth National Conference on Artificial Inteligence AAAI 96*, pages 10–15, 1996.

[18] K. Lang. News Weeder: Learning to Filter Netnews. In *Proc. of the 12th International Conference on Machine Learning ICML95*, 1995.

[19] G. Ledger and T. Merriam. Shakespeare, Fletcher, and the two noble kinsmen. *Literary and Linguistic Computing*, 9(3):235–248, 1994.

[20] D.D. Lewis. The Reuters-21578 Text Categorization Test Collection,
see `http://www.research.att.com:80/ lewis/reuters21578.html`.

[21] D.D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *ACM SIGIR 92*, pages 27–50, 1992.

[22] D.D. Lewis. Feature selection and feature extraction for text categorization. In *Speech and Natural Language: Proc. of a Workshop held at Harriman, New York, February 23–26, Morgan Kaufman*, pages 212–217, 1992.

[23] D.D. Lewis. *Representation and Learning in Information Retrieval.* PhD thesis, University of Massachusetts at Amherst, 1992.

[24] D.D. Lewis and W.A. Gale. A sequential algorithm for training text classifiers. In *ACM SIGIR 94*, pages 3–12, 1994.

[25] D.D. Lewis, R.E. Schapire, J.P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *ACM SIGIR 96*, pages 298–306, 1996.

[26] M. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[27] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):30–34, 1994.

[28] M.E. Maron. Automatic indexing: An experimental inquiry. *Journal of the ACM*, 8:404–417, 1961.

[29] R.A.J. Matthews and T.V.N. Merriam. Neural Computation in Stylometry I: An Application to the Works of Shakespeare and Fletcher. *Literary and Linguistic Computing*, 8:203–209, 1993.

[30] A.D. May. Automatic classification of e-mail messages by message type. *Journal of the American Society for Information Science*, 48(1):32–39, 1997.

[31] T.V.N. Merriam and R.A.J. Matthews. Neural Computation in Stylometry II: Application to the Works of Shakespeare and Marlowe. *Literary and Linguistic Computing*, 9(1):1–6, 1994.

[32] D. Mladeni'c. Personal WebWatcher: design and implementation. Technical Report IJS-DP-7472, University of Ljubljana, Slovenia, 1996. Available online: `http://www-ai.ijs.si/DunjaMladenic/home.html`.

[33] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Inteligence AAAI 96*, pages 54–61, 1996.

[34] H. Schutze, D.A. Hull, and J.O. Pedersen. A comparison of classifiers and document representations for routing problem. In *ACM SIGIR 95*, pages 229–237, 1995.

[35] A. Wald. *Sequential analysis.* John Wiley & Sons, Inc., 1947.