Applied Intelligence 19, 109–123, 2003

Authorship Attribution with Support Vector Machines

JOACHIM DIEDERICH
School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Q-4072, Australia

JÖRG KINDERMANN, EDDA LEOPOLD AND GERHARD PAASS
GMD—Forschungszentrum Informationstechnik, D-52754 Sankt, Augustin

Abstract. In this paper we explore the use of text-mining methods for the identification of the author of a text. We apply the support vector machine (SVM) to this problem, as it is able to cope with half a million of inputs it requires no feature selection and can process the frequency vector of all words of a text. We performed a number of experiments with texts from a German newspaper. With nearly perfect reliability the SVM was able to reject other authors and detected the target author in 60–80% of the cases. In a second experiment, we ignored nouns, verbs and adjectives and replaced them by grammatical tags and bigrams. This resulted in slightly reduced performance. Author detection with SVMs on full word forms was remarkably robust even if the author wrote about different topics.

Keywords: support vector machines, authorship identification, textmining

1. Introduction

As more and more economic and intellectual activities use the world wide web as medium, the scrutiny of the authenticity of a document becomes more and more relevant. An example is internet plagiarism. In this article we discuss advanced statistical text mining methods for authorship attribution.

Authorship attribution can be considered as a categorization problem. In contrast to other classification tasks it is not clear which features of a text should be used to classify an author. During the last decades many text attributes were proposed in the area of information retrieval, which—after careful tuning—were able to solve some cases of disputed authorship.

In this paper we apply the support vector machine (SVM) to authorship attribution. Unlike currently used classification approaches, like neural networks or decision trees, it allows for the processing of hundreds of thousands of features. This offers the opportunity to use all words of a text as inputs instead of a few hundred carefully selected characteristic words only. In similar text classification problems aiming at thematic categorization, the SVM has been shown to be quite effective [1, 2].

A SVM is able to classify a text with respect to content. In the framework of author attribution, it is not clear whether a specific topic addressed by the author or the structural or stylistic features of the authors language lead to a successful classification. To achieve some 'content-invariance' we investigated a more 'content'-free summary of a text. We took counts of grammatical tags combined with bigrams, to capture morphologic details of language patterns.

In the next section we give an overview of linguistic features used for the analysis of an author's style. The following section discusses different statistical methods for authorship assessment. It starts with statistical hypothesis tests and includes semi-parametric approaches as well as the support vector machine. As a preprocessing step for text classification the raw frequency vectors were transformed and normalized, which is discussed in the fourth section. The fifth section describes the numerical experiments of authorship

attribution using a corpus from a German newspaper. The last section summarizes the results.

2. Features Used for Stylometry

The statistical analysis of style, *stylometry*, is based on the assumption that every author's style has certain features inaccessible to conscious manipulation. These features provide the most reliable basis for the identification of an author. However, the style of an author may vary as a result of differences in topics or genre, or the personal development of the author over time. It may also be influenced by the explicit imitation of literary styles. Ideally stylometry should identify features which are invariant to these effects, but are expressive enough to discriminate an author from other writers.

Early stylometric studies introduced the idea of counting features in a text and applied this to *word lengths* and *sentence lengths* [3]. Yule [4] reported a wider variation of sentence lengths than word lengths. There are differences in sentence length for the same author, not only dependent on time but also on the genre of the text [5]. Differences parallel word length distributions in the prose and verse of the same author. Other features are counts of words beginning with a vowel or counts of words with specific lengths [6].

A powerful criterion in stylometry is the 'richness' or '*diversity*' of an author's vocabulary. Zipf observed the number of words $\alpha(f)$ which occure exactly f-times is calculated as

$$\alpha(f) = f^{\gamma},$$

where $\gamma \sim 2$ [7]. He conjectured that the parameter γ depends on the age and intelligence of an author [8]. Sichel [9] was able to successfully fit a family of compound Poisson distributions to the word frequencies of a number of authors and works in different languages. In order to remove the dependency of vocabulary size from the text length N, alternative features have been proposed. These range from the simple type-token ratio to more complex measures as Orlov's Zipf size [10]. An interesting feature is the comparison of the number of words which occur exactly j-times in the training data and the number of words which occur exactly j-times in a new text, for $j = 0, 1, \ldots$ [22]. Thisted and Efron estimated the size of Shakespeare's vocabulary by asking "How many new words would Shakespeare use if he were to write another play?"

Many studies found distinct differences in the vocabulary size of different authors, but also some differences in texts from the same author [5]. Hence these features have only limited value for authorship attribution. It is clear that a collection of different features, e.g. the vocabulary size in different word fields or the knowledge of specific words have a larger discriminatory power.

Obviously word usage highly depends on the topic of the text. For discrimination purposes we need "content-free" or *function words*. In a seminal paper Mosteller and Wallace [12] counted the use of words like 'while' and 'upon' to discriminate between possible authors. Burrows [13] developed the idea of using sets of more than fifty common high-frequency words and conducted a version of principal component analysis on the data. This technique has been successfully applied to the classical 'Federalist Papers' problem and promises large gains if more computing power is available. Binongo and Smith [14] used the frequency of occurrence of 25 prepositions to distinguish between Oscar Wilde's plays and essays.

Instead of using word counts directly one can employ features derived from words. An example is the syntactic class of words [15]. Compared to the use of syntax, word use is more easily influenced by choices which are under the conscious control of the author. As the discourse structure of texts from the same author and the corresponding vocabulary can be variable, syntax-based features can be more reliable for the purpose of authorship attribution. Charniak [16, p. 139ff] discusses other techniques for enhancing statistical language processing with syntactic information.

According to Rudman [17, p. 361] "approximately 1,000 style markers have already been isolated." There clearly is no agreement on significant style markers. It seems that in text categorization nearly all words contain some information. Joachims [18] ranked 10000 word stems of a large corpus according to their information gain with regard to some classification. It turned out that a model using features with ranks 201-500 performed nearly as well as the best features in the top 1-200, and similar to the feature set 4001-9962. Hence even features ranked lowest still contain considerable information and are somewhat relevant. In the following section we discuss statistical techniques for authorship attribution. While the conventional techniques rely on a few carefully selected features, newly developed approaches allow for the use of many hundred or even thousands of input features and alleviate the need for a careful selection.

3. Statistical Techniques for Authorship Attribution

Statistical approaches for authorship attribution start with the assumption that the composition of texts produced by each author is characterized by a probability distribution. Considering a given population of authors the attribution of a text to an author can be considered as a statistical hypothesis test or classification problem. This usually requires four distinct steps:

Feature selection	Identification of possible features for the discrimination of texts, as discussed above.
Model selection	Selection of suitable distributions or models describing the feature values.
Learning	Estimation of free parameters for different authors from available data.
Classification	Selection of a potential author for a new text.

In the next sections we discuss these aspects in more detail.

3.1. Hypothesis Tests

Hypothesis tests assume probability distributions of a known type for the different authors. If words of different types in a text are counted, a natural distribution is the multinomial distribution. Let $x \in X$ be the vector of word frequencies of texts, $q = x/\sum_i x_i$ be the corresponding relative frequency vector and π be the true probability vector. Then the following statistics may be used for goodness-of-fit test of counts [19, p. 472, 513]:

Pearson's chi square

$$X^2(q,\pi) = \sum_{i=1}^{k} \frac{n^2(q_i - \pi_i)^2}{n\pi_i} \tag{1}$$

The log-likelihood ratio statistic

$$G^2(q,\pi) = 2\sum_{i=1}^{k} nq_i \log\left(\frac{nq_i}{n\pi_i}\right) \tag{2}$$

The Freeman-Tukey goodness-of-fit statistic

$$K^2_{FT}(q,\pi) = 4\sum_{i=1}^{k} (\sqrt{nq_i} - \sqrt{n\pi_i})^2 \tag{3}$$

If q is generated according to π (null hypothesis) all three statistics have an asymptotic chi square distribution with $k-1$ degrees of freedom. This distribution has mean value $k-1$ and variance $k-1$. This means values larger than about $(k-1) + 2\sqrt{k-1}$ indicate a significant deviation. There are also two-sample versions of these statistics to compare the distribution of two independent samples. The application of these tests to authorship attribution is hampered by two factors: First the approximation underlying a test is valid only if each expected frequency is larger than about 3, which usually does not hold for a high fraction of the words in a text. Second the distribution produced by an author changes with genre and topic, which contradicts the assumption that all texts follow the same multinomial distribution.

A number of tests have been developed checking different distributional features. Starting with a characteristic x_i of the i-th sentence in an author's text, e.g. its length, the **cusum test** [20] detects significant deviances from the mean value $\bar{x}$. It establishes bounds for the test quantity $\sum_i (x_i - \bar{x})$ which are valid under specific assumptions, e.g. the independence of the terms $(x_i - \bar{x})$. These bounds allow a nice graphical representation. It was used in a number of court cases and received significant public attention [21]. However, a number of independent investigations found the method unreliable [6], as again the stability of these characteristics over multiple texts is not warranted.

A test developed by Thisted and Efron [22] analyses the diversity of an author's vocabulary. They generate the sets M_i, $i = 0, 1, 2, \ldots$, of words that occur exactly i times in the training corpus. Subsequently they determine the number of words in M_i in the corpus and a sample text and propose various significance test under the assumption that word selection of an author is a Poisson process. As for the other tests the results are mixed. Valenza [23] applied these tests to the works of Shakespeare and Marlowe and found good consistency for the Shakespeare plays but poor consistency between Shakespeare poems and plays or Marlowe's plays.

3.2. Semi-Parametric Models for Classification

The advent of powerful computers initiated the development of machine learning techniques with larger flexibility. While regression models [24] and naive Bayesian models [25] for text classification still have structural limitations, a number of more and more

versatile procedures were applied to text categorization, e.g. inductive rule learning [26], Bayesian probabilistic networks [11], multilayer perceptrons [27], radial basis function networks [28], decision trees [29], nearest neighbor classification [30], and support vector machines [1]. These models are universal approximators as they are able to approximate any functional relation arbitrarily well. They model the underlying distribution with a potentially infinite number of parameters, which are selected in such a way that the predictive performance becomes optimal. Each approach has its own strength and weaknesses, which corresponds to its representational bias, i.e. its ability to represent specific structures in an economical way.

Naive Bayesian probabilistic classifiers use the joint probabilities of words and text categories to estimate the probability of categories given a text. They use the 'naive' assumption that the occurrence of a word is conditionally independent of all other words if the category is known. The resulting algorithm is very efficient. It has been extended to a mixture of multinomials and successfully applied to text categorization [25].

Multi-layer perceptrons were used by Tweedie et al. [31] to attribute authorship to the disputed Federalist papers. They used the normalized frequency of eleven common function words in a text as input to the network. The neural network had three hidden and two output nodes. It was trained with conjugate gradient descent and tested by use of k-fold cross-validation. The network unambiguously classified the disputed Federalist papers as being by Madison, which is consistent with the results of other authors using other methods. A problem with neural networks for text classification is the high computational effort for training.

Nevertheless perceptrons and other semi-parametric models pose a problem to the user: they offer little or no insight into the process by which they arrived at a given result nor, in general, the totality of "knowledge" actually embedded in them. A number of techniques to explain these networks have been developed [32] recently.

Radial basis function (RBF) networks start with a number of prototype feature vectors for each class and assume that the feature vector of a new exemplar is 'close' to some prototype of its class. The distance to the prototype is measured by the common Euclidean distance or some generalized version weighting specific features. RBF-networks model an author directly as a mixture of different styles, which may depend on topic and genre. Therefore they are especially suited to stylometric analysis. Also, prior knowledge can be used to initialize weight vectors. This is important for relatively small data sets, a situation that can easily arise in authorship attribution.

RBF-networks were used by [28] for stylometric analysis. They use the frequency of five function words as features, normalized to zero mean and unit variance. They were used to discriminate plays of Shakespeare and Fletcher with a total of 50 samples for each author. The trained RBF-networks produced classifications in agreement with conventional scholarship and the application of computational methods such as multi-layer perceptrons.

k-nearest neighbor classification (kNN) is similar to RBF networks as it uses the distance to prototypes as a criterion. Instead of constructing synthetic prototypes, instances from the training set are employed. For a new test document the algorithm finds the k nearest neighbors among the training documents. The resulting classification is a weighted majority vote of the categories of these neighbors. The details of the voting mechanism depends on the specific procedure [30]. It is one of the top performing methods on the benchmark Reuters corpus.

Decision trees sequentially partition the input space along a single dimension. The corresponding variable is selected in a one-step lookahead greedy search using some heuristic measure of classification quality [33]. This splits the training set into two parts and the procedure starts over. The size of the resulting tree is limited by cross-validation. Recently, a boosting version was applied to text categorization with good results [29].

3.3. Support Vector Machines

Support Vector Machines (SVMs) recently gained popularity in the learning community [34]. In its simplest linear form, an SVM is a hyperplane that separates a set of positive examples from a set of negative examples with maximum interclass distance, the *margin*. Figure 1 shows such a hyperplane with the associated margin.

The formula for the output of a linear SVM is

$$u = w * x + b \tag{4}$$

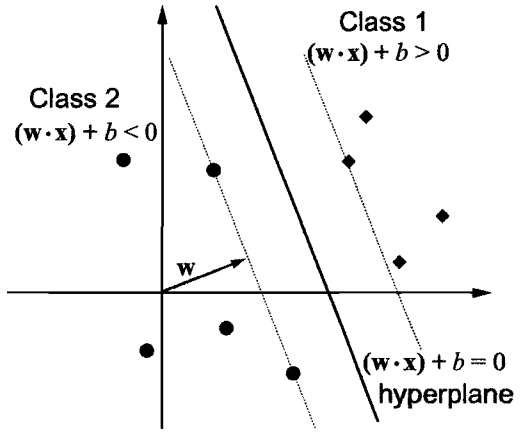

Figure 1. Hyperplane with maximal margin generated by a linear SVM.

where w is the normal vector to the hyperplane, and x is the input vector. The margin is defined by the distance of the hyperplane to the nearest of the positive and negative examples. Maximizing the margin can be expressed as an optimization problem:

$$\text{minimize} \frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w * x_i + b) \geq 1, \forall_i \tag{5}$$

where x_i is the i-th training example and $y_i \in \{-1, 1\}$ is the correct output of the SVM for the i-th training example. Note that the hyperplane is only determined by the training instances x_i on the margin, the *support vectors*. Of course, not all problems are linearly separable. Cortes and Vapnik [35] proposed a modification to the optimization formulation that allows, but penalizes, examples that fall on the wrong side of the decision boundary.

Support vector machines are based on the structural risk minimization principle [34] from computational learning theory. The idea is to find a model for which we can *guarantee* the lowest true error. This limits the probability that the model will make an error on an unseen and randomly selected test example. An SVM finds a model which minimizes (approximately) a bound on the true error by controlling the model complexity (VC-Dimension). This avoids over-fitting, which is the main problem for other semi-parametric models.

The SVM can be extended to nonlinear models by mapping the input space into a very high-dimensional feature space chosen a priori. In this space the optimal separating hyperplane is constructed [36], [34, p. 421]. Let $\Phi : \Re^N \rightarrow F$ be a mapping $x \rightarrow z = \Phi(x)$ such that $N \ll \dim(F)$. Then for a given Φ the linear classification $u = s * z + b$ with parameter s may be learned. Note, however, that the algorithm only uses the dot products. Therefore the high-dimensional values z and s have not to be calculated, only the dot products $k(x, w) := \Phi(x) * \Phi(w) = s * z$ are required, which can be determined in the input space. Examples of common kernels are

Polynomial	$k(x, y) = (xy' + c)^d$
Sigmoid	$k(x, y) = \tanh(\kappa(xy') + \Theta)$
RBF	$k(x, y) = \exp(-\gamma\|x - y\|^2)$

A new vector x is classified into class 1 if the following decision function has a value >0

$$h(x) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i k(x_i, x) + b\right) \tag{6}$$

Only the α_i corresponding to the support vectors x_i in the training set, i.e. the examples on the margin, are different from 0. If we rewrite $h(x) = \text{sgn}(w * x)$ and select some $\gamma > 0, \delta \in (0, 1)$, then for all distributions generating the data the following inequality holds with probability $\geq 1 - \delta$ over the ℓ training patterns:

$$\text{Test Error} \leq \nu + \sqrt{\frac{c}{\ell}\left(\frac{R^2\Lambda^2}{\gamma^2}\log^2 \ell + \log\frac{1}{\delta}\right)} \tag{7}$$

where $\|x\| \leq R$, $\|w\| \leq \Lambda$, the term ν is the fraction of training samples with margin $y_i(wx_i) < \gamma$ and c is a large constant. Therefore the inputs x should all have a comparable length $\|x\|$.

Training an SVM requires the solution of a quadratic programming (QP) problem. Any QP optimization method can be used to learn the α_i and b on the basis of training examples. However, many QP methods can be very slow for large problems. We used a method implemented by Joachims [1] which is especially suitable for text classification as it uses a sparse representation of inputs. Once the weights are learned, new items x are classified by computing $h(x)$ as defined in (6).

The distinctive advantage of the SVM for text categorization is its ability to process many thousand different inputs. This opens the opportunity to use all *words* in a text directly as features. For each word w_i the number of times of occurrence is recorded. Typically a corpus contains more than 100,000 different words, with each text covering only a small fraction.

Joachims [18] used the SVM for the classification of text into different topic categories. As features he uses word stems. To establish statistically significant features he requires that each feature occurs at least three times in a text. The empirical evaluation was done on two test collections: the Reuter-21578 news agency data set covering different topics and the Ohsumed corpus of William Hersh describing diseases. Using about 10000 features in every case, the two SVM versions (polynomial and rbf) performed substantially better than the currently best performing conventional methods (naive Bayes, Rocchio, decision trees, k-nearest neighbor). Joachims [37] used a transductive SVM for text categorization which is able to exploit the information in unlabeled training data.

Dumais et al. [2] use linear SVMs for text categorization because they are both accurate and fast. They are 35 times faster to train than the next most accurate (a decision tree) of the tested classifiers. They applied SVMs to the Reuter-21578 collection, emails and web pages.

Drucker et al. [38] classify emails as spam and non spam. They find that boosting trees and SVMs have similar performance in terms of accuracy and speed. SVMs train significantly faster.

4. Transformations of Frequency Vectors

4.1. Normalization of Length

It is a well known fact that the frequency distribution of words in large texts is fairly skewed. Zipf's law in the original version [39] was given by

$$f(r) = \frac{A}{B + r}, \tag{8}$$

where $f(r)$ is the frequency of the term of rank r in a text and A and B are positive parameters. It can be seen that the distribution of frequencies of terms in texts is extremely uneven. Some units occur very often, whereas as a rule of thumb half of the terms—the so called *hapax* legomena—occur only once. Unfortunately especially the rare units contain highly specific information about the content of the text. Formula (8) has been generalized by various authors (for a summary see e.g. [40]). Zipf himself explained Eq. (8) by a "principle of least effort". A generalization of Zipf's law is given by

$$f(r) = \left(\frac{A}{B + r}\right)^{\frac{1}{\gamma - 1}}, \tag{9}$$

which is known as Zipf-Mandelbrot law [41]. It contains Eq. (8) as a special case. In order to compare documents of different length term-frequency vectors d_i have to be normalized to a standard length. From the standpoint of performance of SVM the best normalization rule is (7)

$$d_i^* = \frac{d_i}{\|d_i\|_{L_2}}, \tag{10}$$

where $\|x\|_{L_p} = (\sum_i |x_i|^p)^{1/p}$ is the p-norm and $\|\cdot\|_{L_2}$ denotes the Euclidean norm. This is the commonly used transformation of term frequencies when SVM are applied to text classification. Let us examine the effect of the spontaneous emergence of new lexical items. Herdan [42, p. 25] stated that the number of types $|V|$ in a text grows with text length $|N|$ according to the formula

$$V = N^c, \qquad 0 < c < 1$$

From the Zipf-Mandelbrot law one can deduce, that when enlarging a text from N_1 to N_2 frequencies of lexical units do not increases by the same factor. They increase by $a \cdot \frac{N_2}{N_1}$, where a is a positive constant smaller than 1. The quantity a can be calculated as follows:

Let $f(r)$ be the rank-frequency distribution of a text N_1 with length $|N_1|$ (measured in running words) and vocabulary size $|V_1|$. Let N_2 be a text which contains N_1 thus $N_1 \subset N_2$ and the same holds for the corresponding vocabularies: $V_1 \subset V_2$. Let $a \cdot f(r)$ be the rank-frequency distribution of N_2, which means that both texts have the same structure. Under these conditions

$$|N_1| = \sum_{r=1}^{|V_1|} f(r)$$

and

$$\begin{aligned} |N_2| = a \sum_{r=1}^{|V_2|} f(r) &= a \sum_{r=1}^{|V_1|} f(r) + a \sum_{r=|V_1|+1}^{|V_2|} f(r) \\ &= a \cdot |N_1| + a \sum_{r=|V_1|+1}^{|V_2|} f(r). \end{aligned}$$

Hence the effect of increasing vocabulary on the p-norm of the term frequency vector can be calculated as follows:

$$\|N_2\|_{L_p} = \sum_{r=1}^{|V_1|} f^p(r) + \sum_{r=|V_1|+1}^{|V_2|} f^p(r). \tag{11}$$

inserting Zipf-Mandelbrot law from Eq. (9) yields

$$a = \frac{\|N_2\|_{L_p}}{\|N_1\|_{L_p} + \sum_{r=|V_1|+1}^{|V_2|} \left(\frac{A}{B+r}\right)^{p/\gamma}}$$

If $p > \gamma$ the sum in the denominator converges as vocabulary size tends to infinity. It diverges if $p \leq \gamma$. As in our material the term-frequency spectrum exhibits $\gamma < 1$, both L_2-norm and L_1-norm are equally justified. The empirical tests show that normalization with respect to L_2 yields the better results compared with L_1 the larger γ.

Using a particular metric is equivalent to incorporating prior knowledge into the solution of the problem. With increasing p more weight is given to the larger values in feature space. This suggests L_1 in favor of L_2 in text classification tasks. On the other hand it is known that SVM work best when input vectors are normalized to unit length with respect to the Euclidean norm. Therefore we considered the following normalization rules.

- normalization to unit length with respect to L_1.
- normalization to unit length with respect to L_2.

4.2. *Transformation of Frequencies*

As lexical units scale to different orders of magnitude in larger documents, it is interesting to examine how term frequency information can be mapped to quantities which can efficiently be processed by SVM. For our empirical tests we use different transformations of type frequencies: raw frequencies and logarithmic frequencies.

As the simplest "frequency-transformation" we use the term-frequencies $f(w_k, d_i)$ of type w_k in document d_i. The frequencies are multiplied by one of the importance weights described below and normalized to unit length. Raw frequencies $f(w_k, d_i)$ with importance weight *idf* (see Section 4.3) normalized with respect to L_2-norm have been used by [1] and others. We also tested other combinations, for example raw frequencies with no importance weight normalized with respect to L_1-norm, which is defined by

$$F_{rel_1}(w_k, d_i) = \frac{f(w_k, d_i)}{f(d_i)} \tag{12}$$

where $f(d_i)$ is the number of types in document d_i.

The second transformation we use is the logarithm. We consider this transform because it is a common approach in quantitative linguistics to consider logarithms of linguistic quantities rather than the quantities themselves. Here we also normalize to unit length with respect to L_1 and L_2. We define

$$F_{log}(w_k, d_i) = \log(1 + f(w_k, d_i)/f(d_i)), \tag{13}$$

combine F_{log} with different importance weights, and normalize the resulting vector with respect to L_1 and L_2.

4.3. *Importance Weights*

Importance weights are often used in order to reduce the dimensionality of a learning task. Feature extraction in text retrieval is often thought of in terms of reduction of dimensionality of the input space. Common importance weights like inverse document frequency (idf) originally have been designed for identifying index words [43].

However, SVM are capable to manage a large number of dimensions. Therefore reduction of dimensionality is not necessary and importance weights can be used to quantify how important a specific given type is in the documents of a text collection. A type which is evenly distributed across the document collection will be given a low importance weight because it is judged to be less specific for the documents it occurs in, than a type which is used in only a few documents. The importance weight of a type is multiplied by its transformed frequency of occurrence. So each of the different importance weights can be combined with each of the frequency transformations described above. We examined the performance of the following combinations:

- no importance weight
- inverse document frequency (idf)
- redundancy

Inverse document frequency (idf) is commonly used when SVM is applied to text classification. Idf is defined by

$$F_{idf}(w_k) = \log \frac{n_d}{n_d(w_k)}, \tag{14}$$

where $n_d(w_k)$ is the number of those documents in the collection, which contain the term w_k and n_d is the number of all documents in the collection. Intuitively, the inverse document frequency of a word is low if it

occurs in many documents and is highest if the word occurs only one document. $F_{idf}(w_k)$ forms a fixed vector of importance weights comprising all types occurring in the training set. The idf-values of those types in a new document d^* which do not occur in the training data is set to 0. This scheme makes it possible to process documents from an open source, alleviating the restrictions imposed by the fixed vocabulary of the training data.

The idf has the advantage of being easy to calculate. Its disadvantage is that it disregards the frequencies of the term w_k in the documents. A term which occurs once in all documents except one, but occurs a hundred times in the one remaining document, should be judged as important for the classification of documents, because it is highly specific to one document. In contrast a term which occurs 20 times in all documents with no exception is not specific for any document and is not useful for training and testing a classification procedure. This effects mainly high frequency terms and collections of larger documents. Empirical studies show that the utilization of $F_{idf}(w_k)$ in many cases lead to an improved performance [44].

Redundancy quantifies the skewness of a probability distribution. We consider the empirical distribution of each type over the different documents and define an importance weight by

$$R(w_k) = M_{max} - H = \log n + \sum_{i=1}^{n_d} \frac{f(w_k, d_i)}{f(w_k)} \log \frac{f(w_k, d_i)}{f(w_k)}, \quad (15)$$

where $f(w_k, d_i)$ is the frequency of occurrence of term w_k in document d_i, $f(w_k)$ is the frequency of occurrence of term w_k in the training set and n_d is the number of documents in the training set. $R(w_k)$ forms a fixed vector of importance weights comprising all types occurring in the training set. Again, we define the value of types which do not occur in the training data to be 0. $R(w_k)$ is a measure of how much the distribution of a term w_k in the various documents deviates from the uniform distribution.

5. Numerical Experiments

We consider the task to decide if a text previously unknown belongs to a given author or not. A plausible scenario is that a registered external student delivers a written exam and the teacher wants to know if the text actually was written by the student. The specific setup is reflected in the loss function $L(y, d)$, which specifies the loss that incurs, if decision d is selected and y is the true class. The loss for the (mis)classification of an extraneous author as target author was set to 5, as it is more harmful not to identify the target author. In the latter case we may recognize the correct author by some additional investigation, while in the first case the faked exam passes unnoticed.

5.1. The Data

For our experiments we used texts from the Berliner Zeitung (BZ), a daily newspaper in Berlin. Every issue since 1994 can be downloaded from `http://www.BerlinOnline.de/wissen-/berliner_zeitung/`. We used the material from Dec 1998 to Feb 1999. The articles were subdivided into twelve topics. We studied three of them: politics, economy, and local affairs. Table 1 shows the number of words in the sub-categories. The percentage of hapaxes (i.e. words occurring only once) is 52% of the types and thus rather large.

To reduce the computational effort we only utilized texts with more than 200 words for the target authors and 300 words for the other authors. This training corpus consists of 2652 documents, about 1.9 Mio. running words (tokens) and about 120,000 different words (types), including all word-forms.

Figure 2 shows the number of documents written by the authors on a logarithmic scale. There are more than 150 authors who have contributed only a single document. As only little information is available about these alternative authors the discrimination task is rather difficult. Figure 9 shows the rank order of different word types.

To exploit the training set S_0 of 2652 documents in a better way, we used the following *cross-testing* procedure. S_0 was randomly divided into 5 subsets $S_1, \ldots, S_5$ of nearly equal size. Then five different SVMs were

Table 1. Summary statistics for the Berliner Zeitung corpus for documents with length ≥200.

	Number of ...			Percent	Average
Topic	Tokens	Types	Texts	hapaxes	text length
Politics	525,000	50,400	1,200	56.3	438
Economy	264,000	30,900	550	57.6	480
Local affairs	1,155,000	78,700	3,233	49.6	357

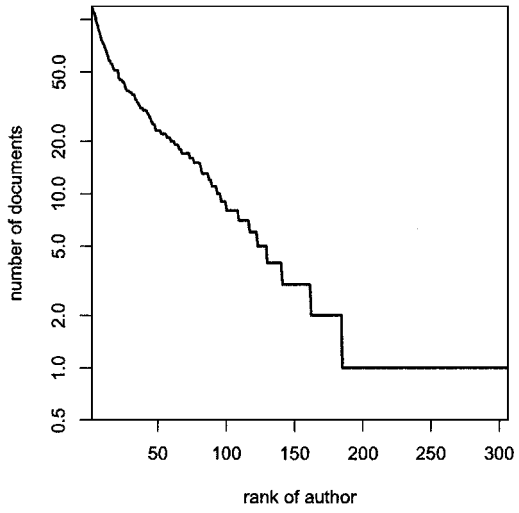

Figure 2. Rank order of number of documents written by the authors.

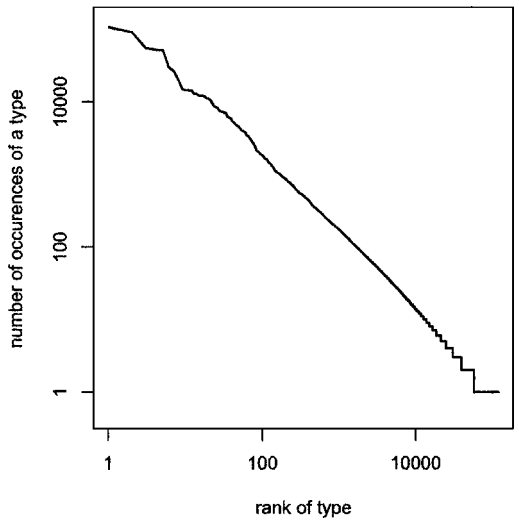

Figure 3. Rank order of frequency of different types (words) in the corpus.

generated using $S_0 \backslash S_i$ as training set of size ≈2121 and S_i was used as test set of size ≈531. The numbers of correctly and wrongly classified documents were added up yielding an effective test set of all 2652 documents.

As discussed above we assume that the cost of assigning an extraneous text to the target author is five times worse than that of not identifying a genuine text. Let $c_{\text{tar}}(i)$ and $e_{\text{tar}}(i)$ respectively denote the number of correctly and incorrectly classified documents of the target author i and let $e_{\text{oth}}(i)$ and let $c_{\text{oth}}(i)$ be the same figures for the other authors in the alternative class. In the corpus about 3% of the texts originate from the target author. We assume that new texts to be classified arrive at the same ratio, i.e. the prior probability is $p_{tar} = 0.03$. Hence the loss for a the classification of a test set is

$$L(i) = p_{tar} \frac{e_{\text{tar}}(i)}{e_{\text{tar}}(i) + c_{\text{tar}}(i)} + 5(1 - p_{tar}) \frac{e_{\text{oth}}(i)}{e_{\text{oth}}(i) + c_{\text{oth}}(i)} \tag{16}$$

To allow comparisons we always report the loss for 1000 new classifications.

In addition we used the precision $p_{\text{prc}}(i) = c_{\text{tar}}(i)/[c_{\text{tar}}(i) + e_{\text{oth}}(i)]$, $p_{\text{prc}}(i) = 0$ if $c_{\text{tar}}(i) = 0$, and the recall $p_{\text{rec}}(i) = c_{\text{tar}}(i)/[c_{\text{tar}}(i) + e_{\text{tar}}(i)]$ to describe the result of an experiment. Precision is the probability that a document predicted to be genuine truly belongs to this class. Recall is the probability that a genuine document is classified into this class. A tradeoff exists between large recall and large precision. By adjusting the parameter b in (6) the recall may be increased at the cost of decreasing precision and vice versa. The derived standard measure of retrieval performance is $F_1(i) = 2p_{\text{prc}}(i)p_{\text{rec}}(i)/[p_{\text{prc}}(i) + p_{\text{rec}}(i)]$ proposed by [45]. As precision and recall do not reflect the actual loss function we did not use it as main criterion in this paper.

For transformation we used the relative frequencies (12) 'rel', the logarithm of relative frequencies (13) 'logRel'. 'rel' was also weighted by the inverse document frequencies (14) and hence called 'tfidf' in the following sections. The transformations rel and logRel were alternatively weighted by the redundancy weight (15).

5.2. *Experiments with Word Forms*

In the first experiment we selected the seven authors with most documents in the combined areas of politics and local affairs from December 1998 to February 1999. The number of texts for these authors are in the range from 82 to 118. For a first set of runs we used the counts of lower case word forms in each document as inputs to the SVM.

For all our experiments we used the program SVM-light written by Thorsten Joachims [1], which is available at `http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html`. We investigated the following

Table 2. Best result for each author in terms of loss for SVM with word forms or the combined tagwords, bigrams of tagwords and word lengths with cross testing.

			Other authors		Target author			Percent	
Author	Transformation	Kernel	# ok	# false	# ok	# false	Loss	Prec.	Recall
			Word Forms						
Aulich	logRel L_1	linear, quadr., cubic, rbf	2652	0	94	14	3.9	100.0	87.0
Aulich	logRel L_2	linear	2652	0	94	14	3.9	100.0	87.0
Fuchs	logRel L_2	linear	2642	0	98	20	5.1	100.0	83.1
Kunert	logRel L_1 with redundancy	rbf $\gamma = 0.5$	2659	1	71	29	10.5	98.6	71.0
Muenner	rel L_2	linear	2673	0	80	7	2.4	100.0	92.0
Neumann	rel L_1	rbf $\gamma = 2$	2647	2	73	38	13.9	97.3	65.8
Neumann	rel L_2	quadr.	2647	2	73	38	13.9	97.3	65.8
Schmidl	logRel L_1	linear, quadr., cubic, rbf	2666	0	66	28	8.9	100.0	70.2
Schmidl	logRel L_2	linear	2666	0	66	28	8.9	100.0	70.2
Schomaker	logRel L_1	cubic, rbf	2678	0	25	57	20.9	100.0	30.5
			Tagwords + Bigrams of Tagwords + Word Lengths						
Aulich	logRel	linear	2652	0	85	23	5.9	100.0	79.0
Fuchs	logRel	linear	2642	0	89	29	7.2	100.0	75.0
Kunert	logRel	linear	2660	0	61	39	11.8	100.0	61.0
Muenner	logRel	linear	2673	0	67	20	6.7	100.0	77.0
Neumann	logRel	linear	2649	0	51	60	16.0	100.0	46.0
Schmidl	rel	rbf $\gamma = 2$	2666	0	60	34	10.8	100.0	63.8
Schomaker	rel	quadr.	2677	1	17	65	25.6	94.4	21.0

kernels: linear, quadratic, cubic, and rbf with different widths γ.

For each training set we computed SVM models for a number of combinations of kernels and transformations. We used the transformations relative frequency (rel), relative frequency with redundancy factor (rel w. redund.), logarithmic relative frequency (logRel), logarithmic relative frequency with redundancy factor (logRel w. redund.), and inverse document frequency (tfidf) in conjunction with linear, quadratic, cubic and rbf-kernel with $\gamma = 0.5, 1, 2$. Furthermore the resulting frequency vectors were normalized using either the L_1 or the L_2 norm. Therefore we end up with 60 combinations for each author. SVM models were estimated with these combinations for each of the five training sets.

Table 2 shows details for the best result of each author in terms of loss. Precision and recall values are also displayed. There are multiple entries for one author in cases, where several combinations of frequency transformations and SVM kernel functions showed identical performance with respect to loss. It is impressive that nearly none of the more than 2600 documents of other authors are attributed to the target author. This yields a precision of about 100%. On the other hand in a number of cases the target author is not classified correctly. This recall ranges from 28% to 92%. However, for all but one author the result is better than 66%. If a document is assigned to the target author we are pretty sure that the allocation is correct, whereas in the opposite case we have to check the result by other means. This situation corresponds to the intended application.

Table 3 shows the averaged loss over all authors for the experiment with full word forms. It turns out that for most frequency transformations the choice of the SVM kernel function has little or no effect on the performance in terms of loss. Apart from this observation, logarithmic relative frequencies with L_1 normalization have the overall best performance. The rbf kernel seems to be sensitive to the choice of the γ parameter in some cases. It is therefore considered to be less adequate.

Table 3. Loss of full word-forms experiment for different transformations and SVM kernels averaged over the authors.

	Full forms with L_1 transformation ...					Full forms with L_2 transformation ...				
Kernel	logRel	logRel w. redund.	rel	rel w. redund.	tfidf	logRel	logRel w. redund.	rel	rel w. redund.	tfidf
linear	10.6	12.1	12.2	14.5	11.2	10.9	13.0	11.1	14.4	12.0
quadr.	10.6	12.1	12.0	14.5	11.2	11.5	13.2	11.2	13.1	12.0
cubic	10.4	12.1	12.0	14.5	11.2	12.3	13.3	11.4	13.7	12.5
rbf $\gamma = 0.5$	10.4	12.0	12.2	14.5	11.2	12.1	13.2	11.3	13.1	12.3
rbf $\gamma = 1$	10.4	12.0	11.7	14.4	11.2	12.9	13.9	11.4	13.4	13.1
rbf $\gamma = 2$	10.5	12.0	11.4	14.4	11.2	14.6	16.8	12.4	16.5	14.0

5.3. Bigrams of Tags and Function Words

The second set of inputs was aimed at using less content information and more structural data. Here we lemmatized the corpus using the lemmatizer Morphy. Morphy was designed by Lezius et al. [46] and is freely available on the Web. Lemmatization produced 455 different categories. An example is shown in Table 4.

We excluded all nouns (SUB), verbs (VER), and adjectives (ADJ) and used the word categories instead. All other types were taken as function words with little content. Their tags comprised auxiliary verbs, articles, conjunctions, punctuation marks, adverbs, and prepositions. Ignoring function words appearing less than ten times, the corpus contained a total of 817 function word types, comprising 55% of the tokens. The remaining 97600 types were non-function words. Finally we combined the function words with tags yielding a total of 2844 types, as some function words can play several syntactic roles. The result is shown in the last column of Table 4. Note that taggers usually commit about 5% errors. These errors can be considered as part of the preprocessing process. In the remainder of the section we call the combined tags and function words 'tagwords'.

Thus the input vector of the SVM comprised the following sub-vectors:

- The frequency of words of different lengths. Words longer than 25 letters were combined in a category.
- The frequency of tagwords.
- The frequency of bigrams of tagwords. Here the document borders were marked by a special delimiter.

Table 4. Example of tagged word forms and tagged function words.

Lemmas	Word-forms	Grammatical tags	Tagwords: Tags and function words
der	Der	ART_DEF_NOM_SIN_MAS	der_ART_DEF_NOM_SIN_MAS
Algengürtel	Algengürtel	SUB_NOM_SIN_MAS	SUB_NOM_SIN_MAS
vor	vor	PRP_DAT	vor_PRP_DAT
der	der	ART_DEF_DAT_SIN_FEM	der_ART_DEF_DAT_SIN_FEM
norwegisch	norwegischen	ADJ_DEF_DAT_SIN_FEM	ADJ_DEF_DAT_SIN_FEM
Küste	Küste	SUB_DAT_SIN_FEM	SUB_DAT_SIN_FEM
haben	hat	VER_3_SIN	haben_VER_3_SIN
sich	sich	REF_AKK_SIN_3	sich_REF_AKK_SIN_3
gestern	gestern	ADV	gestern_ADV
erneut	erneut	ADV	erneut_ADV
um	um	PRP_AKK	um_PRP_AKK
25	25	ZAN	25_ZAN
Kilometer	Kilometer	SUB_AKK_PLU_MAS	SUB_AKK_PLU_MAS
vergröß ern	vergröß ert	VER_PA2	VER_PA2

Table 5. Loss of tagforms experiment for different transformations and SVM kernels averaged over the authors.

	Bigrams of tagwords with transformation ...		
Kernel	logRel L_2	rel L_2	tfidf L_2
linear	12.4	17.0	13.4
quadratic	12.6	15.4	13.6
cubic	13.2	15.0	13.6
rbf $\gamma = 0.5$	13.0	15.0	13.6
rbf $\gamma = 1$	14.2	14.8	13.9
rbf $\gamma = 2$	15.7	14.2	14.4

From the maximal number of 667 thousand possible bigrams 70315 occurred in the corpus. To this comprehensive vector of counts the transformations were applied as discussed above. Obviously the resulting input vector is very sparse.

Again the result of the cross-testing is shown in Table 2. This time we restricted the range of frequency transformartions to L_2-normalized ones. With respect to precision the results even outperform the full word forms. In only one case an extraneous text is wrongly assigned to a target author. This performance is nearly optimal. With respect to recall however, bigrams are less reliable than the univariate word forms. On the average the recall percentage is about 10 points worse than that of the word forms. Nevertheless bigrams of tagwords seem to yield high information content which can be exploited by the SVM for authorship attribution.

Table 5 shows the average loss for each combination of transformation and kernel over all authors. Again in most cases the linear kernel with logarithmic relative frequencies has minimal loss. The quadratic kernel performs nearly as well, while the cubic and rbf kernel seem to be less adequate. Simple relative frequencies do much worse than the logarithmic version.

In summary bigrams of tagwords work surprisingly well, taking into account that most content information has been eliminated.

5.4. *Testing on a Different Topic*

The function words were selected to minimize content related information. This election should perform well if an author changes topic. We tested this by training the SVM with data from the local affairs topic and testing it with data from economy and politics. We selected five target authors which simultaneously contributed to both areas and had 70 or more texts in the training set. Naturally the number of test cases with 7 to 21 was far less than in the previous experiment. Only L_2-normalized frequency transformations were tested.

The results are shown in Table 6. Again for all configurations and authors the number of misclassifications of extraneous texts is zero. Hence precision is 100% in most cases. With respect to recall word forms perform best. They have values in the range of 55–100% in contrast to 15–40% for bigrams. We also experimented with trigrams which performed worse than bigrams. One explanation for the high errors may be that most bigrams are hapaxes and occur only once in the text. Therefore an intelligent aggregation procedure is required which retains discriminating information but increases statistical significance. Note however that these results only give a first impression as the number of test cases is very low.

5.5. *Comparison with Other Classifiers*

Finally we performed some preliminary experiments with alternative classification procedures, decision trees and multilayer perceptrons. As in Section 5.2 we trained them on the texts from Dec 1998 to February 1999. Here we used only one author, Fuchs, and a single training set $S\backslash S_i$ of size 2490 with 83 positive examples and a test set S_i of size 1244 with 35 positive examples. It was impossible to use the frequencies of all word forms as inputs. Therefore we aggregated the tagwords so that each category contained about 200 tokens in the training set. This produced about 400 types which were used as inputs.

We tested all of the frequency transformations used earlier, not including redundancy factors. The best results are shown in Table 7. As the result of MLP depends on the starting values of its weights we show the average of 10 runs. While decision trees perform much worse, MLPs are comparable to the SVM.

The results must be compared to those of SVMs using the full number of input features as shown in Table 2. For author Fuchs, we find in the tagwords section of this table a loss value of 7.2, precision of 100%, and recall of 75%. Therefore these preliminary experiments indicate that SVMs are superior for author

Table 6. Best result for each author in terms of loss for SVM with word forms, and bigrams or trigrams of tagwords trained on local affairs and tested on politics and economy.

			Other authors		Target author			Percent	
Author	Transf.	Kernel	# ok	# false	# ok	# false	Loss	Prec.	Recall
Word Forms									
Emmerich	logRel	linear	1253	0	5	2	8.6	100	71.4
Geschonneck	rel	linear	1252	0	8	0	0.0	100	100.0
Neumann	logRel	cubic	1251	0	5	4	13.3	100	55.6
Richter	tfidf	cubic	1239	0	12	9	12.9	100	57.1
Schomaker	rel	cubic	1253	0	5	2	8.6	100	71.4
Word Length + Tagwords + Bigrams of Tagwords									
Emmerich	tfidf	cubic	1253	0	1	6	25.7	100	14.3
Geschonneck	tfidf	linear rho = 1	1252	0	3	5	18.8	100	37.5
Neumann	tfidf	linear rho = 1	1251	0	3	6	20.0	100	33.3
Richter	logRel	quadr.	1239	0	4	17	24.3	100	19.0
Schomaker	tfidf	cubic	1253	0	2	5	21.4	100	28.6

Table 7. Results for multilayer perceptron, decision tree and SVM on tagged and aggregated function words.

		Other authors		Target author			Percent	
Method	Transformation	# ok	# false	# ok	# false	Loss	Precision	Recall
MLP	logRel L_2	1208	1	18	17	19.8	93.3	51.4
Tree	logRel L_1	1192	17	5	30	93.1	22.7	14.3
SVM	logRel L_2	1209	0	18	17	14.6	100.0	51.4

identification, if we are willing to code a very large number of features.

6. Conclusions

This paper discusses some preliminary work on author identification and introduces support vector machines as a new approach. SVMs especially suited for this task as the feature spaces have very high dimensions, most features carry important information and the data for specific instances is sparse. The experimental results show that SVMs consistently achieve good performance for the identification tasks. There is no need to select specific features, we may simply take all word frequencies. In addition the preprocessing and weighting of features is not critical, many approaches lead to nearly identical results.

SVMs for authorship attribution and text mining can process documents of significant length and databases with a large number of texts. SVM technology is firmly grounded in computational learning theory and training times compare favorably with other methods such as neural networks. Therefore SVMs are currently the method of choice for authorship attribution.

A second problem investigated in this paper is the question whether function words are sufficient for author identification. It turned out that bigrams of function words perform less well than word forms but still show a performance superior to other classifiers which can only handle a reduced set of features. If the procedure is tested on texts by the same authors on another topic the performance is significantly reduced, with the full forms performing still better than the function word bigrams. It seems that function word bigrams—at least of the type investigated in this paper—carry significantly less information on an author than the full function words. It remains to be explored in future research if bigrams of tags or other structural features can be used as 'content free' characteristics for author identification.

References

1. T. Joachims, "Making large-scale SVM learning practical," Technical Report, Uni Dortmund, 1998.
2. S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *7th International Conference on Information and knowledge Management*, 1998.
3. T.C. Mendenhall, "The characteristic curves of composition," *Science*, vol. IX, pp. 237–249, 1887.
4. G.U. Yule, "On sentence length as a statistical characteristic of style in prose with application two cases of disputed authorship," *Biometrika*, vol. 30, pp. 363–390, 1938.
5. J. Gani, "Literature and statistics," in *Encyclopedia of Statistical Sciences*, edited by S. Katz and N.-L. Johnson, Wiley, 1985, vol. 5, pp. 90–95.
6. D.I. Holmes, "The evolution of stylometry in humanities scholarship," *Literary and Linguistic Computing*, vol. 13, no. 3, pp. 111–117, 1998.
7. George K. Zipf, *Human Behaviour and the Principle of Least Effort. An Introduction to Human Ecology*, Houghton-Mifflin: Boston, 1932.
8. George K. Zipf, "Observations on the possible effects of mental age upon the frequency-distribution of words from the viewpoint of dynamic philology," *J. of Psychology*, vol. 4, pp. 239–244, 1937.
9. H.S. Sichel, "On a distribution law for word frequencies," *Journal of the Americal Statistical Association*, vol. 70, pp. 542–547, 1975.
10. J.K. Orlov, "Ein Modell der Häufigkeitsstruktur des Vokabulars," in *Studies in Zipf's Law*, edited by H. Guiter and M. Arapov, Brockmeyer, Bochum, 1983, pp. 154–233.
11. K. Tzeras and S. Hartmann, "Automatic indexing based on Bayesian inference networks," in *16th Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '93)*, 1993, pp. 22–34.
12. F. Mosteller and D.L. Wallace, *Inference and Disputed Authorship: The Federalist*, Addison-Wesley: Reading, MA, 1964.
13. J.F. Burrows, "Word patterns and story shapes: The statistical analysis o narrative style," *Literary and Linguistic Computing*, vol. 2, pp. 61–70, 1987.
14. J. Binongo and M. Smith, "A bridge between statistics and literature: The graphs of Oscar Wilde's literary genres," *J. Applied Statistics*, vol. 26, pp. 781–787, 1999.
15. R.H. Baayen, H. van Halteren, and F.J. Tweedie, "Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution," *Literary and Linguistic Computing*, vol. 11, no. 3, pp. 121–131, 1996.
16. E. Charniak, *Statistical Language Learning*, MIT Press: Cambridge, MA, 1993.
17. J. Rudman, "The state of authorship attribution studies: Some problems and solutions," *Computers and the Humanities*, vol. 31, pp. 351–365, 1998.
18. T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European Conference on Machine Learning (ECML)*, edited by C. Nedellec and C. Rouveirol, 1998.
19. Y.M.M. Bishop, S.E. Fienberg, and P.W. Holland, *Discrete Multivariate Analysis: Theroy and Practice*, MIT-Press, Cambridge, MA, 1975.
20. A.L. Goel, "Cumulative sum control charts," in *Encyclopedia of Statistics*, edited by S.Kotz and N. Johnson, Wiley, 1982, vol. 2, pp. 233–241.
21. Jill M. Farringdon, *Analysing for Authorship: A Guide to the Cusum Technique*, University of Wales Press: Cardiff, 1996.
22. B. Thisted and R. Efron, "Did Shakespeare write a newly discovered poem?" *Biometrika*, vol. 74, pp. 445–455, 1987.
23. R.J. Valenza, "Are the Thisted-Efron authorship tests valid?" *Computers and the Humanities*, vol. 25, pp. 27–46, 1991.
24. Y. Yang and C. Chute, "An example-based mapping method for text categorization and retrieval," *ACM Transaction on Information Systems*, vol. 12, pp. 252–277, 1994.
25. A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, 1998.
26. I. Moulinier, G. Raskinis, and J. Ganascia, "Text categorization: A symbolic approach," in *Proc. of the Fifth Symp. on Document Analysis and Information Retrieval*, 1996.
27. H. Ng, W. Gob, and K. Low, "Feature selection, perceptron learning and a usability case study for text categorization," in *20th Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '97)*, 1997, pp. 67–73.
28. D. Lowe and R. Matthews, "Shakespeare vs. Fletcher: A stylometric analysis by radial basis functions," *Computers and the Humanities*, vol. 29, pp. 449–461, 1995.
29. C. Apte, F. Damereau, and S. Weiss, "Text mining with decision rules and decision trees," in *Proc. Conf. on Automated Learning and Discovery, Workshop 6: Learning from Text and the Web*, 1998.
30. K. Lam and C. Ho, "Using a generalized instance set for automatic text categorization," in *21th Ann. Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '98)*, 1998, pp. 81–89.
31. F.J. Tweedie, S. Singh, and D.I. Holmes, "Neural network applications in stylometry: The Federalist papers," *Computers and the Humanities*, vol. 30, pp. 1–10, 1996.
32. R. Andrews, J. Diederich, and A.B. Tickle, "A survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Systems*, vol. 8, pp. 373–389, 1995.
33. J.R. Quinlan, "Inferno: A cautious approach to uncertain inference," *The Computer Journal*, pp. 255–269, 1983.
34. V.N. Vapnik, *Statistical Learning Theory*, Wiley: New York, 1998.
35. C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
36. B.E. Boser, I.M. Guyon, and V.N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th ACM Workshop on Computational Learning Theroy*, edited by D. Haussler, ACM Press, 1992, pp. 144–152.
37. T. Joachims, "Transductive inference for text classification using support vector machines,' in *Int. Conf. on Machine Learning (ICML)*, 1999.
38. H. Drucker, D. Wu, and V. Vapnik, "Support vector machines for spam categorization," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1048–1054, 1999.
39. George K. Zipf, *Selected Studies of the Principle of Relative Frequency in Language*. Harvard University Press: Cambridge, MA, 1932.

40. R.J. Chitashvili and R.H. Baayen, "Word frequency distributions," in *Quantitative Text Analysis*, edited by G. Altmann and L. Hřebíček, wvt, Trier, 1993, pp. 46–135.
41. B.B. Mandelbrot, "On the theory of word frequencies and on related Markovian models of discourse," in *Proceedings of Symposia in Applied Mathematics*, vol. XII, pp. l90–219, 1953.
42. G. Herdan, *The Advanced Theory of Language as Choice and Chance*, Springer, Berlin, 1966.
43. G. Salton and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw Hill: New York, 1983.
44. G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, pp. 513–523, 1988.
45. C. van Rijsbergen, *Information Retrieval*, Butterworths: London, 1979.
46. W. Lezius, R. Rapp, and M. Wettler, "A freely available morphological analyzer, disambiguator and context sensitive lemmatizer for German," in *Proc. COLING-ACL, 1998*. The program is available under http://psycho1.uni-paderborn.de/lezius.

Prof. Joachim Diederich is an Honorary Professor in the School of Information Technology and Electrical Engineering as well as the Centre for Online Health at the University of Queensland, Brisbane, Australia. Dr. Diederich's professional career includes four years at the German National Research Center for Information Technology (GMD) and more than two years at the International Computer Science Institute (ICSI) in Berkeley, California. He has also been a Full Professor at Queensland University of Technology (QUT) and served as an Adjunct Full Professor at the University of Strasbourg (France).

Dr. Diederich's qualifications include a Habilitation (Higher Doctorate) in Computer Science from the University of Hamburg (Germany), a Doctorate in Computational Linguistics from the University of Bielefeld (Germany), and a Masters degree in Psychology from the University of Münster (Germany).

Dr. Diederich's research interests are in the area of multimedia data mining with a special emphasis on natural language processing and text mining. Dr. Diederich has published 8 books (three as single author) as well as 140 journal articles, book chapters and peer-reviewed conference papers. This includes his 1988 book "Simulation schizophrener Sprache" (Modelling schizophrenic language), an extended study on the computer simulation of cognitive disorder in schizophrenia, and the more recent "Konstruktives konnektionistisches Lernen" (Incremental neural network learning) published in 2000.

Dr. Jörg Kindermann is a mathematician and linguist. After graduation at the University of Bielefeld in 1987 he worked at the German National Research Center for Computer Science. Since 2000 he is a senior research scientist at the Fraunhofer Institute Autonomous Intelligent Systems. His research interests are text and multimedia mining and statistical learning algorithms.

Dr. Edda Leopold is postdoctoral research fellow of the Knowledge Discovery Team of the Fraunhofer Institute for Autonomous Intelligent Systems. She has earned degrees in mathematics and musicology in 1989 and 1990 from the University of Gießen and received her Ph.D. in quantitative linguistics in 1998 from the University of Trier.

Dr. Gerhard Paass has studied mathematics, statistics, computer science, and economy and received a Ph.D. from Bonn University. He has designed statistical and knowledge-based algorithms at German National Research Center for Computer Science (GMD) since the mid eighties. Among others he has worked on probabilistic Bayes networks, neural networks, bootstrap methods, Bayesian Markov Chain Monte Carlo and Support Vector Machines. Currently he is leader of text mining and multimedia mining projects at Fraunhofer Institute Autonomous Intelligent Systems.