Chapter 8

DATA MINING AND TEXT MINING FOR SCIENCE & TECHNOLOGY RESEARCH

Edda Leopold, Michael May, and Gerhard Paaß

Fraunhofer Institut Autonome Intelligente Systeme, Sankt Augustin, Germany
E-mail: edda.leopold@ais.fraunhofer.de

Abstract: The goal of the paper is to give an overview on the state of the art of data mining and text mining approaches which are useful for bibliometrics and patent databases. The paper explains the basics of data mining in a non-technical manner. Basic approaches from statistics and machine learning are introduced in order to clarify the groundwork of data mining methods. Text mining is introduced as a special case of data mining. Data and text mining applications especially useful for bibliometrics and querying of patent databases are reviewed and three case studies are described.

1. INTRODUCTION: DATA MINING, TEXT MINING, INFORMATION RETRIEVAL

'Knowledge discovery' or 'data mining' is the partially automated process of extracting patterns or models from usually large databases. As a scientific discipline it is a relative newcomer, building on work in many areas such as machine learning, statistics, information retrieval, and database technology. From those individual disciplines it can be distinguished by its aim of integrating the various individual lines of research and by its stronger emphasis on technology and applications (Hand et al., 2001). The latter makes it a highly relevant topic for science and technology research.

H.F. Moed et al. (eds.), Handbook of Quantitative Science and Technology Research, 187-213.

In this paper we introduce basic approaches from statistics and machine learning in order to clarify the groundwork of data mining methods. Text mining which is especially relevant for science & technology research is introduced as a special case of data mining. The data mining techniques which are applied to text have to deal with the specific qualities of textual data: skewness of frequency distributions, synonymy, polysemy, sparseness of data, etc. This has important implications for the choice of data mining techniques which are suitable to the textual domain. Furthermore, there are techniques which are specific to text mining but are not covered by data mining proper (see section 3).

Information retrieval (IR) is a discipline which dates back to the seventies. It deals with the representation, storage, organisation, and access to information items (Baeza-Yates and Ribeiro-Neto, 1999). Given a user's query the goal of an information retrieval system is to retrieve information which might be useful or relevant to the user. Many techniques which have been developed in IR are nowadays employed in the area of text mining.

Text mining offers a variety of approaches for extracting information and knowledge from textual data:

- The *classification* of unlabelled text documents into a set of predefined classes can be used for the generation of ontologies and semantic spaces.
- *Clustering* of unlabelled documents according to their similarity can be deployed for the detection of related information.
- *Semantic spaces (also called topic maps)* can be utilised to obtain an insight into the semantic relations between documents in a document collection.
- The *segmentation* of texts into thematically coherent units and methods for detecting new and emerging topics in text documents allow a more efficient access to textual information.
- The *attribution of authorship* based on lexical and syntactical characteristics of the text can be used the for detection of plagiarism and therefore has implications to the management of intellectual property rights.
- *Named entity recognition* permits the search for persons, organisations, chemical substances and the like in textual data.

For all these tasks the internet and emerging topics such as the semantic web pose new challenges. After giving a survey on the various approaches (section 2), more concrete case studies are given to illustrate the data mining approach and its usefulness for bibliometrics and the management of patent databases (see section 4). It will be shown how web documents can be automatically inserted into a predefined ontology using text classification techniques. It will be illustrated how text mining can used to clarify disputed authorship and it will be described how semantic spaces can be used in order to refine the categorisation of an existing database of pre-classified documents.

2. DATA MINING FUNDAMENTALS

Data mining methods can be roughly divided into *supervised* and *unsupervised* methods. Supervised data mining methods learn from training data whereas unsupervised data mining methods use other cues such as, the Euclidean metric on input space.

In order to apply these methods to textual documents we have to represent them as numeric vectors, which can be readily processed by statistical estimation procedures. Surprisingly it is sufficient for many applications to simply count the number of occurrences of each word in a document, the so called *bag–of–words* representation (Salton and McGill, 1983). To indicate the utilisation of data mining methods for text mining, we will describe the different procedures using this representation. Later we will discuss alternative representations of documents in more detail.

The *supervised* data mining methods which we are going to present classify documents into predefined classes. This means they can be used to insert new documents into an already existing ontology. The *unsupervised* data mining methods cluster texts according to their (semantic) similarity or reduce the dimensionality of text representations.

2.1 Supervised Data Mining Methods

This section discusses advantages and disadvantages of the most important supervised learning methods used in data mining. During the last few years a number of estimation techniques have been proposed and evaluated which may be used for data mining and text classification. An example would be to automatically assign each incoming publication to a classification code of the *Science Citation Index* (SCI) such as ‘sport’, ‘politics’, or ‘arts’.

Whatever the specific method employed, a supervised data mining task starts with a *training data set*. A data mining method is given a set of instances (text documents) which are already labelled according to the class they belong to (‘sport’, ‘politics’, etc.). The task is then to learn a model based on the information provided in the training data (the words contained in the document) such that the document can be classified into one of the categories based on that information.

In the second step a *test data set* with the same general structure is given to the algorithm. However, the class information, although known to the user, is hidden from the algorithm. Using the model built in the first step, the algorithm classifies the instances into the predefined categories. Many more complex variants of these basic scheme exists, but the division into training and test data, where the model is built on the training set and its performance evaluated on the test set is common to most of them. This form of learning is called *supervised* learning, because the learning process is guided by the already known class information. The performance of the algorithm is measured by determining how successful the algorithm is in predicting the unknown class.

There are different assessment methods which measure the performance of a classifier. Suppose that there are two classes of documents in a document collection, a positive class ('sport') and a negative class ('non-sport'). Let tp and tn denote the number of documents that the classifier correctly identifies as positive and negative respectively, and let fp and fn denote the number of documents that are wrongly classified an positive or negative. *Precision* (prec.) and *recall* (rec.) are defined as follows:

$$\text{rec.} = \frac{tp}{tp + fp} \qquad \text{prec.} = \frac{tp}{tp + fn}.$$

In terms of Information Retrieval recall indicates how many of the relevant documents are retrieved and precision quantifies how many of the retrieved documents are in fact relevant. Obviously there is a trade off between precision and recall. When an IR system searches restrictively it may retrieve a few irrelevant documents, therefore precision is high. However, many relevant documents might have been overlooked, which corresponds to a low recall. When, on the other hand, the search is more exhaustive, recall increases and precision goes down. The *F*-score is a compromise between recall and precision for measuring the overall performance of a supervised classifier. It is defined as

$$F_\alpha = \left(\frac{\alpha}{\text{prec.}} + \frac{1-\alpha}{\text{rec.}} \right)^{-1},$$

where α is a factor which determines the weighting of precision and recall. A value of $\alpha = 0.5$ is often chosen for equal weighting of precision and recall. Accuracy (acc.) and Error (err.) are further assessment methods. They measure the fraction of correctly (or wrongly) classified documents in relation to the total number of documents. More formally

$$\text{acc.} = \frac{tp + tn}{tp + fp + tn + fn}, \quad \text{err.} = \frac{fp + fn}{tp + fp + tn + fn}.$$

Error and Accuracy are inappropriate performance measures for most text mining tasks, because the number of documents in the negative class is usually very large and so is the number of correctly classified negative documents. Therefore tn is large, which makes Accuracy less sensitive to the small but interesting quantities tp, fp, and fn. (Manning & Schütze 1999)

2.1.1 Naïve Bayes

Probabilistic classifiers rely on the assumption that the words of a document d_i have been generated by a probabilistic mechanism. For classification only the influence of the underlying class such as 'sports' or 'politics' is of interest. Therefore it is assumed that the class $c(d_i)$ of a document determines the probability $p(w_1,...,w_N|c(d_i))$ of its words. Now we may use the *Bayesian Formula* to determine the probability of some class if the words $w_1,\dots,w_N$ of a document are known

$$p(c_m \mid w_1,\dots,w_N) = \frac{p(w_1,\dots,w_N \mid c_m)p(c_m)}{\sum_{k=1}^{K} p(w_1,\dots,w_N \mid c_k)p(c_k)}.$$

Note that the documents may belong to one of K different classes. The *prior probability* $p(c_m)$ denotes the probability that some arbitrary document belongs to class c_m before its words are known. Often the prior probabilities of all classes may be taken to be equal. The conditional probability on the left is the desired *posterior probability* that the document with words $w_1,\dots,w_N$ belongs to the class c_m. We should assign the class with the highest posterior probability to our document.

For document classification it turned out that the specific order of the words in a document is not very important. Even more, we may assume that for documents of a given class a word appears in the document irrespective of other words. This leads to a simple formula for the probabilities of words

$$p(w_1,\dots,w_N \mid c_m) = \prod_{n=1}^{N} p(w_n \mid c_m).$$

Combining this with the Bayes formula defines the Naïve Bayes classifier. Simplifications of this sort are required because many thousand different words occur in a corpus.

The naïve Bayes classifier involves a training step which simply requires the estimation of the probabilities of words $p(w_n|c_m)$ in each class by its relative frequencies in a training sample. In the *classification step* the estimated probabilities are used to classify a new instance according to the Bayes rule. Although this model is unrealistic it yields surprisingly good classifications (Dumais et al. 1998, Joachims 1998). In contrast to other classification approaches it estimates the probabilities of classes. It may be extended in several directions (Lewis 1998; Sebastiani 2002).

2.1.2 k-Nearest Neighbour

Instead of building explicit models for the different classes we may select training documents which are "similar" to a test document. The class of the test document subsequently may be inferred from the class labels of the "similar" training documents. If k similar documents are considered the approach is also known as *k-nearest neighbour classification*. There are a large number of similarity measures used in text mining. If w_{in} is the count of the n-th word in a document d_i the *cosine similarity measure* of documents d_i and d_j is defined as

$$S(d_i,d_j)=\sum_{n=1}^{N} w_{in}w_{jn} \Big/ \sqrt{\sum_{n=1}^{N} w_{in}^2}\sqrt{\sum_{n=1}^{N} w_{jn}^2}\ .$$

Other similarity measures are discussed in (Baeza-Yates & Ribeiro-Neto 1999). For deciding whether a document d_i belongs to a class c_m the similarity $S(d_i,d_j)$ of all documents d_j in the training set is determined. The k most similar training documents (neighbours) are selected. The proportion of neighbors having the same class may be taken as an estimator for the probability of that class. If the largest proportion exceeds some threshold the corresponding class is assigned to the document d_i. The threshold as well as the optimal number k of neighbours may be estimated from additional training data by cross-validation.

Nearest neighbour classification is a non-parametric method and it can be shown that for large datasets the error rate of the 1-nearest neigbor classifier is never larger than twice the optimal error rate (Hastie et al., 2001). Several studies have shown that nearest neighbour methods have very good performance in practice (Joachims, 1998; Yang, 1999). Their drawback is the computational effort during classification, where basically the similarity of a document with respect to all other documents of a training set has to be determined. Some extensions are discussed in (Sebastiani, 1991).

2.1.3 Support Vector Machines

A Support Vector Machine (SVM) is a supervised classification algorithm which recently has been applied successfully to text classification tasks. SVMs have proved to be an efficient and accurate text classification technique (Joachims, 1998; Dumais et al., 1998; Drucker et al., 1999, Leopold and Kindermann, 2002). Like other supervised machine learning algorithms, an SVM works in two steps. In the first step — the *training* step — it learns a decision boundary in input space from preclassified training data. In the second step — the *classification* step — it classifies input vectors according to the previously learned decision boundary.

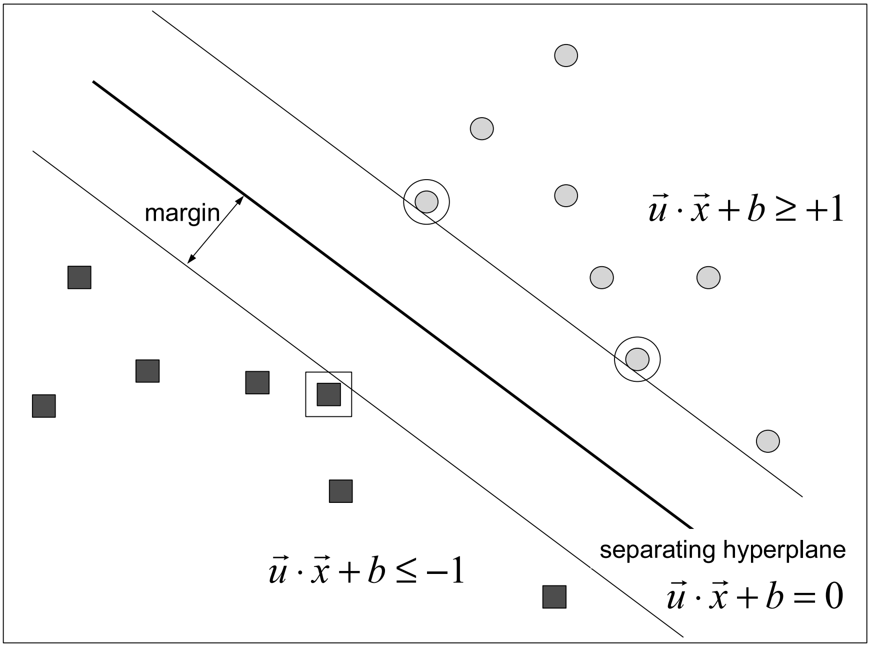
margin
$\vec{u}\cdot\vec{x}+b\geq+1$
$\vec{u}\cdot\vec{x}+b\leq-1$
separating hyperplane
$\vec{u}\cdot\vec{x}+b=0$

Figure 8.1. A decision hyperplane separates two classes (squares: y=-1 and circles: y=+1). The SVM algorithm seeks to maximise the margin around a hyperplane that separate a positive class from a negative class. The support vectors are framed.

A *single* support vector machine can only separate *two* classes: a positive class c_1 (indicated by $y = +1$) and a negative class c_2 (indicated by $y = -1$).

The SVM aims to find a class separating hyperplane with the largest possible margin, see Figure 8.1. (Vapnik, 1998; Joachims 2002) This results in a hyperplane which is defined by a normal vector $\vec{u}$ and an offset b .

In the classification step an unlabelled 'term frequency' vector is estimated to belong to the class

$$\hat{y} = \operatorname{sgn}(\vec{u} \cdot \vec{x}_i + b) \tag{2}$$

where $\vec{x}_i$ is the term frequency vector which represents document d_i. SVMs can be adjusted to different geometries in the feature space replacing the dot product by a kernel function $K(\vec{x}_i, \vec{x}_j)$. We have observed, however, that in the case of textual data the choice of the kernel function has a minimal effect on the accuracy of classification: Kernels which imply a high-dimensional feature space show slightly better results in terms of precision and recall, but they are subject to overfitting. (Leopold & Kindermann 2002)

The most important property of SVMs is that learning is independent of the dimensionality of the feature space. SVMs seek for the hyperplane that separates the two classes with the maximal margin (see Figure 8.1). Thus the separating hyperplane is defined in terms of data points touching the margin — the support vectors — rather than by coordinates of the feature space. This allows for a good generalisation even in the presence of a large number of features and makes SVM especially suitable for the classification of texts.

2.1.4 Decision Trees

Decision trees are classifiers which consist of a set of rules which are applied in a sequential way and finally yield a decision. They can be best explained by observing the training process, which starts with a comprehensive training set. It uses a divide and conquer strategy: For a training set M with labelled documents the word w_i or term is selected, which can predict the class of the documents in the best way. Then M is partitioned into two subsets, the subset M_i^+ with the documents in which w_i occurs, and the subset M_i^- with the documents without w_i. This procedure is recursively applied to M_i^+ and M_i^-. The procedure stops if all documents in a subset belong to the same class c_j generating a tree of rules with an assignment to actual classes in the leaves.

Decision trees are a standard tool in data mining (Mitchell, 1997). They are fast and scalable both in the number of variables and the size of the training set. For text mining, however, they have the drawback that the final decision depends only on relatively few terms. A decisive improvement may be achieved by boosting decision trees. This results in determining a set of complementary decision trees constructed in such a way that the overall error is reduced. Schapire and Singer (2000) use even simpler one step decision trees that contain only one rule and get impressive results.

2.2 Unsupervised Learning

Unsupervised learning methods aim at extracting all interesting patterns directly from the data. They do not require training data. In this section we describe clustering and reduction of dimensions.

2.2.1 Clustering

Clustering is one of the core data mining techniques. It refers to an unsupervised learning process in which individual items are grouped on the basis of their mutual similarity or distance. Again it is necessary to define implicitly or explicitly a similarity measure between documents. We may represent each document as a vector of word frequencies in some order and use the Euclidean distance, the cosine similarity (which has been defined in section 2.1.2) or some other similarity measure.

Clustering of documents has already been developed in the seventies (see, e.g., van Rijsbergen, 1979). A very simple clustering algorithm resembles the k-NN clustering in section 2.1.2. It includes documents d_i and d_j in the same cluster when the similarity measure $S(d_i, d_j)$ is below a given threshold.

Hierarchical clustering techniques create clusters by iteratively merging (agglomerative clustering) or splitting (divisive clustering) of previously identified clusters. The process of hierachical clustering therefore leads to the creation of a dendrogram, which is a tree of clusters, allowing one to adjust the clustering granularity according to the current needs.

Partitional techniques start with a fixed number of clusters, which are improved iteratively. A prominent member is the *k*-means algorithm (Hartigan, 1975). Its principle is that each cluster is represented by the means of all its members and serves as a basis for an iterative cluster regrouping. There exist a large number of clustering schemes, a survey is given by Hastie et al. (2001).

A specific variant is model based clustering, which assumes that the clusters are generated according to a statistical model. For text documents discrete distributions, e.g., multinomial distributions are most appropriate. This allows statistical techniques to estimate the most probable clustering and the adequacy of clusters (Nigam et al., 1999). The similarity measure is implicitly defined by the distributional model.

If there is no statistical model it is difficult to determine the optimal number and the validity of clusters. Potential cluster quality measures such as cluster stability, cluster compactness, or inter-cluster separation can be quantified with cluster validation indices.

2.2.2 Dimensionality Reduction

One key approach to data mining is the reduction of a large number of variables to a few constructs which capture the 'main' properties of the data. This is especially interesting for text mining where many thousand variables are common. We start with the term document matrix A in which each row contains the count of words of a document. *Principal component analysis* is a well-known approach from multivariate statistics (Hastie et al., 2001). It starts with the correlation matrix $A'A$ of all variables and uses eigen analysis to determine the largest eigenvectors of the correlation matrix. *Latent semantic analysis* (LSA) is a technique popular in text mining (Deerwester et al., 1990) which can be shown to yield the same results as principle component analysis (Thisted, 1988). A large number of comparable techniques has been discussed under the name of *factor analysis*.

The factors can be considered as independent linear combinations of the original variables that explain a maximal proportion of the variation in the dataset. In subsequent analyses, e.g., classifications or similarity computations, they may be used instead of the original variables without losing too much expressive power. The similarity of documents in terms of the principal components may be interpreted as topical similarity and can be used to find related documents, or documents matching some specified query. By this approach we may even estimate the similarity of documents even if they do not have any words in common. (Landauer and Dumais, 1997).

If different words have a high correlation to the same factor, this often indicates a similar meaning, i.e., synonymy. On the other hand the same word may have substantial correlations to two or more factors, which may indicate different meanings of the same word, i.e., polysemy.

One objection to latent semantic indexing is that it relies on the correlation matrix, and implicitly minimises square distances. More appropriate for count data in text mining is *probabilistic latent semantic analysis* (Hofman, 2001). It assumes a discrete unobservable variable z (latent factor) which may take the values 1 to m. The model assumes that for each word w_{ij} in a document d_i a value of the latent factor is generated according to a document specific distribution $p(z|d_i)$. Depending on the value of the latent factor the word w_{ij} then is generated according to a factor-specific distribution $p(w_{ij}|z)$.

The probabilities are estimated in an iterative way using the Expectation Maximisation (EM) algorithm which has been introduced by Dempster, Laird and Rubin (1977). Probabilistic latent semantic analysis (PLSA) results in a better linguistic interpretability and is compatible with the well corroborated linguistic models (see Chitashvili and Baayen 1993) of word frequency distributions.

LSA and PLSA have an interesting application to bibliometric search problems: The *Science Citation Index* offers a search by words or a combination of words. This means that documents that do not contain the word cannot be retrieved although they might deal with the requested content. Using LSA or PLSA queries and documents (or abstracts) could be mapped to its latent factors. This would enable a concept oriented search where synonyms of the word also indicate relevance to the query, and documents in which a word appears in a different meaning are rejected.

3. TECHNIQUES SPECIFIC TO TEXT MINING

As mentioned previously, text mining is data mining applied to natural language texts. The main issues which are connected with the transfer of general data mining techniques to the textual domain are the representation of texts, its pre-processing and the special statistical characteristics of textual data, which constitutes a special challenge to data mining algorithms.

Although the bag–of–words representation is very simple and effective, it neglects the succession of words in the texts and therefore abstracts away from the syntactic relations which exist between the different linguistic units. Furthermore, the level of analysis is not confined to words. Units that are smaller than words yield good results when small corpora are considered.

3.1 Morphological Pre-processing and Feature Selection

Preprocessing is concerned with the elimination of textual information which is irrelevant or even misleading to solving the subsequent data mining task. As a rule of thumb half of the words occur only once even in a large text corpus of some million running words (52% of the words of the corpus displayed in Figure 8.2 are hapax legomena). These words cannot occur in both the training set and the test set. They are therefore omitted.

Stemming is another pre-processing method in which the words that occur in the corpus are mapped to a basic form. Stemming is a more general notion than lemmatisation where the basic form is linguistically defined. The Porter Stemmer (Porter, 1980), which performs a cascade of regular expressions, is often used for English texts. Stemming, however, is a more challenging task for other languages that are more productive at the morpho-syntactic level.

Stemming can be performed at different levels of depth and has to be used with care. Resolving every morpho-syntactic rule assuredly leads to a loss of information. But even the removal of the plural morpheme can result in a loss of semantic information. Stricker et al. (2000) give an example for French: The word 'action' in the sentence "Le jugement est plus nuancé selon le domaine d'*action* du gouvernement." the word 'action' can be translated with action. However, in the sentence "Den Danske Bak a acquis en décembre dernier 90% des actions de Fokus Bank." the word 'actions' means shares, and this meaning is clearly indicated by the plural ending.

In some languages it is useful to split complex words such as, e.g., compounds into their morphological components, and preserve them for subsequent processing. The resulting features can be reduced further by applying other feature selection. Compound splitting can thus be considered as a sub-task of stemming, where compounds are split into its components. This is less necessary for the English language, but compound splitting is usually beneficial when applied to compounding languages like German or Dutch.

3.1.1 Term weighting and selection

Some words in the language's vocabulary are very frequent and equally distributed amongst the documents in the corpus. In many cases these words are superfluous from a statistical viewpoint and should be removed prior to the application of data mining algorithms. There are different techniques for performing this task.

The simplest method for the removal of uninformative words is to use a predefined list of stop words, and to delete all words in the text that match an element of the list. Stop word lists typically consist of function words (articles, pronouns, and conjunctions). The problem of stop word lists is that they may be inappropriate to the corpus or the task in question. In a corpus of texts on computers the word 'computer' will probably be equally distributed amongst the documents and thus fairly uninformative. In such a case the word 'computer' should be treated as a stop word. When the task is authorship recognition function words may be important cues for authorship recognition, although they are unlikely to be useful for content classification.

Other methods of identifying uninformative terms make direct use of its statistical distribution among the texts of the corpus. When pre-processing is performed prior to classification the distribution of terms in different classes in the training set can be compared against each other. Terms are omitted when a statistical test suggests that they are equally distributed in different classes. Pearson's chi-squared test, for instance, has been applied successfully (Paaß et al., 2002) prior to text classification based on sequences of syllables. Similar techniques such as information gain, mutual information, cross entropy or odds ratio are described in (Mladenic and Grobelnik, 1999).

Term weighting schemes like the well known *inverse document frequency* exploit the distribution of term frequencies in the text. The number of documents in which a given term appears is called *document frequency* denoted as *df*. The so called inverse document frequency, which was defined by (Salton, 1983) as $idf = (\log df)^{-1}$, is widely used in the literature on automatic text processing in order to tune term–frequencies according to the thematic relevance of a term. Other term weighting schemes such as, e.g., the redundancy used by (Leopold and Kindermann 2002) consider the entire distribution over the documents rather than solely the number of texts. A survey about different weighting schemes is given in (Manning and Schütze 1999). Although *idf* is the most popular weighting scheme, other indexing functions have also been used, including probabilistic techniques (Gövert et al., 1999). These weighting schemes are especially useful when the length of the documents exceeds some 1,000 words.

3.1.2 Statistic properties of textual units: Zipf's law

The skewness of frequency distribution makes linguistic data a special challenge for any statistical method of analysis. Zipf's law, which is empirically very well confirmed, ensures that the *r*-most frequent word in the corpus occurs $f(r) = a/(c+r)^g$ times, where *a*, *c* and *g* are parameters. (*g* varies from 1 corresponding to normal language use to 2 corresponding to a fairly restricted or standardised language use such as, e.g., in the Reuters news wire corpus.). Interestingly Zipf-like distributions hold for nearly any type of linguistic units: frequencies of syllables or lemmas also follow the Zipf distribution. As a rule of thumb the parameter *g* decreases with increasing unit size. (Note that double logarithmic scaling in Figure 8.2 displays functions of the form $f(r) = a/r^g,\ a,r > 0$ as a straight line with negative slope.) One consequence of Zipf's law is that the word frequency distribution is very skewed, i.e., some few words are very frequent (some 10^5 occurrences), whereas the frequency of most of the words is some magnitudes smaller (< 10). Usually (i.e., unrestricted language use assumed) half of the words in a text corpus occur only once. Unfortunately it is well known that especially the rare words are particularly informative about the content of the document in which they occur. This means that rare words may not be considered irrelevant and may not be omitted prior to the text mining process.

A further consequence of Zipf's law is that most of the words in the corpus are absent in most of the documents in the corpus. This phenomenon is usually addressed as the sparse data problem. The so called bag–of–words representation of documents, which counts the number of occurrences of each word in the document and thus ignores the succession of words, is often used for text mining purposes. Text corpora often contain some millions of running words and some 100,000 different word types, and most of the types which occur in the corpus do not occur in a particular document. Therefore vectors that result from the bag–of–word representation are sparse (i.e., most entries equal zero) and moreover they are very long (some 100,000 dimensions).

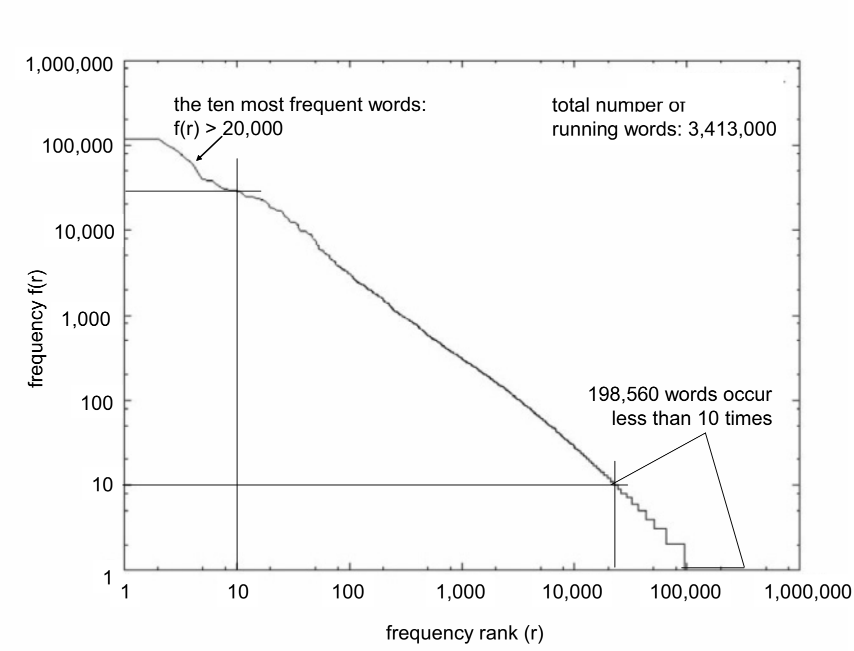

Figure 8.2. Rank-Frequency Distribution of Frankfurter Rundschau, July 1998

3.1.3 Subword Units

In order to relieve the sparse data problem it is sometimes useful to use units that are smaller than words, so called sub-word units (sequences of letters, or syllables) instead of words. Sub-word units, yield good results especially when small corpora are considered. The *F*-scores presented in Figure 8.3 were obtained by a SVM classifier from a corpus of German radio programs. The corpus consists of 950 documents comprising about 650 running words each. The texts were converted to a phonetic transcript using the BOSS II speech synthesis system (Stöber et al., 2000). In the corpus the words consist on average of 5.3 phonemes, and syllables comprise on average 2.8 phonemes. We experimented with different kinds of units: sequences of 2 to 6 phonemes (phoneme-*n*-grams), sequences of 1 to 6 syllables (syllable-*n*-grams) and 1 to 3 words.

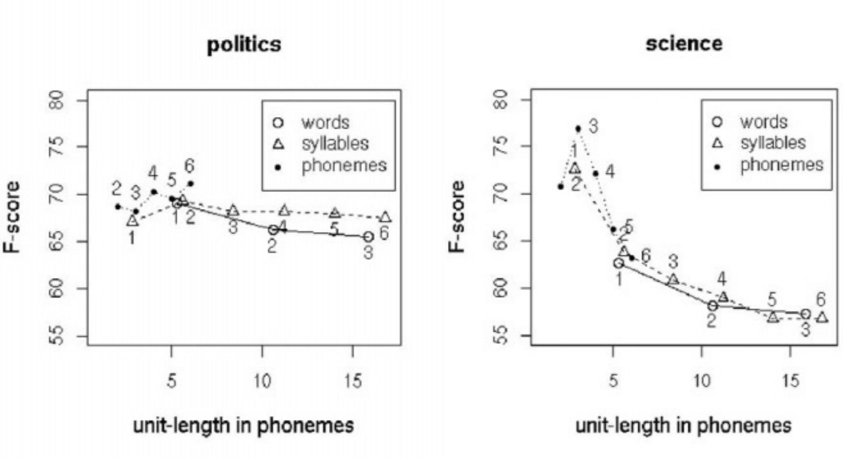

Figure 8.3. Classification accuracy achieved with different representations of texts. Left panel: large class (220 documents); right panel: medium size class (120 documents).

In Figure 8.3 the *F*-scores are presented for the largest class in the corpus, 'politics', which comprises 220 documents, and for a smaller class, 'science' comprising 120 documents. One can see that accuracy of SVM Classifier depends strongly on the size of the units. The optimal unit size is smaller when the class contains fewer documents (3 phonemes for the class 'science'). The large class makes little profit from small unit sizes.

An explanation for these results is the well known fact that there is a trade off between the length of linguistic units and their frequency. This has already been proven for words (Guiter, 1974), but it also applies to other units like sequences of syllables. This means that the smaller the categories and the longer the units (types) the more types become so improbable that they exclusively occur in either the test set or the trainings set. Therefore units have to be shorter in order to compensate for the small size of the class.

3.1.4 Shallow parsing

The notion Natural Language Parsing addresses the task of automatically detecting the linguistic structure of the sentences of a text. Although the above mentioned bag–of–words representation yields surprisingly good results for many text mining applications, it seems obvious that subsequent algorithms work the better the more structure is extracted from the original texts. However, even if it were possible to formalise and represent the complete grammatical and lexical structure of a natural language, the parser, would still need a very high degree of robustness and efficiency. Realising such a system for large amounts of texts is impossible for the time being. This has led to so called shallow parsing approaches, in which certain language regularities which are known to cause complexity problems are handled in a pragmatic way (Neumann & Piskorski, 2002).

Neumann and Schmeier (2002) showed, for example, that morphogical analysis of short German texts seems to be a better choice than simple trigramming. For the collection of small documents (emails of 60 words in average) SVM yielded the best results when combined with a shallow parsing compared to trigrams or morphs (accuracy 61.42 vs. 58.29 and 54.29 respectively). For the collection of longer texts (average length 578 words) there was no significant difference in accuracy between morphs and trigrams.

3.2 Presentation of Results

3.2.1 Graphical representation

Self-organising Maps (SOM) were invented in the early 80s (Kohonen, 1980). They use a specific neural network architecture to perform a recursive regression leading to a reduction of the dimension of the data. For practical applications SOMs can be considered as a distance preserving mapping from a more than three-dimensional space to two-dimensions. A description of the SOM algorithm and a thorough discussion of the topic is given by Kohonen (1995).

Figure 8.4 shows an example of a SOM visualising the semantic relations of news messages. First the news messages were represented in a four-dimensional semantic space, where each coordinate corresponds to a topic category (‘culture’, ‘economy’, ‘politics’ and ‘sports’). Then the SOM algorithm is applied (with 100×100 nodes using Euclidean metric) in order to map the four-dimensional document representations to two dimensions admitting a minimum distortion of the distances. The grey tone indicates the topic category.

3.2.2 Text summarisation

Visualisations described above have to be accompanied by some description providing the user with some additional information about the content of the respective documents. One of the options for providing this additional information is text summarisation. Text transformation can be defined as a reductive transformation of a source text by the selection and generalisation of what is important in the source (Spark-Jones, 1999).

In the currently most explored approach for summarisation the summary is composed of sentences which are selected as semantically representative for the document content. An example of such an extractive multi-document summarisation approach is given by Kraaij et al. (2002).

Other recently addressed text summarisation research topics have been multi-lingual summarisation and hybrid multi-source summarisation (Chen, 2002). The potential of concepts and conceptual relations as a vehicle for terminological knowledge representation has been exploited by the knowledge–based text summarisation approach (Hahn & Reimer, 1999).

4. CASE STUDIES

4.1 Case study 1: Classification of Web Documents

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents and services (Kosala & Blockeel 2000). Three main tasks may be solved by web mining: *Web structure mining* tries to discover regularities underlying the link structures in the web. *Web usage mining* evaluates the frequency and temporal sequence in the actual access of web pages by users. *Web content mining* describes the discovery of useful information from web documents.

The EU project Diastasis aims at describing the Web surfing behaviour of citizens. The ultimate goal is to establish representative statistics which, on the one hand, describe the socio-economic situation of people, and, on the other hand, the content of the web pages they download. This information will be compiled and distributed by national statistical institutes.

A representative sample of people, who have given their consent, is observed with respect to their web surfing behaviour. This means that the IP addresses of web pages they download as well as the exact time of download are recorded in a log file.

In order to describe the content of web pages a hierarchy of meaningful categories is required, which is expressive enough to cover all aspects of web pages. We used the Yahoo web directory (http://www.yahoo.com) for this purpose, as each category is linked to example pages which can be used for training purposes. The idea is to train a text classifier for the higher categories of the hierarchy. This offers the opportunity to classify the content of arbitrary web pages into meaningful categories.

Initially each linked HTML document from the directory is downloaded. Images and HTML code are deleted so that only text remains. This text is pre-processed with the usual pre-processing steps (stemming, stopword removal, etc.) described above. Finally SVM classifiers are trained for each category of interest.

To describe the content of web pages used by the citizens the corresponding pages have to be downloaded again according to their IP addresses. Subsequently these pages may be classified into one or more categories of the ontology.

A difficulty arises because the documents contain different languages. For the specific population of Barcelona the main languages are Spanish, Catalan and English. To solve this problem a language classifier is used which assigns documents to languages based on triples of letters. Yahoo directories exist for all three languages and the directories have nearly identical structure. Hence classifiers can be trained for the Yahoo hierarchy in these three languages.

To identify the users's surfing behaviour the temporal sequence of page views has to monitored. To accommodate this user sessions are identified. A session is defined as a continuous browsing period containing no idle periods longer than thirty minutes of length. It is non-trivial to create a completely accurate session for a user, considering the web log mechanism. For example, when a user accesses a locally cached web resource this access does not appear in the log; thus repeated visits to sites will often only appear once. Our method of recreating sessions via external observation is also susceptible to error. If a user spends thirty minutes reading a web page before visiting the next page in the logical session, they will be identified as two separate sessions.

Dynamic web sites also present a challenge. Here a page is dynamically constructed by the server and may change after each visit; for example, most popular portal web sites. Therefore it is difficult to build an accurate classification model of the site's pages, as an access after some time may yield completely different content. These difficulties can be overcome with a more extensive logging system and real time training and classification.

To combine the web content data with the user data we first have to describe the temporal sequence of web utilisation, e.g., accessing financial information in the morning and visiting web auctions in the evening. This is a task of web usage mining. The resulting web surfing information is combined with socio-economic user data collected by questionnaires and surveys. The final information will be published by national statistical institutes.

The method outlined here can also be used for bibliographical purposes. Publications, possibly in different languages, are first classified according to their language and then inserted into different topic categories, which might be defined, for instance, by abstracts and their classification codes of the *Science Citation Index* (instead of the documents of the Yahoo web directory).

4.2 Case study 2: Authorship Attribution

Authorship attribution can be considered as a special case of text classification, in the sense that a text is classified according to whether it was written by a specified author or not. However various approaches to authorship attribution differ significantly from usual text classification techniques.

There are a number of statistical techniques which have been imported from other fields and which dominate the field of computer–based authorship attribution. Most notably are the Efron-Thisted Test (Thisted & Efron, 1987), QSUM (or cusum) (Holmes, 1998) feed–forward artificial neural networks (Tweedie et al., 1996), Radial Basis Function (RBF) networks (Lowe and Matthews, 1995), genetic algorithms (Holmes and Forsyth, 1995), Recurrent neural networks (Towsey et al., 1998). According to Rudman (1998) approximately 1,000 style markers have already been isolated for authorship attribution. There is, however, no agreement of significant style markers amongst researchers

Authorship attribution has previously suffered from the problem that the important features in a document are unknown and that a text as a whole cannot be analysed. The use of a limited set of function words or 'short words' is clearly restrictive and there is an ongoing discussion on the relevance of appropriate style marker.

This calls for the application of techniques such as support vector machines which allow one to process bag–of–words representations of complete texts rather than a small number of selected features. SVMs can process documents of significant length, databases with a large number of texts and do not require pre-determined features. As Leopold and Kindermann (2002) show, SVM are capable of managing input vectors with a very large number of dimensions (up to 400,000), with no term selection required.

For the experiments on authorship attribution data from the Berliner Zeitung (BZ), a daily newspaper in Berlin, were used. We used the data from December 1998 to February 1999. The articles are divided into twelve topics. From the three largest topics: politics (1,200 articles); economy (550 articles); and local affairs (3,233 articles) all articles with more than 200 words were considered. The resulting corpus consisted of 2,652 documents, about 1,900,000 running words (tokens) and about 120,000 different words (types). These documents were represented in two different ways:

1. A vector word counts was generated from the document. Neither stemming nor stop word removal was performed. (a simple bag–of–words representation);
2. For each document tagwords were extracted and bigrams were generated from them. Additionally the number of words with a given word length is counted. The document is represented by the vector of counts of tagwords, of bigrams, and of word lengths.

An SVM classifier was trained for seven authors in the corpus. Different parameter settings have been applied and five-fold cross-validation was performed. Tables 8.1 and 8.2 show the results for the optimal choice of parameters (Diederich et al., 2003).

Table 8.1. Results of classification based on word forms

name of author	*target author*		*other authors*		*percent*	
	# correct	*# false*	*# correct*	*# false*	*precision*	*recall*
Aulich	94	14	2652	0	100.0	87.0
Fuchs	98	20	2642	0	100.0	83.1
Kunert	71	29	2659	1	98.6	71.0
Muenner	80	7	2673	0	100.0	92.0
Neumann	73	38	2647	2	97.3	65.8
Schmidl	66	28	2666	0	100.0	70.2
Schomaker	25	57	2678	0	100.0	30.5

In most cases the target author was recognised correctly. Notably the number of false negative classifications is extremely low. That is erroneously attributed authorship is very improbable. There is no significant difference between the results for words (Table 8.1) and tagwords (Table *8.2*). This suggests that SVM combined with the bag–of–words representation in its simplest form is sufficient for a reliable identification of authorship.

Table 8.2. Results of classification based on tagwords, bigrams of tagwords, and word lengths

name of author	*target author*		*other authors*		*percent*	
	# correct	*# false*	*# correct*	*# false*	*precision*	*recall*
Aulich	85	23	2652	0	100.0	79.0
Fuchs	89	29	2642	0	100.0	75.0
Kunert	61	39	2660	0	100.0	61.0
Muenner	67	20	2673	0	100.0	77.0
Neumann	51	60	2649	0	100.0	46.0
Schmidl	60	34	2666	0	100.0	63.8
Schomaker	17	65	2677	1	94.4	21.0

Authorship attribution using SVMs allows for the verification of a pretended authorship given that enough training examples that is, previous publications of the author in question are available. Note that precision is very high in the results presented above. This means that if the classifier is able to identify an author its decision is very reliable.

4.3 Case study 3: Classifier Induced Semantic Spaces

Latent Semantic Analysis (Landauer and Dumais, 1997) as well as Probabilistic Latent Semantic Analysis (Hofman, 2001), which is described in section 2.2.2, are often used for the construction of semantic spaces. Semantic spaces typically reflect some aspect semantic nearness of linguistic units. The dimensions of such a semantic space are often interpreted as ‘artificial concepts’ which represent common meaning components of different words and documents. Such artificial concepts, however, cannot be interpreted in a semantically transparent way.

Another way of generating semantic spaces which produce a semantically transparent representation can be constructed from the internal representation of supervised text classifiers. Recall the classification step of Support Vector Machines, described in section 2.1.3. Estimating the class membership by equation (2) consists of a loss of information since only the algebraic sign of the right hand term is evaluated. However, the value of

$$v = \vec{u} \cdot \vec{x} + b$$

in equation (2) is a real number and can be used in order to create a real valued semantic space, rather than just to estimate if $\vec{x}$ belongs to a given class or not.

Suppose there are several, say K, classes of documents. Each document is represented by an input vector $\vec{x}$. For each document the variable $y_i^k \in \{-1,+1\}$ indicates whether $\vec{x}$ belongs to the *k*th class, $k = 1,\ldots,K$, or not. For each class an SVM can be trained which yields the parameters $\vec{u}^k$ and b^k. After the SVMs have been learned, the classification step (equation (2)) can be applied to a (possibly unlabeled) document represented by $\vec{x}$ resulting in a *K*-dimensional vector $\vec{v}$, where the *k*th component is given by

$$v^k = \vec{u}^k \cdot \vec{x} + b^k$$

The component v^k quantifies how much a document belongs to class *k*. Thus the document represented by its term frequency vector is mapped to the *K*-dimensional vector in the classifier induced semantic space. Each dimension in this space can be interpreted as the membership degree of the document to each of the *K* classes.

Figure 8.4 shows a Self-organising Map (see section 3.2.1), which is generated from a classifier induced semantic space. SVMs for the four classes 'culture', 'economy', 'politics', and 'sports' were trained by news messages from the 'Basisdienst' of the German Press Agency (dpa) April 2000. Classification and generation of the SOM was performed for the news messages of the first 10 days of April. 50 messages were selected at random and displayed as white crosses. The categories are indicated by different grey tone. Shadings within the categories indicate the confidence of the estimated class membership.

It can be seen that the change from sports (15) to economy (04) is filled by documents which cannot be assigned confidently to either classes. The area between politics (11) and economy (04), however, contains documents, which definitely belong to both classes. Note that classifier induced semantic spaces go beyond a mere extrapolation of the annotations found in the training corpus. It gives an insight into how typical a certain document is for each of the classes. Furthermore Classifier induced semantic spaces allow one to reveal previously unseen relationships between classes. The bright islands in area 11 on Figure 8.4 show, for example, that there are messages classified as economy which surely belong to politics.

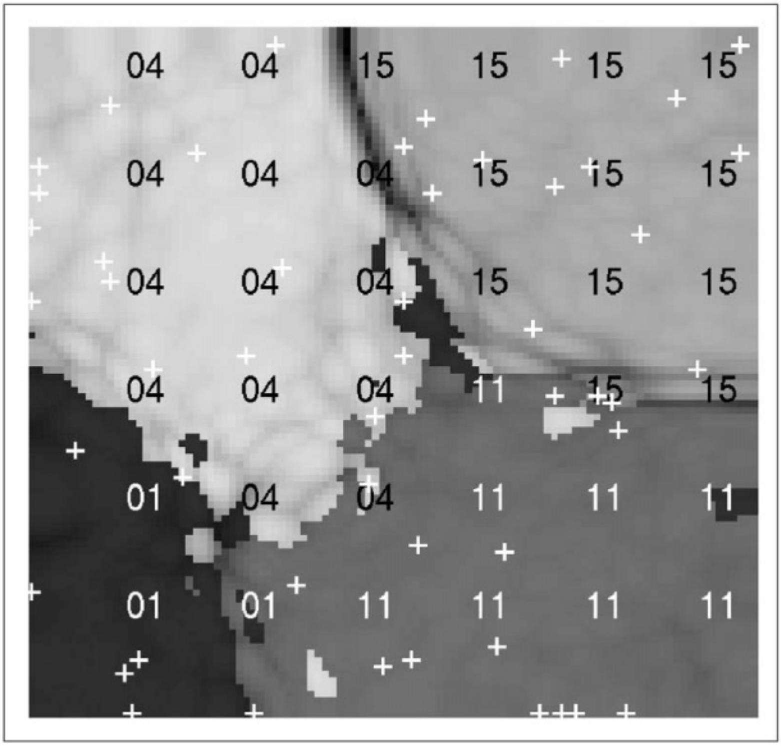

Figure 8.4. SOM of a classifier induced semantic space. The numbers indicate the classified topics 01: culture, 04: economy, 11: politics, 15: sports.

Classifier induced semantic spaces can also be used for bibliometric purposes such as, e.g., a refinement of the coarse–grained classification codes of the *Science Citation Index* or the identification of emerging topics; just substitute the topic classes by the respective classification codes and train SVMs by the SCI abstracts.

- *Refinement of search*: When a document collection is represented in a SOM like the one in Figure 8.4 one can see how much a document belongs to different classes. A document which is located at the border between two classes (say 'sports' and 'art') is likely to belong to both classes (perhaps they deal of dance) although they might be classified into only one class. Representing abstracts in a SOM allows for searching exactly these borderline documents.

- *Detection of subcategories:* Clusters which can be observed on a SOM are likely to belong to a 'sub-category' which is not explicitly represented in the coding scheme of the *Science Citation Index*. This results a more fine–grained categorisation of the documents collection. The thematic domain of the new subcategory can be inferred from the position on the SOM, since the dimensions of the semantic space are semantically interpretable. *Clustering* of unlabelled documents according to their similarity can be deployed for the detection of related information.
- *Emerging topics:* Publications which represent unusual themes are likely to be separated from all other documents in the space. Items which do not fit to the category codes are situated in the negative simplex of the coordinate system. When publications in the negative simplex $v^k < 0,\ k = 1,\ldots,K$, accumulate it is likely that a 'new' topic has emerged, which is not covered by the present classification codes.

5. CONCLUSION

Text mining offers a variety of methods for the automatic analysis of texts, which can also be gainfully applied to bibliometric problems and patent statistics. Latent Semantic Analysis and the more recently developed Probabilistic Latent Semantic Analysis allows for a concept oriented rather than key-word based search in bibliographical indices or patent databases. The classification of publications or abstracts into a predefined ontology like the one defined by the classification codes of the SCI can be done automatically using document classification techniques. Authorship attribution can be used for the detection of plagiarism. Classifier induced spaces can be utilised for the identification of emerging topics. Self–organising Maps can help to detect subcategories. They can also be used for a refined search in a collection of categorised abstracts or documents.

REFERENCES

Andrews, R., Geva S. (1994). *Rule extraction from a constrained error backpropagation MLP*. Australian Conference on Neural Networks, Brisbane, Queensland 1994 (pp. 9–12).

Baeza-Yates, R., Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison-Wesley.

Chen, H.H. (2002). *Multilingual summarization and question answering*. Workshop on Multilingual Summarization and Question Answering, COLING'02, Taipeh, Taiwan 2002.

Chitashvili, R.J., Baayen, R.H. (1993). *Word frequency distributions*. In G. Altmann, L. Hřebíček (Eds.), Quantitative Text Analysis (pp. 54–135). Wvt: Trier.

Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science,* 41 (6), 391–407.

Dempster, A.P., Laird, N.M., Rubin, D.B. (1997). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society,* B, 39, 1–38.

Diederich, J., Kindermann, J., Leopold, E., Paaß, G. (2003). Authorship attribution with Support Vector Machines. *Applied Intelligence,* 19 (1–2), 109–123.

Dumais, S., Platt, J., Heckerman, D., Sahami, M. (1998). *Inductive learning algorithms and representations for text categorization*. In Proceedings of the 7th International Conference on Information and Knowledge Management (pp. 148–155). ACM.

Gövert, B., Lalmas, M., Fuhr, N. (1999). *A probabilistic description–oriented approach for categorising Web documents*. In Proceedings of CIKM-99, 8th ACM International Conference on Information and Knowledge Management, Kansas City, Missouri, 1999 (pp. 475–482). ACM.

Guiter, H. (1974). *Les rélations fréquence – longueur – sens des mots (langues romanes et anglais),* In XIV congresso internazionale di linguistica e filologia romanza (pp. 373–381). Napoli.

Hahn, U., Reimer, U. (1999). *Knowledge–based text summarization*. In: I. Mani, M. T. Maybury (Eds.), Advances in Automated Text Summarization (pp. 215–232). Cambridge, London: MIT-Press.

Hand, D., Mannila, H., Smyth, P (2001). *Principles of data mining*. MIT Press.

Hartigan, J.A. (1975). *Clustering algorithms.* New York: John Wiley.

Hastie T., Tibshirani, R., Friedman, J. (2001). *The elements of statistical learning.* New York: Springer.

Hofman, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning,* 42, 177–196.

Holmes, D.I. (1998). The evolution of stylometry in Humanities Scholarship. *Literary and Linguistic Computing,* 13 (3), 111–117.

Holmes, D.I., Forsyth, R.S. (1995). The Federalist revisited: New directions in authorship attribution. *Literary and Linguistic Computing,* 10 (2), 111–127.

Kohonen, T. (1980). *Content–adressable memories.* Springer.

Kohonen, T. (1995). *Self-organising Maps.* Springer.

Kosala, R. Blockeel, H. (2000). *Web mining research: A Survey*. In P.S. Bradley, S. Sarawagi, U.M. Fayyad (Eds.), SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM, 2 (pp. 1–15). ACM Press.

Kraaij, W., Spitters, M., Hulth, A. (2002). *Headline extraction based on a combination of uni- and multidocument summarization techniques.* In Proceedings of the ACL workshop on Automatic Summarization/Document Understanding Conference DUC 2002 , June 2002, Philadelphia, USA.

Joachims, T. (1998a). *Making large-scale SVM learning practical*, Technical report University of Dortmund.

Joachims, T. (1998b). *Text categorization with Support Vector Machines: learning with many relevant features*. Proceedings of the 10th European Conference on Machine Learning, Springer Lecture Notes in Computer Science, Vol. 1398 (pp. 137–142). Springer.

Landauer, T.K., Dumais, S.T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104 (2), 211–240.

Lang, K. (1995). *Newsweeder: Learning to filter netnews.* In A. Prieditis, S. Russell (Eds.), Proceedings of the 12th International Conferrence on Machine Learning (pp. 331–339). San Francisco: Morgan Kaufmann Publishers.

Leopold, E., Kindermann, J. (2002). Text categorization with Support Vector Machines. How to represent texts in input space? *Machine Learning,* 46, 423–444.

Lowe, D., Matthews, R. (1995). Shakespeare vs. Fletcher: A stylometric analysis by radial basis functions. *Computers and the Humanities,* 29, 449–461.

Manning, C.D., Schütze, H.(1999). *Foundations of statistical natural language processing*. Cambridge MA, London: MIT Press.

Mitchell, Tom (1997). *Machine Learning*. Boston et al.: McGraw-Hill.

Mladenic, D., Grobelnik M. (1999). *Feature selection for unbalanced class distribution and naive Bayes*. In I. Bratko, S. Dzeroski (Eds.), Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999) (pp. 258–267). San Francisco: Morgan Kaufmann.

Neumann, G., Schmeier, S. (2002). Shallow natural language technology and text mining. *Künstliche Intelligenz*, 2002 (2), 23–26.

Neumann, G., Piskorski, J. (2002). A Shallow text processing core engine. *Computational Intelligence*, 18 (3), 451–476.

Nigam, K., McCallum, A.K., Thrun, S., Mitchel, T. (1999). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39 (1/2), 103–134.

Paaß, G., Leopold, E., Larson, M., Kindermann, J., Eickeler, S. (2002). *SVM Classification using sequences of phonemes and syllables.* Tapio Elomaa & Heikki Mannila & Hannu Toivonen (Eds.), Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2002); August 19–23, 2002 Helsinki, Finland, Lecture Notes in Artificial Intelligence 2431 (pp. 373–384) Berlin, Heidelberg: Springer.

Porter, M.F. (1980). An algorithm for suffix stripping. *Program (Automated Library and Information Systems)*, 14 (3), 130–137.

Rudman, J. (1998). The state of authorship attribution studies: some problems and solutions. *Computers and the Humanities,* 31, 351–365.

Salton, G., McGill, M.J. 1983. *Introduction to modern information retrieval.* New York: McGraw Hill.

Shapire, R.E., Singer, Y. (2000). BoosTexter: a boosting based system for text categorization. *Machine Learning*, 39, 135–168.

Sparck-Jones, K. (1999). *Automatic summarizing: factors and directions.* In I. Mani, M.T. Maybury (Eds.), Advances in Automated Text Summarization.

Srivastava, J., Cooley, R., Deshpande, M., Tan, P.-N. (2000). Web usage mining: discovery and applications of usage patterns from web data, *SIGKDD Exploratins,* 1 (2), 12–23.

Stöber, K., Wagner, P., Helbit, J., Köster, S., Stall, D., Thomae, M., Blauert, J., Hess, W., Hoffmann, R., Mangold, H. (2000). *Speech synthesis by multilevel selection and concatenation of units from large speech Corpora.* In: W. Wahlster (Ed.), Verb-mobil. Springer, 2000.

Stricker, M., Vichot, F., Dreyfus, G., Wolinski, F. (2000). *Vers la conception de filtres d'informations efficaces*. In Reconnaissance des Formes et Intelligence Artificielle (RFIA '2000) (pp. 129–137).

Thisted, R., Efron, B. (1987). Did Shakespeare write a newly discovered poem? *Biometrika*, 74 (3), 445–55.

Thisted, R. (1988). *Elements of statistical computing.* London: Chapman&Hall.

Towsey, M., Diederich, J., Schellhammer, I., Chalup, S., Brugman, C. (1998). Natural language learning by recurrent neural networks: A comparison with probabilistic approaches. *Computational natural language learning conference*. Australian Natural Language Processing Fortnight. Sydney: Macquarie University, 15–17 Jan 1998.

Tweedie, F.J., Singh, S., Holmes, D.I. (1996). Neural network applications in stylometry: the federalist paper. *Computers and the Humanities,* 30, 1–10.

van Rijsbergen, C.J. (1979). *Information Retrieval.* London, Boston: Butterworths.

Vapnik, V.N. (1998). *Statistical Learning Theory.* New York et al.: Wiley & Sons.

Weiss, S.M., Apt, C., Damerau, F., Johnson, D.E., Oles, F.J., Goetz, T., Hampp, T. (1999). Maximizing textmining performance. *IEEE Intelligent Systems*, 14 (4), 63–69.