Text Classification Using Small Number of Features

Masoud Makrehchi and Mohamed S. Kamel

Pattern Analysis and Machine Intelligence Lab
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
{makrechi,mkamel}@pami.uwaterloo.ca

Abstract. Feature selection method for text classification based on information gain ranking, improved by removing redundant terms using mutual information measure and inclusion index, is proposed. We report an experiment to study the impact of term redundancy on the performance of text classifier. The result shows that term redundancy behaves very similar to noise and may degrade the classifier performance. The proposed method is tested on an SVM text classifier. Feature reduction by this method remarkably outperforms information gain based feature selection.

1 Introduction

Recently text classification has been one of the fast paced applications of machine learning and data mining [1]. There are many applications using text classification techniques such as natural language processing and information retrieval [2]. Since text classification is a supervised learning process, a wide range of learning methods, namely nearest neighbour, regression models, Bayesian approach, decision trees, inductive rule learning, neural networks and support vector machines have been proposed [3, 4].

Most text classification algorithms use vector space model or bag of words to represent text documents. In this model, every word or group of words, depends on working with a single word or a phrase, called a term, which represents one dimension of the feature space. A positive number is assigned to each term. This number can be the frequency of the term in the text [5].

One problem with this modelling is high dimensionality of feature space, meaning a very large vocabulary that consists of all terms occurring at least once in the collection of documents. Although high dimensional feature space has destructive influences on the performance of most text classifiers, its impact on increasing complexity is worse and expensive. Then, two main objectives of feature selection are improving both classification effectiveness and computational efficiency. [6, 7].

In aggressive feature selection, most irrelevant, non-predictive, and non-informative features are removed and classification task is performed by very

few features with minimum loss of performance and maximum reduction of complexity. In [6] the number of selected features is as low as 3% of features. More aggressive feature selection, including only 1% of all features, has been reported in [8]. Both reports are about feature selection for text classifiers. In this type of feature selection strategies, the main concern is the complexity reduction, as well as improving the classifier performance.

One well-known approach for removing a large number of non-predictive features is feature ranking [6, 8]. Being ranked by a scoring metric such as information gain, Chi-Squared or odds-ratio, all features are descendingly sorted and a very few number of best features are kept and the rest of features are removed. However, these methods have a serious disadvantage, which is ignoring the correlation between terms because most ranking measures consider the terms individually. An experiment, detailed in the next section, shows that the impact of term redundancy is as distractive as noise.

In this paper, a new approach for feature selection, with more than 98% reduction, is proposed. The method is based on a multi stage feature selection including pre-processing tasks, information gain based term ranking and removing redundant terms by a proposed method which uses mutual information measure and inclusion index. The paper consists of five sections. After the introduction, impact of redundancy on the performance of text classifier is discussed in Section 2. In Section 3, the proposed multi stage feature reduction and a method to identify and remove redundant terms are introduced. Experimental results and conclusion are presented in Sections 4 and 5, respectively.

2 Impact of Redundancy on the Performance of Text Classifiers

Redundancy is a kind of data dependency and correlation which can be estimated by different ways, such as the Jaccard distance, Cosin similarity, co-occurrence and co-location measures [9–11]. In this paper, redundancy between two terms is measured by mutual information. An experiment is set up in order to illustrate the influence of redundancy on the classifier performance. An SVM classifier with a linear kernel is employed. The data collection is the well known 20 Newsgroups data set. In this experiment, classification accuracy is used as a performance evaluation measure. We show that adding redundancy, in the case of very low number of features, can degrade the accuracy. The testing process is as follows.

Let $\mathbf{T} = \{t_1, t_2, \ldots, t_N\}$ be the vocabulary. The terms are ranked by information gain, such that t_1 is the best term and t_N is the worst one. A smaller set $\mathbf{V}$, so called the set of selected features, is defined as follows; $\mathbf{V} = \{v_1, v_2, \ldots, v_n\}, \mathbf{V} \subset \mathbf{T}, n \ll N$. Three different forms of V are generated by the following schemas;

- n best terms: The n first terms of $\mathbf{T}$ are selected such that $v_i = t_i, 1 \leq i \leq n$.
- $n/2$ best terms + $n/2$ redundant terms: In this schema, vector $\mathbf{V}$ contains two parts. First part is selected like first schema, except instead of n, $n/2$ best terms are picked up. The $n/2$ terms in the second part are artificially

Table 1. The impact of redundancy and noise on the accuracy of the SVM text classifier.

number of terms	5	10	15	20	25	30	35	40	average
n **best terms**	0.1793	0.3465	0.5843	0.6991	0.7630	0.8455	0.9299	0.9369	**0.6606**
50% **redundancy**	0.1493	0.2499	0.3473	0.4456	0.5029	0.5922	0.6646	0.6925	**0.4555**
50% **noise**	0.1485	0.2483	0.3302	0.4038	0.5024	0.5833	0.6752	0.7185	**0.4513**

generated by adding very small noise to each term of the first part. By this formulation, the rate of redundancy is at least 50%. Since we use binary features (without weights), added noise is a uniform binary noise changing the corresponding binary features from zero to one or vice versa. In order to achieve high degree of redundancy, few number of features, 2% of whole features, are chosen to be affected by noise.

- $n/2$ best terms + $n/2$ noise: It is the same as previous schema except the second part consists of noisy terms. Because of using feature ranking measures, $n/2$ last (worst) terms which can be treated as noise, are added to the first part.

All three feature vectors with different values for n, $n = \{5, 10, \ldots, 40\}$, are submitted to the SVM classifier. In order to estimate the accuracy, a five-fold cross validation schema is employed. In this process, the collection is divided into five subsets. The experiment is repeated five times. Each time we train the classifier with four subsets and leave the fifth one for test phase. The average of five accuracies is the estimated accuracy.

Table 1 illustrates the result. It clearly shows that redundancy and noise reduce the accuracy. Comparing the averages depicts both schemas have almost similar impact on the classifier. In a small ranked feature vector, the risk of having redundant term is quite high. For example in a five-term feature vector, if there is only one redundant term, we are actually using four terms instead of five because one of the terms is useless. By removing the redundant term, we make room for another term which can improve the predictive power of the feature vector.

3 Proposed Approach

The main goal of the proposed schema is providing a solution for feature selection with a high rate of reduction, by which the number of selected features $\mathbf{V}$ is much less than those in the original vocabulary $\mathbf{T}$. We propose a three-stage feature selection strategy including pre-processing tasks, information gain ranking, and removing redundant terms. The first stage involves pre-processing tasks that include Porter word stemming which can reduce almost 40% of terms, removing general stopwords reducing about 200 terms, and removing most and least frequent terms. Since we are using the 20 Newsgroups data set, the original vocabulary has about $118,275$ terms. The pre-processing tasks cut down the size of the vocabulary 75.50%. In this step, we are not losing much information because the pre-processing tasks remove non-informative, noise, stopwords, and misspelled words.

In the second stage, information gain is used to select most informative and predictive terms. Information gain is one of the most efficient measures for feature ranking in classification problems [8]. Yang and Pedersen [7] have shown that sophisticated techniques such as information gain or Chi-Squared can reduce the dimensionality of the vocabulary by a factor of 100 with no loss (or even with a small increase) of effectiveness. Here, the terms in the vocabulary after preprocessing which includes $28,983$ terms, are ranked by information gain. The 10% of best terms are chosen as most informative and predictive terms.

Information gain and other filter based feature selection methods ignore the correlation between features and evaluate them individually. The main motivation of the work reported in this paper is improving information gain ranking by identifying any correlation between terms, and extracting and removing redundancies, which is the third stage. At this level, about 5% to 20% of ranked features are selected. While employing very few features, any term redundancy influences the output of the classifier and reduces the accuracy. The proposed approach has two core elements; mutual information and inclusion index which are detailed in the following subsections.

3.1 Mutual Information

Mutual information is a measure of statistical information shared between two probability distributions. Based on the definition in [12], mutual information $I(x;y)$ is computed by the relative entropy of a joint probability distribution like $p(x,y)$ and the product of the marginal probability distributions $p(x)$ and $p(y)$

$$I(x;y) = D(p(x,y)||p(x)p(y)) = \sum_{x}\sum_{y} p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{1}$$

Mutual information has been applied in text mining and information retrieval for applications like word association [13] and feature selection [14]. Mutual information is viewed as the entropy of co-occurrence of two terms when observing a category. We practically compute mutual information between two other mutual information measures. Each measure represents shared information between a term like t_i and a class such as c_k. Since we are interested in the distribution of a pair of terms given a specific category, the joint distribution is considered as the probability of occurrence of the two terms t_i and t_j in those documents belonging to the class c_k. Eq. 1 can be rewritten as follows

$$I(t_i;c_k) = \sum_{t_i}\sum_{c_k} p(t_i,c_k) log \frac{p(t_i,c_k)}{p(t_i)p(c_k)} \tag{2}$$

where $I(t_i;c_k)$ is the mutual information of the distribution of term t_i and category c_k. Mutual information itself has been used as a ranking measure and showed very poor result [7]. Eq. 2 might be writhen for term t_j exactly the same way. In other word, $I(t_i;c_k)$ is the entropy of $p(t_i \cap c_k)$ which is the probability

distribution of the occurrence of the term t_i in the class c_k. The total mutual information (φ) is calculated as follows

$$\varphi\left\{I(t_i;c_k);I(t_j;c_k)\right\} = \varphi(t_i \cap c_k, t_j \cap c_k) \tag{3}$$

$$\varphi(t_i \cap c_k, t_j \cap c_k) = \sum_{t_i,c_k}\sum_{t_j,c_k} p(t_i \cap c_k, t_j \cap c_k) log \frac{p(t_i \cap c_k, t_j \cap c_k)}{p(t_i \cap c_k).p(t_j \cap c_k)} \tag{4}$$

$\varphi\left\{I(t_i;c_k);I(t_j;c_k)\right\}$ is a point-wise mutual information. The total mutual information of two terms when observing whole category information is the average of the mutual information over c. This measure is simply represented by the summarized form $\varphi(t_i;t_j)$.

$$\varphi(t_i;t_j) = \sum_{k=1}^{C} \varphi(t_i \cap c_k, t_j \cap c_k) \tag{5}$$

where C is the number of categories. Since φ has no upper bound, normalized mutual information Φ which has upper bound and is a good measure to compare two shared information, is proposed as follows [16].

$$\Phi(t_i;t_j) = \frac{\varphi(t_i;t_j)}{\sqrt{I(t_i;c).I(t_j;c)}}, 0 \leq \Phi(t_1;t_2) \leq 1 \tag{6}$$

From [16], φ and $I(t_i;c)$ can be estimated as following equations,

$$I(t_i;c) = \sum_{k=1}^{C} \frac{n_{t_i}^{c_k}}{n} log \frac{\frac{n_{t_i}^{c_k}}{n}}{\frac{n_{t_i}}{n}.\frac{n_{c_k}}{n}} = \frac{1}{n}\sum_{k=1}^{C} n_{t_i}^{c_k} log \frac{n.n_{t_i}^{c_k}}{n_{t_i}.n_{c_k}} \tag{7}$$

$$\varphi(t_i;t_j) = \sum_{k=1}^{C} \frac{n_{t_i,t_j}^{c_k}}{n} log \frac{\frac{n_{t_i,t_j}^{c_k}}{n}}{\frac{n_{t_i,t_j}}{n}.\frac{n_{c_k}}{n}} = \frac{1}{n}\sum_{k=1}^{C} n_{t_i,t_j}^{c_k} log \frac{n.n_{t_i,t_j}^{c_k}}{n_{t_i,t_j}.n_{c_k}} \tag{8}$$

where n is the total number of documents in the collection, n_{c_k} depicts the number of documents in k^{th} category, n_{t_i} (n_{t_i,t_j}) is the number of documents which have term t_i (both t_i and t_j). The number of documents which belongs to the k^{th} class, and includes the term t_i (t_i and t_j) is represented by $n_{t_i}^{c_k}$ ($n_{t_i,t_j}^{c_k}$). Eq. 6 is estimated as follows,

$$\Phi(t_i;t_j) = \frac{\sum_{k=1}^{C} n_{t_i,t_j}^{c_k} log \frac{n.n_{t_i,t_j}^{c_k}}{n_{t_i,t_j}.n_{c_k}}}{\sqrt{\sum_{k=1}^{C} n_{t_i}^{c_k} log \frac{n.n_{t_i}^{c_k}}{n_{t_i}.n_{c_k}}.\sum_{k=1}^{C} n_{t_j}^{c_k} log \frac{n.n_{t_j}^{c_k}}{n_{t_j}.n_{c_k}}}} \tag{9}$$

Φ is equal to one if the two terms are completely identical and correlated when observing a category, and $\Phi = 0$ if the two terms are completely uncorrelated. It should be noted that although point-wise mutual information $\varphi\left\{I(t_i;c_k);I(t_j;c_k)\right\}$ can be negative [15], the average mutual information

$\varphi(t_i; t_j)$ is always positive and its normalized version is less than or equal to one.

Φ is calculated for all possible pairs of terms in the vocabulary. The result is $\mathbf{\Phi}$ matrix in the order of $M \times M$, where M is the size of the vocabulary or the number of terms. Since Φ is a symmetric measure, and always $\Phi(t_i; t_i) = 1$, in order to construct the matrix, $\frac{M(M-1)}{2}$ number of Φ calculations are necessary, that it very expensive. One approach to overcome the problem is to calculate $\mathbf{\Phi}$ matrix for a very small subset of terms S of the vocabulary V. It means instead of the full $\mathbf{\Phi}$ matrix, a sub-matrix of $\mathbf{\Phi}$ is provided. In other words, we need to calculate $\mathbf{\Phi}$ for most likely correlated terms. Let us suppose that there are n_s groups of correlated terms in the vocabulary. The problem is identifying these groups and calculating $\mathbf{\Phi}$ for each of them. We propose inclusion index and matrix for this purpose.

3.2 Inclusion Index

Let $D = \{d_1, d_2, \ldots, d_n\}$ be the collection of documents. Every document is represented by a vector of words, called the document vector space, for example,

$$d_k = \{w_{k,1}.t_1, w_{k,2}.t_2, \ldots, w_{k,M}.t_M\} \tag{10}$$

where $w_{k,q}$ is the weight of the q^{th} term in the k^{th} document. Here we use binary weighting which depends on wether the term is in the document or not. As a consequence, D can be represented by an $N \times M$ matrix in which every row (d_k) is a document and every column (t_i) represents the occurrence of the term in every document. Based on this notation, inclusion, which is a term-term relation, is defined in [17]. Inclusion index $Inc(t_i, t_j)$, representing how much t_j includes t_i, is calculated by,

$$Inc(t_i, t_j) = \frac{||t_i \cap t_j||}{||t_i||}, \quad Inc(t_i, t_j) \neq Inc(t_j, t_i) \tag{11}$$

where $||.||$ is the cardinal number of the set. $Inc(t_i, t_j) = 1$ when t_j is completely covering t_i and called full inclusive. $Inc(t_i, t_j) = 0$ means there is no overlap between the two terms. There is also partial inclusion when $0 < Inc(t_i, t_j) < 1$. t_j is called more inclusive than t_i if $Inc(t_i, t_j) > Inc(t_j, t_i)$. The inclusion matrix **Inc** is an $M \times M$ matrix in which each entry is an inclusion index between two terms.

3.3 Redundancy Removal Algorithm

The main idea in identifying redundant terms is finding the sets of correlated terms. For example, {rec,hockei,motorcycl,bike,nhl,playoff} shows one of these sets including six correlated terms. The sets are extracted using inclusion matrix **Inc**. Algorithm 1 represents the detail of extracting the sets and then identifying redundant terms.

Algorithm 1 Extracting redundant terms.

for $1 \leq i, j \leq M$ if $Inc(i,j) > threshold \Rightarrow inc(i,j) \leftarrow 1$ else $inc(i,j) \leftarrow 0$
for $1 \leq i \leq M$ $TermIndex(i) \leftarrow 0$
$i \leftarrow 1$
while *"the set of zero element in TermIndex is not empty"*
 $k \leftarrow$ *index of* 1^{st} *zero element in TermIndex*
 $TermIndex(k) \leftarrow 1$, $u \leftarrow k$, $l \leftarrow 1$
 while *"l is non-zero"*
  $z \leftarrow$ *the set of non-zero elements of* k^{th} *column of* **inc**
  if *"z is non-empty"* $\Rightarrow$ *append z to u, sort u*
  $TermIndex(k) \leftarrow 1$
  $x \leftarrow$ *number of zero elements of TermIndex according to u*
  $l \leftarrow$ *number of elements in x*
  if $l > 0 \Rightarrow k \leftarrow u(x(1))$
 end while
 $CorrrelatedTermSet(i) \leftarrow u$
 $i \leftarrow i + 1$
end while
remove all sets from CorrelatedTermSet which have less than two elements
for $q = 1$ *to number of set of correlated terms*
 calculate Φ_q, calculate $\mathbf{Inc_q}$
 for $i = 1$ to *number of elements in* q^th *set of correlated terms*
  for $j = 1$ to *number of elements in* q^th *set of correlated terms*
   $R_q(i,j) \leftarrow Inc_q(i,j).\Phi_q(i,j)$
   if $i = j \Rightarrow R_q(i,j) \leftarrow 0$
  end for
 end for
 keep maximum element of each row of $\mathbf{R_q}$ *and make other else zero*
 $RedundantTerms \leftarrow$ *terms according to the whole zero columns of* $\mathbf{R_q}$
end for

Let S_q be the q^{th} set of correlated terms. Instead of calculating full matrix of $\mathbf{\Phi}$, it is only obtained for the terms in the S_q. The resulting matrix is represented by $\mathbf{\Phi_q}$. We do the same for $\mathbf{Inc_q}$. Matrix $\mathbf{R_q}$, which is called redundancy matrix, is calculated by entry-entry multiplication of $\mathbf{\Phi_q}$ and $\mathbf{Inc_q}$ as follows

$$R_q(i,j) = \Phi_q(i,j).Inc_q(i,j), \quad 1 \leq i, j \leq n_q \tag{12}$$

where n_q is the number of terms in S_q. The i^{th} row of $\mathbf{R_q}$, which is an $n_q \times n_q$ matrix, shows that the i^{th} term (in S_q) in which terms is included or with which ones are being covered. In each row the maximum entry is kept and the others are set to zero. Finally, every term that its corresponding column in $\mathbf{R_q}$ is full zero (all elements are zero), is assigned as a redundant term because it does not include any other term. Table 2 shows the resulting matrices for a set of correlated terms.

4 Experimental Results

The proposed approach has been applied on 20 Newsgroups data set using an SVM classifier with linear kernel. Although there are some reports showing feature selection for SVM classifier not only is unnecessary but also can reduce its performance [6,18], in addition to [8], in this paper we show that for a very small size of feature vector, SVM performance can be improved by feature selection through redundancy reduction.

Table 2. An example of extracting redundant terms from q^{th} set of correlated terms, (A) normalized mutual information matrix $\mathbf{\Phi_q}$, (B) inclusion sub-matrix $\mathbf{Inc_q}$, (C) multiplication of the two matrices ($\mathbf{\Phi_q}$ and $\mathbf{Inc_q}$), (D) term redundancy matrix $\mathbf{R_q}$. Based on $\mathbf{R_q}$, all terms, whose corresponding columns are zero, are redundant and should be removed.

(A)

	rec	**hockei**	**motorcycl**	**bike**	**nhl**	**playoff**
rec	1	0.4448	0.4415	0.2866	0.2078	0.2059
hockei	0.4448	1	0	0	0.4555	0.4300
motorcycl	0.4415	0	1	0.5886	0	0
bike	0.2866	0	0.5886	1	0	0
nhl	0.2078	0.4555	0	0	1	0.1754
playoff	0.2059	0.4300	0	0	0.1754	1

(B)

	rec	**hockei**	**motorcycl**	**bike**	**nhl**	**playoff**
rec	1	0.2221	0.2255	0.1162	0.0669	0.0680
hockei	0.9951	1	0	0	0.2998	0.2883
motorcycl	0.9903	0	1	0.4911	0	0
bike	0.9906	0	0.9530	1	0	0
nhl	0.9945	0.9945	0	0	1	0.2623
playoff	1	0.9459	0	0	0.2595	1

(C)

	rec	**hockei**	**motorcycl**	**bike**	**nhl**	**playoff**
rec	0	0.0988	0.0995	0.0333	0.0139	0.0140
hockei	0.4426	0	0	0	0.1366	0.1240
motorcycl	0.4372	0	0	0.2891	0	0
bike	0.2839	0	0.5609	0	0	0
nhl	0.2067	0.4530	0	0	0	0.0460
playoff	0.2059	0.4067	0	0	0.0455	0

(D)

	rec	**hockei**	**motorcycl**	**bike**	**nhl**	**playoff**
rec	0	0	0.0995	0	0	0
hockei	0.4426	0	0	0	0	0
motorcycl	0.4372	0	0	0	0	0
bike	0	0	0.5609	0	0	0
nhl	0	0.4530	0	0	0	0
playoff	0	0.4067	0	0	0	0

The proposed schema has been evaluated by comparing its results with those of stand-alone information gain ranking. A five-fold cross validation is used for better estimation of classifier performance. In addition to classifier accuracy, two more performance indices have been used, including micro-average, and macro-average. They are calculated based on a_j, the number of samples which are correctly classified as class j, and b_j, the number of samples wrongly classified as class j.

$$macro - average = \frac{\sum_{i=1}^{C} \frac{a_i}{a_i + b_i}}{C}, \quad micro - average = \frac{\sum_{i=1}^{C} a_i}{(\sum_{i=1}^{C} a_i) + (\sum_{i=1}^{C} b_i)} \tag{13}$$

where C is the number of categories. Table 3 presents the results of two methods with different performance measures. Each method has been applied on the SVM classifier with eight levels of aggressive feature selections. In all measures, and most feature selection levels, the proposed method has outperformed information gain ranking. The last column of the table depicts the averages which clearly show that the proposed approach is more efficient.

Table 3. Comparing the results of aggressive feature selection using information gain ranking and the proposed method (bold) for the SVM text classifier.

number of terms	5	10	15	20	25	30	35	40	average
accuracy	0.1793	0.3465	0.5843	0.6991	0.7630	0.8455	0.9299	0.9369	0.6606
	0.2028	**0.4854**	**0.7098**	**0.8032**	**0.9031**	**0.9036**	**0.9027**	**0.9022**	**0.7266**
micro-average	0.1113	0.3334	0.5567	0.6799	0.7391	0.8481	0.9331	0.9403	0.6427
	0.1601	**0.4645**	**0.6860**	**0.7690**	**0.8844**	**0.8988**	**0.8871**	**0.8845**	**0.7043**
macro-average	0.1004	0.2932	0.5065	0.6470	0.7185	0.8196	0.9300	0.9370	0.6190
	0.1260	**0.4120**	**0.6610**	**0.7640**	**0.8826**	**0.8842**	**0.8822**	**0.8827**	**0.6868**

5 Conclusion

Aggressive feature selection, with higher than 95% feature reduction, was discussed. This sort of feature selections is very applicable to text classifiers while because of dealing with huge size of feature space so called vocabulary. Text classifiers, working with very small feature vectors, are very sensitive to noise, outliers and redundancies. Then, improving any classical feature selection method like feature ranking for aggressive reduction is strongly necessary.

Term redundancy in text classifiers causes a serious drawback in most feature rankings, such as information gain, because they always ignore correlation between terms. The result of an experiment in the paper showed that the effect of term redundancy can be worse than noise. To find and reduce term redundancy, a method was proposed for improving aggressive feature selection by information gain ranking. The method was based on identifying and removing term redundancy using mutual information measure and inclusion index. Terms were grouped in a few sets of correlated terms using inclusion matrix. In the next step each set was modelled by the term redundancy matrix.

Aggressive feature selection approaches by stand-alone information gain ranking and proposed method (removing the redundant term from ranked feature vector by information gain) were compared in an SVM text classifier framework. Results showed that with three evaluation measures, the proposed schema outperformed the aggressive feature selection by the stand-alone information gain. The proposed method improved information gain 10% in accuracy, 9.5% in macro-average, 11% in micro-average. Better results are expected for other feature ranking methods such as Chi-Squared and odds-ratio.

Acknowledgement

This research was supported in part by the National Science and Engineering Research Council of Canada.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47

2. Lam, W., Ruiz, M.E., Srinivasan, P.: Automatic text categorization and its applications to text retrieval. IEEE Transactions on Knowledge and Data Engineering **11** (1999) 865–879
3. Berry, M.W.: Survey of Text Mining: Clustering, Classification, and Retrieval. Springer (2004)
4. Yiming, Liu, X.: A re-examination of text categorization methods. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. (1999) 42–49
5. Wong, S.K.M., Raghavan, V.V.: Vector space model of information retrieval: a reevaluation. In: Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval. (1984) 167–185
6. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: Proceedings of the eleventh international conference on Information and knowledge management. (2002) 659–661
7. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In Fisher, D.H., ed.: Proceedings of ICML-97, 14th International Conference on Machine Learning, Nashville, US (1997) 412–420
8. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: Using aggressive feature selection to make svms competitive with c4.5. In: Proceedings of ICML-04, Twenty-first international conference on Machine learning, Banff, Alberta, Canada (2004) 321–328
9. Dagan, I., Lee, L., Pereira, F.C.N.: Similarity-based models of word cooccurrence probabilities. Machine Learning **34** (1999) 43–69
10. Xu, J., Croft, W.B.: Corpus-based stemming using cooccurrence of word variants. ACM Trans. Inf. Syst. **16** (1998) 61–81
11. Soucy, P., Mineau, G.W.: A simple feature selection method for text classification. In Nebel, B., ed.: Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence, Seattle, US (2001) 897–902
12. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience (1991)
13. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. Comput. Linguist. **16** (1990) 22–29
14. Wang, G., Lochovsky, F.H.: Feature selection with conditional mutual information maximin in text categorization. In: CIKM '04: Proceedings of the Thirteenth ACM conference on Information and knowledge management. (2004) 342–349
15. Mackay, D.: Information Theory, Inference and Learning Algorithms. Cambridge University (2003)
16. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining partitionings. In: Proceedings of AAAI 2002, Edmonton, Canada, AAAI (2002) 93–98
17. Salton, G.: Recent trends in automatic information retrieval. In: SIGIR '86: Proceedings of the 9th annual international ACM SIGIR conference on Research and development in information retrieval. (1986) 1–10
18. Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In Nédellec, C., Rouveirol, C., eds.: Proceedings of ECML-98, 10th European Conference on Machine Learning. Number 1398, Chemnitz, DE, Springer Verlag, Heidelberg, DE (1998) 137–142