

Using Restrictive Classification and Meta Classification for Junk Elimination

Stefan Siersdorfer, Gerhard Weikum
{stesi,weikum}@mpi-sb.mpg.de

Max-Planck-Institute for Computer Science, Germany

Abstract. This paper addresses the problem of performing supervised classification on document collections containing also *junk documents*. With "junk documents" we mean documents that do *not* belong to the topic categories (classes) we are interested in. This type of documents can typically not be covered by the training set; nevertheless in many real world applications (e.g. classification of web or intranet content, focused crawling etc.) such documents occur quite often and a classifier has to make a decision about them. We tackle this problem by using restrictive methods and ensemble-based meta methods that may decide to leave out some documents rather than assigning them to inappropriate classes with low confidence. Our experiments with four different data sets show that the proposed techniques can eliminate a relatively large fraction of junk documents while dismissing only a significantly smaller fraction of potentially interesting documents.

1 Introduction

1.1 Motivation

Automatic document classification is useful for a wide range of applications such as organizing web, intranet, or portal pages into topic directories, filtering news feeds or mail, focused crawling on the web or in intranets, and many more. In the classical scenario it is often assumed that all topic categories (classes) are known and that the training corpus provides example documents for all these categories. However in many real world applications these assumptions do not hold. As an example consider a focused crawler where we are interested just in a limited number of topics and, as the case may be, subtopics. Here we have to deal with the problem that the web covers such a plethora (and growing number) of other topics that it is impossible to build a training set that comprises all these topics. However a focused crawler will very likely see such "junk documents", although the underlying classifier has never seen (and never had a chance to see) any training data for the "junk" class, and will have to make a decision about them. It is not clear how a classifier trained to discriminate topics based on training data about "computer science", "mathematics" and "physics" will behave on documents about, say, "esoterism"; there is a significant difference between negative examples and "junk" documents.

In this paper we propose restrictive classification methods to tackle the "junk problem". In restrictive classification, we consider classifiers for a given topic that make a ternary decision on a newly seen document: they can accept the document for the topic, reject it for the topic, or abstain if there is neither sufficiently evidence for acceptance nor for rejection. With the abstention option we aim to achieve a lower error on the remaining documents and to eliminate the junk documents that would be spuriously assigned to one of the classes of interest.

1.2 Contribution

In [17] a framework for restrictive classification and meta methods with ternary decisions was introduced. It was assumed that all underlying classifiers had sufficient training data: both positive and negative samples of every thematic that might occur among the test documents. In the current paper we drop this assumption and make a major step forward to cope with corpora that are not necessarily "in tune" with the thematic classes that were defined a priori. This is a very significant case with "open" corpora like the web with a huge amount of topics and documents for which comprehensive training is absolutely impossible.

The current paper makes the following technical contributions:

1. It develops decision procedures for junk elimination based on restrictive classifiers and meta classifiers
2. It develops a probabilistic explanation model and analytically shows that the elimination ratio of junk documents is larger than the loss of potentially interesting documents
3. It presents comprehensive experiments, using four different data sets, including a web document collection, that demonstrate the benefits of the proposed methods.

1.3 Related Work

There is a considerable prior of work on text document classification using all kinds of probabilistic and discriminative models [6]. The machine learning literature has studied a variety of meta methods such as bagging, stacking, or boosting [4, 20, 12, 9], and also combinations of heterogeneous learners (e.g., [22]).

In [17] restrictive meta methods based on training set splitting and a probabilistic model are developed for automatic handling of the tradeoffs between different aspects of the classifier quality (loss, classification error, and efficiency). In [16] a similar probabilistic model is applied to meta clustering.

However, to our knowledge, these techniques were, up to now, not considered in the context of junk reduction.

1.4 Outline

The rest of the paper is organized as follows. In Section 2 we briefly review the technical basics of classification methods. Section 3 presents our notion of

restrictive methods: we describe simple restrictive methods and the restrictive combination of different classification methods, and we provide a probabilistic model for junk reduction. Section 4 provides experiments on different real-world data sets.

2 Technical Basics

Classifying text documents into thematic categories usually follows a supervised learning paradigm and is based on training documents that need to be provided for each topic. Moreover, the best classification methods, most notably, SVMs, need both positive and negative samples for training. This prerequisite is not satisfied in the presence of “junk documents” for which no training samples are available.

Both training documents and test documents, which are later given to the classifier, are represented as multidimensional feature vectors. In the prevalent bag-of-words model the features are derived from word occurrence frequencies, e.g. based on $tf \cdot idf$ feature weights [3, 13]. Often feature selection algorithms are applied to reduce the dimensionality of the feature space and eliminate “noisy”, non-characteristic features, based on information-theoretic measures for feature ordering (e.g., relative entropy or information gain).

Feature vectors of topic labeled text documents (e.g., capturing $tf \cdot idf$ weights of terms) are used to train a classification model for each topic, using probabilistic (e.g., Naive Bayes) or discriminative models (e.g., SVM). Linear support vector machines (SVMs) construct a hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ that separates the set of positive training examples from a set of negative examples with maximum margin. This training requires solving a quadratic optimization problem whose empirical performance is somewhere between quadratic and cubic in the number of training documents [5]. For a new, previously unseen, document \mathbf{d} the SVM merely needs to test whether the document lies on the “positive” side or the “negative” side of the separating hyperplane. The decision simply requires computing a scalar product of the vectors \mathbf{w} and \mathbf{d} . SVMs have been shown to perform very well for text classification (see, e.g., [7, 10]).

Multiple classifiers can be combined using a meta classifier approach [4, 20, 9], for example, by voting on the final decision (including weighted voting, see, e.g., [19]). Such a setup is interesting not only to combine different algorithmic techniques, but mostly for combining classifiers that have been trained with different training sets or for different feature spaces.

In this paper we consider only binary classifiers that make a decision for a single topic, based on positive and negative training examples.

3 Elimination of Junk by Restrictive Classification

3.1 Tradeoffs in restrictive classification

In this paragraph we describe the tradeoffs that occur in restrictive classification. Consider a training set T consisting of documents from two classes pos and neg ,

and a set of unlabeled documents U containing documents from pos and neg and $junk$ documents, that are not in these classes. (The scenario can be easily generalized to a set of l classes $C = \{c_1, \dots, c_l\}$ instead of two classes.) Given a document $d \in U$, a restrictive classifier gives us the result $+1$ if it classifies the document into pos , -1 if it classifies the document into neg , 0 if the classifier abstains. The possible combinations between the real classes and the result of a classifier are shown in the contingency table in figure 1. In this notation e.g. $N+$ is the set of documents in neg which are assigned to class pos by the classifier, $J0$ is the set of junk documents from U where the classifier abstains, etc.

		classification		
		+	-	0
real class	pos	P+	P-	P0
	neg	N+	N-	N0
	junk	J+	J-	J0

Fig. 1. Contingency Table for Restrictive Classification with Junk Reduction

An appropriate restrictive classifier should optimize the following quality measures:

1. Maximize *junk – reduction* (fraction of junk documents dismissed by the classifier):

$$junkRed := \frac{|J0|}{|J+| + |J-| + |J0|} \quad (1)$$

2. Minimize *loss* (fraction of dismissed documents from the classes of interest pos and neg):

$$loss := \frac{|P0| + |N0|}{|P+| + |P-| + |N+| + |N-| + |P0| + |N0|} \quad (2)$$

3. Minimize *error* (fraction of non-dismissed documents classified into the wrong class):

$$error := \frac{|P-| + |N+| + |J+| + |J-|}{|P+| + |P-| + |N+| + |N-| + |J+| + |J-|} \quad (3)$$

As document reduction (not to confuse with the loss), we define the fraction of documents in U , where the classifier abstains:

$$docRed := \frac{|P0| + |N0| + |J0|}{|U|} \quad (4)$$

The document reduction can be observed directly from the classifier output without knowing the real class labels of the documents in U . The document reduction has an implicit influence on *junkRed*, *loss* and *error*.

In practice we observe a tradeoff between the loss at the one hand and junk-decimation and error on the other hand.

3.2 Making Simple methods restrictive

We can use confidence measures to make simple methods restrictive. For SVMs or the Centroid method a natural confidence measure is the distance of a test document vector from the separating hyperplane. So we can tune these methods by requiring that accepted or rejected documents have a distance above some threshold, and abstain otherwise. The threshold is our tuning parameter.

Given a document reduction of R percent, we can make a classifier restrictive by dismissing the R percent of the test documents with the lowest confidence values.

3.3 Restrictive Meta Methods

For meta classification we are given a set $V = \{v_1, \dots, v_k\}$ of k binary classifiers with results $R(v_i, d)$ in $\{+1, -1, 0\}$ for a document d , namely, $+1$ if d is accepted for the given topic by v_i , -1 if d is rejected, and 0 if v_i abstains. We can combine these results into a meta result: $Meta(d) = Meta(R(v_1, d), \dots, R(v_k, d))$ in $\{+1, -1, 0\}$ where 0 means abstention. A family of such meta methods is the linear classifier combination with thresholding [15]. Given thresholds t_1 and t_2 , with $t_1 > t_2$, and weights $w(v_i)$ for the k underlying classifiers we compute $Meta(d)$ as follows:

$$Meta(d) = \begin{cases} +1 & \text{if } \sum_{i=1}^n R(v_i, d) \cdot w(v_i) > t_1 \\ -1 & \text{if } \sum_{i=1}^n R(v_i, d) \cdot w(v_i) < t_2 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

This meta classifier family has some important special cases, depending on the choice of the weights and thresholds:

- 1) voting [4]: Meta returns the result of the majority of the classifiers.
- 2) unanimous decision: if all classifiers give us the same result (either $+1$ or -1), Meta returns this result, 0 otherwise.
- 3) weighted averaging [19]: Meta weighs the classifiers by using some predetermined quality estimator, e.g., a leave-one-out estimator for each v_i .

The restrictive and tunable behavior is achieved by the choice of the thresholds: we dismiss the documents where the linear result combination lies between t_1 and t_2 . In the rest of the paper we will consider only the unanimous-decision meta classifier as the simplest and most conservative of the above cases in order to demonstrate the feasibility of our approach. Of course other meta options might be worthwhile. The tuning of the thresholds t_1 and t_2 is beyond the scope of this paper.

3.4 A Probabilistic Model for Restrictive Meta Methods

In this subsection we develop a simplified probabilistic model to a better understanding of why meta classification works and provide approximations for *loss*, *error* and *junkRed*. Consider the unanimous-decision meta method described above.

We associate a Bernoulli random variable X_i with each classification method v_i , where $X_i = 1$ if v_i classifies a document into class *pos* and $X_i = 0$ if v_i classifies a document into class *neg*. We want to compute the probability $P(X_1 = \dots = X_k | \text{junk})$ that the classifiers v_i provide a unanimous decision if they are presented a junk document. From basic probability theory it follows that

$$\begin{aligned} & P(X_1 = 1 \wedge X_2 = 1 | \text{Junk}) \\ &= \text{cov}(X_1, X_2 | \text{Junk}) + P(X_1 = 1 | \text{Junk}) \cdot P(X_2 = 1 | \text{Junk}) \end{aligned} \quad (6)$$

Where

$$\text{cov}(X_1, X_2 | \text{Junk}) = \frac{1}{n-1} \sum_j (x_1 - \bar{x}_1)(x_2 - \bar{x}_2) \quad (7)$$

is the covariance for the data points (x_1, x_2) of the joint distribution of (X_1, X_2) on the set of junk documents.

To model the case of $l > 2$ classification methods we use a tree dependence model, which is a well known approximation method in probabilistic IR ([14]). We define a *Dependence Graph* $G = (V, E)$ where V consists of the Bernoulli variables X_i and which contains for all X_i, X_j ($i \neq j$) an undirected edge $e(X_i, X_j)$ with weight $w(e(X_i, X_j)) = \text{cov}(X_i, X_j)$. We approximate the Dependence Graph by a maximum spanning tree $G' = (V, E')$ which maximizes the sum of the edge weights. The nodes in G' with no edges in between are considered as independent. So we obtain:

$$\begin{aligned} & P(X_1 = x_1, \dots, X_k = x_k | \text{Junk}) = \\ & P(X_{\text{root}} = 1 | \text{Junk}) \prod_{(i,j) \in E'} \frac{P(X_i = x_i, X_j = x_j | \text{Junk})}{P(X_i = x_j | \text{Junk})} \end{aligned} \quad (8)$$

where X_{root} is the root node of the tree G' and $x_i \in \{0, 1\}$. Now we introduce the following special case: For any two classification methods v_i, v_j the covariance has approximately the same value *cov*. With $w(e(X_i, X_j)) = \text{cov}$ we can (without loss of generality) choose X_1 as the root node and the edges (X_i, X_{i+1}) as tree edges.

Now we have:

$$\begin{aligned} & P(X_1 = 1, \dots, X_k = 1 | \text{Junk}) = \\ & P(X_1 = 1 | \text{Junk}) \prod_{i=1}^{k-1} P(X_{i+1} = 1 | X_i = 1 | \text{Junk}) = \\ & P(X_1 = 1 | \text{Junk}) \prod_{i=1}^{k-1} \frac{P(X_i = 1, X_{i+1} = 1 | \text{Junk})}{P(X_i = 1 | \text{Junk})} \end{aligned} \quad (9)$$

By considering equation 6 and the above assumption about the covariance we obtain

$$P(X_1 = 1, \dots, X_k = 1|Junk) = P(X_1 = 1|Junk) \prod_{i=1}^{k-1} \frac{P(X_i = 1|Junk)P(X_{i+1} = 1|Junk) + cov}{P(X_i = 1|Junk)} \quad (10)$$

Analogously we obtain $P(X_1 = 0 \wedge \dots \wedge X_k = 0|Junk)$.

If we assume that for *junk* documents the classes *pos* and *neg* are equally likely, we can substitute in the above formulas:

$$P(X_i = 1|Junk) = P(X_i = 0|Junk) = \frac{1}{2} \quad (11)$$

For the junk reduction we substitute the above formulas into:

$$junkRed = 1 - P(X_1 = \dots = X_k|Junk) = 1 - (P(X_1 = 0 \wedge \dots \wedge X_k = 0|Junk) + P(X_1 = 1 \wedge \dots \wedge X_k = 1|Junk)) \quad (12)$$

To compute the probabilities that all classifiers v_i classify a document into the same class, if the document belongs to one of the classes in $C = \{pos, neg\}$, we associate a Bernoulli variable X'_i with each classification method v_i , where $X'_i = 1$ if v_i classifies a document correctly, 0 otherwise. We want to compute the probabilities $P(X'_1 = 1 \wedge \dots \wedge X'_k = 1|C)$ and $P(X'_1 = 0 \wedge \dots \wedge X'_k = 0|C)$ that all classifiers classify a document correctly / incorrectly if the document belongs to one of the classes in C .

With analog arguments as above we obtain the following approximation:

$$P(X'_1 = 1, \dots, X'_k = 1|C) = P(X'_1 = 1|C) \prod_{i=1}^{k-1} \frac{P(X'_i = 1|C)P(X'_{i+1} = 1|C) + cov'}{P(X'_i = 1|C)} \quad (13)$$

where cov' is the covariance on the documents in C . Analogously we obtain $P(X'_1 = 0, \dots, X'_k = 0|C)$.

Let $P(C)$ be the probability that a document belongs to a class in C and $P(Junk)$ be the probability that a document is a junk document. Then we obtain approximations for *docRed*, *loss* and *error* by inserting the above expressions into:

$$loss = 1 - (P(X'_1 = 1, \dots, X'_k = 1|C) + P(X'_1 = 0, \dots, X'_k = 0|C)) \quad (14)$$

$$\frac{P(C)P(X'_1 = 0, \dots, X'_k = 0|C) + P(Junk)P(X_1 = \dots = X_k|Junk)}{1 - junkRed \cdot P(Junk) - loss \cdot P(C)} \quad (15)$$

$$docRed = junkRed \cdot P(Junk) + loss \cdot P(C) \quad (16)$$

As an illustrative example we consider the case that the $k > 2$ classification methods have the same probability $p < 0.5$ (i.e. the classification methods perform better than random) to misassign a document from C , that in the case of a junk document the assignment of the classes *pos* or *neg* are equally likely and that we have in all cases a covariance $c < p(1 - p)$ (i.e. the classification methods are not perfectly correlated.) and that our document corpus contains 50 percent junk documents. In this case we would obtain for *junkRed*, *loss* and *error*:

$$junkRed = 1 - \left(\frac{c + 1/4}{1/2} \right)^{k-1} \quad (17)$$

$$loss = 1 - \left((1 - p) \left(\frac{c + (1 - p)^2}{1 - p} \right)^{k-1} + p \left(\frac{c + p^2}{p} \right)^{k-1} \right) \quad (18)$$

$$error = \frac{p \left(\frac{c + p^2}{p} \right)^{k-1} + \left(\frac{c + 1/4}{1/2} \right)^{k-1}}{\left(\frac{c + 1/4}{1/2} \right)^{k-1} + (1 - p) \left(\frac{c + (1 - p)^2}{1 - p} \right)^{k-1} + p \left(\frac{c + p^2}{p} \right)^{k-1}} \quad (19)$$

It is easy to show that for $k \rightarrow \infty$ the loss converges monotonically to 1, and the error to 0 (i.e. with more classification methods we can obtain a lower error but pay the price of a higher loss). Furthermore also *junkRed* converges to 1 and the salient invariant $loss > junkRed$ holds. Even $\frac{1 - loss}{1 - junkRed}$ converges to ∞ ; this means, that with increasing k we dismiss much more junk documents than documents of interest. The covariance plays the role of a “smoothing constant”: with higher correlated classification methods the convergence of both loss and error is slowed down.

4 Experiments

4.1 Setup

We performed a series of experiments with real-life data from

1. Newsgroups collection at [1]. This collection contains 17847 postings collected from 20 Usenet newsgroups. Particular topics ('rec.autos', 'sci.space', etc.) contain between 600 and 1000 documents.
2. The Reuters articles [11]. This is the most widely used test collection for text categorization research. The collection contains 21578 Reuters newswire stories, subdivided into multiple categories ('earn', 'grain', 'trade', etc.).
3. The Internet Movie Database (imdb) at [2]. Documents of this collection are short and impressive movie descriptions that include the storyboard, cast overview, and user comments. This collection contains 20 topics according to particular movie genres ('drama', 'horror' etc.).

For each data set we identified all topics with sufficiently many documents. These were 20 topics for newsgroups, 7 for reuters and 9 from imdb. Among these topics we randomly chose 100 topic pairs from newsgroups, 30 from imdb and 20 from reuters. For each topic pair we choose randomly 25,50 or 100 training documents per class and kept 500 documents per class for newsgroups, 200 documents per class for imdb and 400 documents per class from reuters (distinct from the training set and also randomly chosen) for the validation of the classifiers for each pair. Additionally we "spoiled" the validation set for each pair by increasing this set by 50 percent by adding randomly chosen "junk documents" from different topics. Finally, we computed macro-averaged results for these topic pairs.

4.2 Results

In all discussed experiments, the standard bag-of-words model (using term frequencies to build L1-normalized feature vectors) with different feature selection methods was used for document representation and we used SVM as learning algorithm

In our experiments we considered the following base methods:

- *base1*: Feature selection by Mutual Information (top 200 terms); learning by SVM
- *base2*: Feature selection by Information Gain(top 200 terms); learning by SVM
- *base3*: Feature selection by Chi Squared Statistics (top 200 terms); learning by SVM

There are many alternative ways to build the base classifiers, e.g. using Naive Bayes, Decision Trees, etc. Here we chose SVM because it has been shown to often outperform other methods in text classification tasks - see e.g. [8]. Furthermore it has been shown that the above feature selection methods are highly correlated [21]. We accepted this here because we wanted to obtain base classifiers with comparable performance. To find the optimal number of features for the different feature selection methods is beyond the scope of this paper.

In the first experimental serial we compared the meta results with the results of the underlying base methods and the restrictive base methods (inducing the same document reduction as the meta method). (Figures 2 and 3)

In the second experimental serial we compared each base method for different degrees of restrictivity ¹ (inducing different document reductions). (Figure 4).

The main observations are:

- The average error of the meta method was for all experiments lower or at least equal to the error of the *best* underlying base method.

¹ We randomly chose the training and test documents once more for these experiments, causing minimal differences in the results for *docRed* = 0 compared to the base methods of the first experimental serial.

- The junk reduction is (for restrictive base methods as well as for meta methods) always significantly higher than the loss (i.e. we dismiss a higher percentage of junk than of documents of interest).
- For the imdb data set the ratio $junkRed : loss$ is best for the best base method, for the reuters and newsgroups data sets this ratio is best for the meta method.
- We can clearly observe the tradeoffs between $loss$ on the one hand and $error$ and $junkRed$ on the other hand described and analyzed in chapter 3.

As an application example we tested junk reduction for a web crawl. We obtained our training set from a bookmark file containing 79 documents of the categories "Movies" and "Computer Science" and started the crawl on the portals shown in figure 5. By this crawl we obtained an overall number of 1061 documents consisting of 400 documents about computer science, 348 about movies, and 313 junk documents. We evaluated the techniques described above on this data set. The results are shown in figures 2 through 4 (data set "web"). As in the previous experiments, the junk reduction was much higher than the loss for all restrictive methods. In terms of loss, error and junkReduction the meta method performed better than two of the 3 underlying base methods but the best restrictive base method outperformed the meta method in this experiment.

4.3 Discussion

The experiments show that all restrictive methods (i.e. meta methods as well as restrictive base methods) dismiss a significantly higher percentage of junk than of documents of interest, and additionally decrease the classification error on all data sets.

Comparing meta classifiers and restrictive base classifiers there is no clear winner: For the imdb and web data, the best base classifier outperformed the meta classifier; for newsgroups and reuters, the meta classifier outperformed the base classifiers.

5 Conclusion and Future Work

In this paper we have shown, by a probabilistic model as well as by experiments on various data sets, that restrictive classification methods can be used to eliminate junk documents. Theory and experiments show that the junk reduction is significantly higher than the loss, and the classification error is decreased. This holds for restrictive base methods as well as meta methods.

Possible topics for future work are:

- The improvement of restrictive meta methods and their parameter tuning.
- A theory that enables us to build a probabilistic model for restrictive base models (and not just for restrictive meta methods).
- Application of clustering methods to a data set, adjusted by our junk elimination method, to identify subtopics among the topics of interest.

- Application of clustering methods to the dismissed documents to identify distinct subtopics among the junk documents. This could be used to find new topics of interest or to build refined junk filters.
- Application of semisupervised learning to train on junk documents and to further improve classification quality in terms of error, loss and junk reduction, e.g., by first applying our junk reduction methods as an initial step and then performing some iterative, EM(Expectation Maximization)-like algorithm.

The work presented here is embedded in the BINGO! project [18], a toolsuite for building information portals and specialized search engines. Our long-term objective is to better understand the engineering of how to incorporate, adapt, and tune machine learning methods into more intelligent next-generation systems for information organization and search.

# TrainDocs	Meta		restrictive Base			Base			Dataset
	avg(docRed)	avg(error)	base1 avg(error)	base2 avg(error)	base3 avg(error)	base1 avg(error)	base2 avg(error)	base3 avg(error)	
25	0.159	0.489	0.493	0.489	0.489	0.52	0.515	0.518	IMDB
50	0.208	0.457	0.463	0.457	0.457	0.506	0.499	0.499	
100	0.188	0.432	0.439	0.433	0.433	0.483	0.475	0.479	
25	0.165	0.344	0.358	0.358	0.358	0.419	0.416	0.417	Newsg.
50	0.166	0.316	0.327	0.328	0.329	0.398	0.396	0.397	
100	0.143	0.31	0.318	0.315	0.315	0.385	0.381	0.381	
25	0.099	0.326	0.335	0.334	0.331	0.378	0.374	0.375	Reuters
50	0.086	0.318	0.323	0.319	0.318	0.366	0.362	0.362	
100	0.078	0.314	0.321	0.316	0.315	0.360	0.357	0.356	
79	0.074	0.301	0.282	0.319	0.327	0.323	0.348	0.351	Web

Fig. 2. Error of Meta Classification on Reuters, Newsgroups and IMDB

# TrainDocs	Meta			restrictive Base						Dataset
	avg(docRed)	avg(loss)	avg(jRed)	base1 avg(loss)	base2 avg(loss)	base3 avg(loss)	base1 avg(jRed)	base2 avg(jRed)	base3 avg(jRed)	
25	0.159	0.147	0.183	0.149	0.146	0.148	0.181	0.186	0.182	IMDB
50	0.208	0.192	0.239	0.188	0.186	0.187	0.246	0.252	0.248	
100	0.188	0.167	0.231	0.165	0.162	0.163	0.234	0.24	0.238	
25	0.165	0.109	0.276	0.118	0.122	0.12	0.259	0.251	0.254	Newsg.
50	0.166	0.098	0.301	0.103	0.108	0.108	0.29	0.281	0.282	
100	0.143	0.077	0.275	0.078	0.079	0.078	0.272	0.271	0.273	
25	0.099	0.047	0.202	0.055	0.057	0.055	0.186	0.184	0.187	Reuters
50	0.086	0.034	0.188	0.038	0.037	0.037	0.181	0.182	0.184	
100	0.078	0.024	0.186	0.034	0.029	0.028	0.167	0.178	0.179	
79	0.074	0.044	0.144	0.032	0.055	0.06	0.173	0.118	0.143	Web

Fig. 3. Loss and JunkReduction of Meta Classification on Reuters, Newsgroups and IMDB

docRed	base1 avg(error)	base2 avg(error)	base3 avg(error)	base1 avg(loss)	base2 avg(loss)	base3 avg(loss)	base1 avg(jRed)	base2 avg(jRed)	base3 avg(jRed)	Dataset
0	0.517	0.509	0.509	0	0	0	0	0	0	IMDB
0.1	0.498	0.492	0.489	0.094	0.091	0.092	0.111	0.117	0.116	
0.2	0.48	0.472	0.47	0.183	0.182	0.183	0.234	0.237	0.233	
0.3	0.461	0.449	0.451	0.278	0.273	0.277	0.345	0.354	0.346	
0.4	0.441	0.431	0.433	0.372	0.369	0.374	0.456	0.462	0.452	
0.5	0.416	0.407	0.412	0.466	0.464	0.47	0.568	0.572	0.561	
0.6	0.397	0.389	0.391	0.567	0.565	0.567	0.666	0.669	0.666	
0.7	0.375	0.369	0.372	0.67	0.667	0.67	0.76	0.765	0.759	
0.8	0.351	0.345	0.348	0.777	0.776	0.777	0.847	0.849	0.847	
0.9	0.309	0.313	0.307	0.884	0.885	0.885	0.932	0.93	0.93	
0	0.42	0.417	0.417	0	0	0	0	0	0	Newsg.
0.1	0.386	0.384	0.383	0.073	0.075	0.074	0.154	0.15	0.153	
0.2	0.348	0.346	0.345	0.145	0.147	0.146	0.31	0.307	0.309	
0.3	0.307	0.305	0.304	0.218	0.22	0.219	0.463	0.461	0.462	
0.4	0.261	0.259	0.26	0.297	0.298	0.298	0.605	0.605	0.604	
0.5	0.216	0.215	0.216	0.385	0.387	0.387	0.729	0.727	0.726	
0.6	0.176	0.172	0.173	0.488	0.487	0.487	0.825	0.827	0.825	
0.7	0.139	0.135	0.137	0.602	0.6	0.601	0.897	0.899	0.897	
0.8	0.108	0.102	0.105	0.727	0.725	0.726	0.947	0.95	0.948	
0.9	0.081	0.076	0.078	0.86	0.859	0.859	0.98	0.982	0.981	
0	0.38	0.377	0.377	0	0	0	0	0	0	Reuters
0.1	0.336	0.336	0.334	0.057	0.06	0.058	0.185	0.181	0.183	
0.2	0.291	0.292	0.29	0.119	0.121	0.119	0.362	0.359	0.361	
0.3	0.247	0.247	0.245	0.188	0.19	0.189	0.523	0.52	0.522	
0.4	0.209	0.209	0.208	0.272	0.274	0.275	0.656	0.652	0.65	
0.5	0.174	0.172	0.172	0.369	0.369	0.37	0.763	0.762	0.761	
0.6	0.142	0.144	0.144	0.477	0.479	0.48	0.847	0.842	0.841	
0.7	0.113	0.111	0.109	0.596	0.595	0.595	0.908	0.909	0.91	
0.8	0.087	0.083	0.085	0.724	0.723	0.723	0.952	0.955	0.954	
0.9	0.068	0.064	0.068	0.86	0.859	0.859	0.981	0.983	0.981	
0	0.323	0.348	0.351	0	0	0	0	0	0	Web
0.1	0.266	0.311	0.316	0.041	0.074	0.079	0.24	0.163	0.15	
0.2	0.214	0.265	0.284	0.095	0.143	0.164	0.45	0.335	0.284	
0.3	0.168	0.215	0.229	0.166	0.214	0.225	0.62	0.505	0.479	
0.4	0.152	0.198	0.206	0.27	0.31	0.317	0.709	0.613	0.597	
0.5	0.134	0.162	0.177	0.377	0.4	0.409	0.792	0.738	0.716	
0.6	0.127	0.146	0.167	0.497	0.509	0.521	0.843	0.815	0.786	
0.7	0.116	0.119	0.15	0.62	0.62	0.634	0.888	0.888	0.856	
0.8	0.113	0.117	0.122	0.745	0.746	0.747	0.93	0.927	0.923	
0.9	0.131	0.056	0.093	0.873	0.862	0.868	0.962	0.987	0.974	

Fig. 4. Error, Loss and Junk Reduction for Restrictive Base Methods on Reuters, Newsgroups and IMDB and T = 25 TrainDocs per Class and for Web Documents with T = 79 TrainDocs per Class

Computer Science:

http://dir.yahoo.com/Science/Computer_Science/
<http://www.developer.com/>
<http://www.techweb.com/>
http://directory.google.com/Top/Computers/Computer_Science/
<http://library.albany.edu/subject/csci.htm>

Movies:

<http://www.allmovieportal.com/>
<http://www.galatta.com/>
<http://adutopia.subportal.com/cgi-bin/apollo/apollo.cgi>
http://dir.yahoo.com/Entertainment/Movies_and_Film/Genres/
<http://www.badmovies.org/>

Fig. 5. Starting Points for the Web Crawl

References

1. The 20 newsgroups data set. <http://www.ai.mit.edu/~jrennie/20Newsgroups/>.
2. Internet movie database. <http://www.imdb.com>.
3. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
4. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
5. C. Burges. A tutorial on Support Vector Machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.
6. S. Chakrabarti. *Mining the Web*. Morgan Kaufmann, 2003.
7. S. Dumais and H. Chen. Hierarchical classification of Web content. *SIGIR*, 2000.
8. S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of ACM-CIKM 98*, pp. 148-155., 1998.
9. Y. Freund. An adaptive version of the boost by majority algorithm. *Workshop on Computational Learning Theory*, 1999.
10. T. Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. *ECML*, 1998.
11. D. D. Lewis. Evaluating text categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Defense Advanced Research Projects Agency, Morgan Kaufmann, Feb. 1991.
12. N. Littlestone and M. Warmuth. The weighted majority algorithm. *FOCS*, 1989.
13. C. Manning and H. Schuetze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
14. C. V. Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33:2, pp. 106-119, 1977.
15. S. Siersdorfer and S. Sizov. Construction of feature spaces and meta methods for classification of Web documents (in german). *Conference on Database Systems for Business, Technology and Web (BTW)*, 2003.
16. S. Siersdorfer and S. Sizov. Restrictive clustering and metaclustering for self-organizing document collections. In *Proceedings of the 27th annual international conference on Research and development in information retrieval (SIGIR 04)*, 2004.
17. S. Siersdorfer, S. Sizov, and G. Weikum. Goal-oriented methods and meta methods for document classification and their parameter tuning. In *ACM Conference on Information and Knowledge Management (CIKM 04)*, Washington, 2004.
18. S. Sizov, M. Biwer, J. Graupmann, S. Siersdorfer, M. Theobald, G. Weikum, and P. Zimmer. The BINGO! system for information portal generation and expert Web search. *Conference on Innovative Systems Research (CIDR)*, 2003.
19. H. Wang, W. Fan, P. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. *SIGKDD*, 2003.
20. D. Wolpert. Stacked generalization. *Neural Networks*, Vol. 5, pp. 241-259, 1992.
21. Y. Yang and O. Pedersen. A comparative study on feature selection in text categorization. *ICML*, 1997.
22. H. Yu, K. Chang, and J. Han. Heterogeneous learner for Web page classification. *ICDM*, 2002.