Document Representations for Classification of Short Web-Page Descriptions*

Miloš Radovanović and Mirjana Ivanović

University of Novi Sad
Faculty of Science, Department of Mathematics and Informatics
Trg D. Obradovića 4, 21000 Novi Sad
Serbia and Montenegro
{radacha, mira}@im.ns.ac.yu

Abstract. Motivated by applying Text Categorization to sorting Web search results, this paper describes an extensive experimental study of the impact of bag-of-words document representations on the performance of five major classifiers – Naïve Bayes, SVM, Voted Perceptron, kNN and C4.5. The texts represent short Web-page descriptions from the dmoz Open Directory Web-page ontology. Different transformations of input data: stemming, normalization, logtf and idf, together with dimensionality reduction, are found to have a statistically significant improving or degrading effect on classification performance measured by classical metrics – accuracy, precision, recall, F_1 and F_2. The emphasis of the study is not on determining the best document representation which corresponds to each classifier, but rather on describing the effects of every individual transformation on classification, together with their mutual relationships.

1 Introduction

Text Categorization (TC – also known as *Text Classification* or *Topic Spotting*) is the task of automatically sorting a set of documents into *categories* (or *classes*, or *topics*) from a predefined set [1]. Applications of TC include text filtering (e.g. protection from spam e-mail), word sense disambiguation, and categorization of Web pages.

The initial motivation for this paper lies in the development a meta-search engine which uses TC to enhance the presentation of search results [2]. From the context of this system, we intended answer the three questions posed in [3]: (1) what representation to use in documents, (2) how to deal with the high number of features, and (3) which learning algorithm to use. This paper focuses on question one and its interaction with question three, trying (but not completely succeeding) to avoid question two.

Although the majority of works in TC employ the bag-of-words approach to document representation [4], studies of the impact of its variations on classification started appearing relatively recently. Leopold and Kindermann [5] experimented with the Support Vector Machine (SVM) classifier with different kernels, term frequency transforms

* This work was supported by project *Abstract Methods and Applications in Computer Science* (no. 144017A), of the Serbian Ministry of Science and Environmental Protection.

and lemmatization of German. They found that lemmatization usually degraded classification performance, and had the additional downside of great computational complexity, making SVMs capable of avoiding it altogether. Similar results were reported for neural networks on French [6]. Another study on the impact of document representation on one-class SVM [7] showed that, with a careful choice of representation, classification performance can reach 95% of the performance of SVM trained on both positive and negative examples. Kibriya et al. [8] compared the performance of SVM and a variant of the Naïve Bayes classifier, emphasizing the importance of term frequency and inverse document frequency transforms for Naïve Bayes.

This paper presents an extensive experimental study of bag-of-words document representations, and their impact on the performance on five classifiers commonly used in TC. An unorthodox evaluation methodology is used to measure and compare the effects of different transformations of input data on each classifier, and to determine their mutual relationships with regards to classification performance.

The next section outlines the experimental setup – how datasets were collected, which document representations were considered, and which classifiers. Section 3 presents the results – the representations that were found best, and the effects of and relationships between different transformations: stemming, normalization, logtf and idf. The final section concludes, and gives guidelines for future work.

2 The Experimental Setup

The WEKA Machine Learning environment [9] was used to perform all experiments described in this paper. The classical measures – accuracy, precision, recall, F_1 and F_2 [1] – were chosen to evaluate the performance of classifiers on many variants of the bag-of-words representation of documents (i.e. short Web-page descriptions) from the dmoz Open Directory.

Datasets. A total of eleven datasets were extracted from the dmoz ontology, one for each top-level category chosen for the meta-search system, namely *Arts*, *Business*, *Computers*, *Games*, *Health*, *Home*, *Recreation*, *Science*, *Shopping*, *Society* and *Sports*. The examples are either positive – taken from the corresponding category, or negative – distributed over all other categories, making this a binary classification problem.

When constructing the datasets and choosing the number of examples (around 700), care was taken to keep the number of features below 5000, for two reasons. The first reason was to give all classifiers an equal chance, because some of them are known not to be able to handle more than a couple of thousand features, and to do this without using some explicit form of feature selection (basically, to avoid question two from the Introduction). The second reason was the feasibility of running the experiments with the C4.5 classifier, due to its long training time. However, results from Section 3 (regarding the idf transform) prompted us to utilize the simple dimensionality reduction (DR) method based on term frequencies (TFDR), eliminating features representing the least frequent terms, at the same time keeping the number of features at around 1000. Therefore, two bundles of datasets were generated, one with and one without TFDR.

Document representations. Let W be the *dictionary* – the set of all terms (words) that occur at least once in a set of documents D. The bag-of-words representation of document d_j is a vector of weights $\boldsymbol{w}_j = (w_{1j}, \ldots, w_{|W|j})$. For the simplest binary representation where $w_{ij} \in \{0, 1\}$, let the suffix 01 be added to the names of the datasets, so, for instance, Arts-01 denotes the binary representation of the *Arts* dataset. Similarly, the suffix tf will be used when w_{ij} represent the frequency of the ith term in the jth document. *Normalization* (norm) can be employed to scale the frequencies to values between 0 and 1, accounting for differences in document lengths. The logtf transform may be applied to term frequencies, replacing the weights with $\log(1 + w_{ij})$. The *inverse document frequency* (idf) transform is expressed as: $\log(|D|/\mathrm{docfreq}(D, i))$, where $\mathrm{docfreq}(D, i)$ is the number of documents from D the ith term occurs in. It can be used by itself, or be multiplied with term frequency to yield the tfidf representation.

All these transformations, along with stemming (m), add up to 20 different variations of document representations, summarized in Table 1. This accounts for a total of $11 \cdot 20 \cdot 2 = 440$ different datasets for the experiments.

Table 1. Document representations

Not stemmed		**Stemmed**	
Not normalized	Normalized	Not normalized	Normalized
01		m-01	
idf		m-idf	
tf	norm-tf	m-tf	m-norm-tf
logtf	norm-logtf	m-logtf	m-norm-logtf
tfidf	norm-tfidf	m-tfidf	m-norm-tfidf
logtfidf	norm-logtfidf	m-logtfidf	m-norm-logtfidf

Classifiers. Five classifiers implemented in WEKA are used in this study: ComplementNaiveBayes (CNB), SMO, VotedPerceptron (VP), IBk, and J48.

CNB [10,8] is a variant of the classic Naïve Bayes algorithm optimized for applications on text. SMO is an implementation of Platt's Sequential Minimal Optimization algorithm for training SVMs [11]. VP was first introduced by Freund and Schapire [12], and shown to be a simple, yet effective classifier for high-dimensional data. IBk implements the classical k-Nearest Neighbor algorithm [13], and J48 is based on revision 8 of the C4.5 decision tree learner [14].

All classifiers were run using their default parameters, with the exception of SMO, where the option not to normalize training data was chosen. IBk performed rather erratically during initial testing, with performance varying greatly with different datasets, choices of k and distance weighing, so in the end we kept $k = 1$ as it proved most stable. Only later we realized this was because of IBk's use of the Euclidian distance measure, which tends to deform with high numbers of features. For IBk we will report only results without TFDR, since TFDR completely broke its performance.

3 Results

A separate WEKA experiment was run for every classifier with the 20 document representation datasets, for each of the 11 major categories. Results of all evaluation measures were averaged over five runs of 4-fold cross-validation. Measures were compared between *datasets* using the corrected resampled t-test implemented in WEKA, at $p = 0.05$, and the number of statistically significant wins and losses of each document representation added up for every classifier over the 11 categories.

For the sake of future experiments and the implementation of the meta-search system, best representations for each classifier were chosen, based on wins–losses values summed-up over all datasets. The declared best representations were not winners for all 11 categories, but showed best performance overall. Table 2 shows the best document representations for each classifier, along with their wins–losses values, before and after TFDR. Binary representations were practically never among the best, for all datasets.

Table 2. Wins–losses values of best document representations for each classifier, on datasets without (left columns) and with TFDR

	CNB		**SMO**		**VP**		**IBk**		**J48**	
	m-norm-tf		m-norm-logtf		m-logtf		m-norm-logtf		m-logtf	
Accuracy	41	1	1	37	15	2	119		40	40
Precision	45	1	20	6	29	12	11		−6	−5
Recall	4	1	−4	68	0	0	67		56	57
F_1	28	1	0	47	7	0	120		59	52
F_2	9	0	−3	71	0	0	78		63	57
Total	127	4	14	229	51	14	395		212	201

To illustrate the impact of document representations on classification, Table 3 summarizes the performance of classifiers on the best representations, and the improvements over the worst ones, on the Home dataset, without TFDR. Note that the emphasis of this paper is not on fine-tuning the classifiers using document representations, as much as it is on determining the impacts and relationships between different transforms (stemming, normalization, logtf, idf) and TFDR, with regards to each classifier. This is the prevailing subject of the remainder of this section.

Table 3. Performance of classification (in %) using the best document representations on the Home dataset *without* TFDR, together with improvements over the worst representations (statistically significant ones are in **boldface**)

	CNB	**SMO**	**VP**	**IBk**	**J48**
Accuracy	82.56 (**5.26**)	83.19 (1.67)	78.38 (**5.12**)	74.93 (**21.96**)	71.77 (**3.64**)
Precision	81.24 (**8.66**)	85.67 (**3.86**)	80.45 (**7.85**)	71.32 (**14.32**)	90.24 (1.60)
Recall	83.91 (1.81)	78.93 (3.80)	74.06 (0.96)	81.66 (**45.20**)	47.59 (**10.59**)
F_1	82.48 (**3.64**)	82.07 (2.17)	77.02 (**4.23**)	76.07 (**33.90**)	62.12 (**9.09**)
F_2	83.31 (2.19)	80.14 (3.30)	75.20 (2.16)	79.31 (**39.72**)	52.48 (**10.41**)

Effects of stemming. The effects of stemming on classification performance were measured by adding-up the wins–losses values for stemmed and nonstemmed datasets, and examining their difference, depicted graphically in Fig. 1. It can be seen that stemming improves almost all evaluation measures, both before and after TFDR. After TFDR, the effect of stemming is generally not as strong, which is understandable because its impact as a dimensionality reduction method is reduced. CNB is then practically unaffected, only SMO exhibits an increased tendency towards being improved. Overall, J48 is especially sensitive to stemming, which can be explained by its merging of words into more discriminative features, suiting the algorithm's feature selection method when constructing the decision tree.

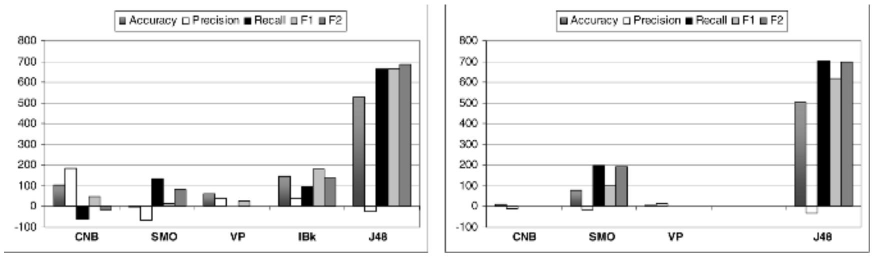

Fig. 1. The effects of *stemming* before (left) and after TFDR

To investigate the relationships between stemming and other transformations, a chart was generated for each transformation, measuring the effect of stemming on representations with and without the transformation applied. Figure 2 shows the effect of stemming on non-normalized and normalized data, without TFDR. It can be noted that normalized representations are affected by stemming more strongly (for the better). The same holds with TFDR applied. The logtf transform exhibited no influence on the impact of stemming, regardless of TFDR.

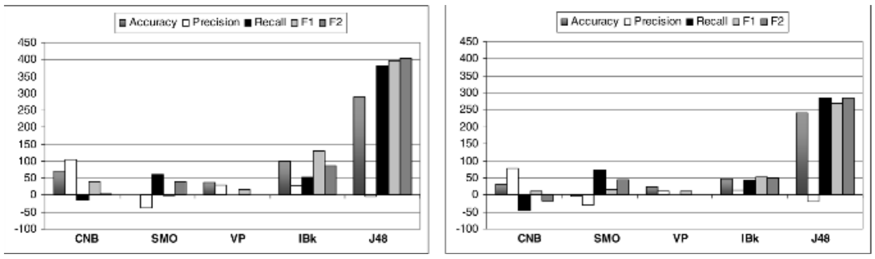

Fig. 2. The effects of *stemming* on non-normalized (left) and normalized datasets, *without* TFDR

The above analysis confirms the common view of stemming as a method for improving classification performance for English. However, this may not be the case for other languages, for instance German [5] and French [6].

Effects of normalization. The chart in Fig. 3 shows that normalization tends to improve classification performance in a majority of cases. Without TFDR, VP was virtually unaffected, CNB and SMO were improved on all counts but recall (and consequently F_2), while the biggest improvement was on IBk, which was anticipated since normalization assisted the comparison of document vectors. J48 was the only classifier whose performance worsened with normalization. Apparently, J48 found it tougher to find appropriate numeric intervals within the normalized weights, for branching the decision tree. After TFDR, CNB joined VP in its insensitivity, while SMO witnessed a big boost in performance when data was normalized.

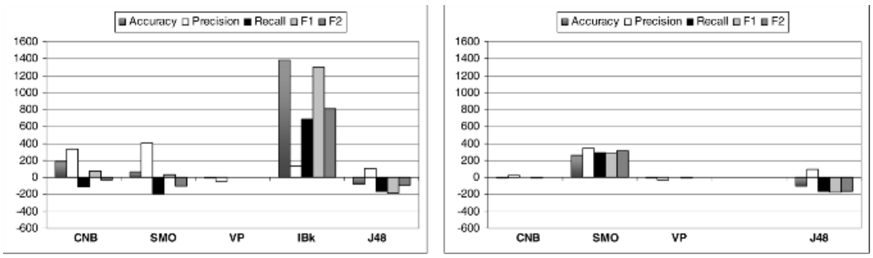

Fig. 3. The effects of *normalization* before (left) and after TFDR

No significant interaction between normalization and stemming was revealed, only that stemmed J48 was more strongly worsened by normalization. It seems that normalization misleads J48 from the discriminative features introduced by stemming.

Normalization and the logtf transform exhibited no notable relationship, while with idf transformed data, normalization had stronger influence on classification. After dimensionality reduction, this tendency was especially noticeable with the improvement of the precision of SMO (Fig. 4). This can be explained by the fact that idf severely worsens the performance of SMO after TFDR, and normalization compensated somewhat for this. This compensating effect of one transform on the performance degrading influences of another was found to be quite common in the experiments.

It is important to emphasize that the datasets used in the experiments consist of short documents, thus normalization does not have as strong an impact as it would have if the differences in document lengths were more drastic. Therefore, the conclusions above may not hold for the general case, for which a more comprehensive study is needed.

Effects of the logtf transform. As can be seen in Fig. 5, the logtf transform causes mostly mild improvements of classification performance. After TFDR, improvements are greater on SMO, while the impact on other classifiers is weaker.

Figure 6 shows that logtf has a much better impact on CNB when idf is also applied, without TFDR. This is similar to the compensating effect of normalization on idf with the SMO classifier. Relations change quite dramatically when TFDR is applied (both charts resemble Fig. 6 right), but the effect of logtf on SMO is again compensating. The improvements on CNB in both cases are especially significant, meaning that logtf and idf work together on improving classification.

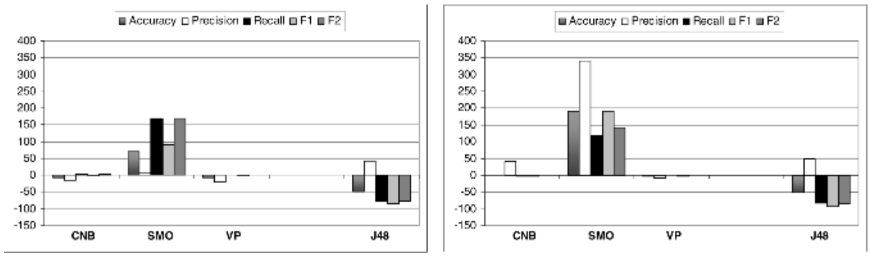

Fig. 4. The effects of *normalization* on datasets without (left) and with the idf transform applied to tf, *with* TFDR

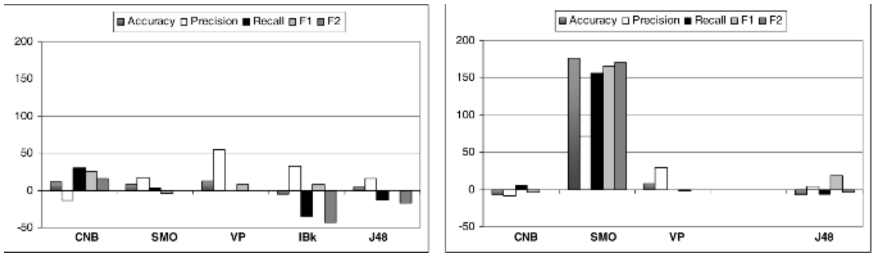

Fig. 5. The effects of the logtf transform before (left) and after TFDR

Before TFDR, the interaction of logtf and norm varied across classifiers: logtf improved CNB and IBk on normalized data, while others were improved without normalization. After TFDR, logtf had a weaker positive effect on normalized data, especially for CNB and SMO, which were already improved by norm (charts not shown).

Understandably, the logtf transform has a stronger positive impact on nonstemmed data, regardless of dimensionality reduction, with the exception of VP which exhibited no variations. This is in line with the witnessed improvements that stemming introduces on its own, and the already noted compensation phenomenon.

Effects of the idf transform. Applying the idf transform turned out to have the richest repertoire of effects, from significant improvement to severe degradation of classification performance. Figure 7 (left) illustrates how idf drags down the performance of all classifiers except SMO, without TFDR. For this reason we introduced TFDR in the first place, being aware that our data had many features which were present in only a few documents. We expected idf to improve classification, or at least degrade it to a lesser extent. That did happen, as Fig. 7 (right) shows, for all classifiers *except* SMO, whose performance drastically degraded! The simple idf document representation rose from being one of the worst, to one of the best representations, for all classifiers but SMO.

No significant correlation was detected by applying idf on stemmed and nonstemmed data. However, plenty of different effects were noticeable with regards to normalization. Without TFDR (Fig. 8), a stronger worsening effect on non-normalized data was exhibited with CNB, VP and IBk, while for SMO normalization dampened idf's

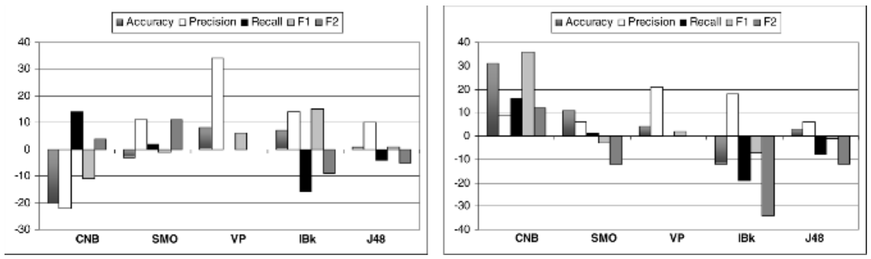

Fig. 6. The effects of the logtf transform on datasets without (left) and with the idf transform applied to tf, *without* TFDR

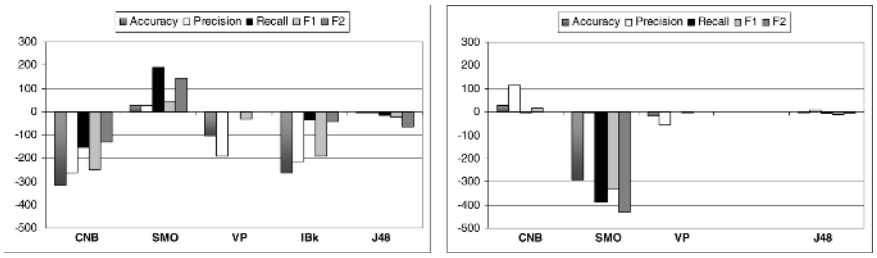

Fig. 7. The effects of idf applied to tf before (left) and after TFDR

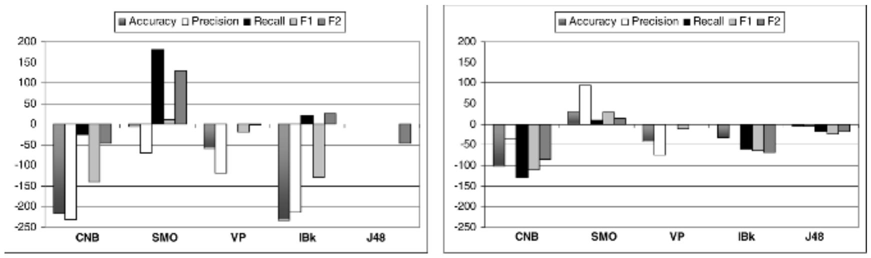

Fig. 8. The effects of idf applied to tf on non-normalized (left) and normalized datasets, *without* TFDR

improvement of recall, but overturned the degradation of accuracy and precision. With TFDR, the picture is quite different (Fig. 9): normalization improved the effects on CNB and VP, with SMO witnessing a partial improvement on precision, while J48 remained virtually intact. The impact of idf on (non-)logtfed datasets showed no big differences.

The above analysis shows the need to be careful when including the idf transform in the representation of documents. Removing infrequent features is an important prerequisite to its application, since idf assigns them often unrealistic importance, but that may not be enough, as was proved by the severe degradation of SMO's performance.

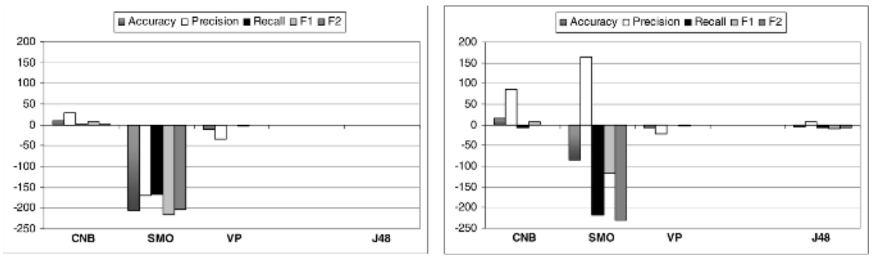

Fig. 9. The effects of idf applied to tf on non-normalized (left) and normalized datasets, *with* TFDR

4 Conclusions and Further Work

By using transformations in bag-of-words document representations there is, essentially, no new information added to a dataset which is not already there (except for the transition from 01 to tf representations). The general-purpose classification algorithms, however, are unable to derive such information without assistance, which is understandable because they are not aware of the nature of data being processed. Therefore, it can be expected of transforms to have a significant effect on classification performance, as was demonstrated at the beginning of Section 3.

Besides determining a best representation for each classifier, the experiments revealed the individual effects of transforms on different measures of classification performance, and some of their relationships. Stemming generally improved classification, partly because of its role as a dimensionality reduction method. It had an exceptionally strong improving impact on J48, which can be explained by its merging of words into more discriminative features, suiting the algorithm's feature selection method when constructing the decision tree. Normalization enhanced CNB, SMO and especially IBk, leaving VP practically unaffected and worsening J48. Although dmoz data consists of short documents, normalization did have a significant impact, but no definite conclusions may be drawn for the general case. The logtf transform had mostly a mild improving impact, except on SMO after TFDR, which exhibited stronger improvement. SMO is known to work well with small numeric values, which explains its sensitivity to normalization and logtf. The situation with idf was trickier, with the effects depending strongly on dimensionality reduction for CNB and SMO, but in opposite directions: CNB was degraded by idf before, and improved after TFDR; for SMO it was vice versa.

The most common form of relationship between transforms that was noticed were the compensating effects of one transform on the performance degrading impact of another (e.g. norm and logtf on idf). The logtf and idf transforms seemed to work together on improving CNB after TFDR. The impact of idf on normalization was most complex, with great variation in the effects on different evaluation measures. Note that the method for determining relations between transforms appeared not to be commutative, e.g. the effects of normalization on idfed data and of idf on normalized data were not the same.

The comments above refer to the general case of performance measuring. Some transforms (e.g. idf) may improve one measure, at the same time degrading another.

Often, the preferred evaluation measure, chosen with the application of the classifier in mind, will need to be monitored when applying the results presented in this paper.

The main difficulty with comprehensive TC experiments is sheer size. Roughly speaking, factors such as datasets, document representations, dimensionality reduction methods, reduction rates, classifiers, and evaluation measures, all have their counts multiplied, leading to a combinatorial explosion which is hard to handle. We tackled this problem by excluding detailed experimentation with DR, and using dmoz as the only source of data. Therefore, no definite truths, but only pointers can be derived from the described experience. A more comprehensive experiment, featuring other common corpora (Reuters, OHSUMED, 20Newsgorups etc.), and more dimensionality reduction methods, is called for to shed more light on the relationships of all above mentioned factors. In the next phase, however, we plan to conduct experiments with DR methods on dmoz data, with the document representations that were determined best for each classifier, before applying the winning combination to categorization of search results.

References

1. Sebastiani, F.: Text categorization. In Zanasi, A., ed.: Text Mining and its Applications. WIT Press, Southampton, UK (2005)
2. Radovanović, M., Ivanović, M.: CatS: A classification-powered meta-search engine. In: Advances in Web Intelligence and Data Mining. Studies in Computational Intelligence 23, Springer-Verlag (2006)
3. Mladenić, D.: Text-learning and related intelligent agents. IEEE Intelligent Systems, Special Issue on Applications of Intelligent Information Retrieval **14**(4) (1999) 44–54
4. Gabrilovich, E., Markovitch, S.: Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5. In: Proceedings of ICML04, 21st International Conference on Machine Learning, Baniff, Canada (2004)
5. Leopold, E., Kindermann, J.: Text categorization with Support Vector Machines. How to represent texts in input space? Machine Learning **46** (2002) 423–444
6. Stricker, M., Vichot, F., Dreyfus, G., Wolinski, F.: Vers la conception automatique de filtres d'informations efficaces. In: Proceedings of RFIA2000, Reconnaissance des Formes et Intelligence Artificielle. (2000) 129–137
7. Wu, X., Srihari, R., Zheng, Z.: Document representation for one-class SVM. In: Proceedings of ECML04, 15th European Conference on Machine Learning. LNAI 3201, Pisa, Italy (2004)
8. Kibriya, A.M., Frank, E., Pfahringer, B., Holmes, G.: Multinomial naive bayes for text categorization revisited. In: Proceedings of AI2004, 17th Australian Joint Conference on Artificial Intelligence. LNAI 3339, Cairns, Australia (2004) 488–499
9. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. 2nd edn. Morgan Kaufmann Publishers (2005)
10. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R.: Tackling the poor assumptions of naive Bayes text classifiers. In: Proceedings of ICML03, 20th International Conference on Machine Learning. (2003)
11. Platt, J.: Fast training of Support Vector Machines using Sequential Minimal Optimization. In: Advances in Kernel Methods – Support Vector Learning. MIT Press (1999)
12. Freund, Y., Schapire, R.E.: Large margin classification using the perceptron algorithm. Machine Learning **37**(3) (1999) 277–296
13. Aha, D., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6**(1) (1991) 37–66
14. Quinlan, R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)