Enhancing Text Classification by Information Embedded in the Test Set

Gabriela Ramírez-de-la-Rosa, Manuel Montes-y-Gómez, Luis Villaseñor-Pineda

Laboratory of Language Technologies
National Institute of Astrophysics, Optics and Electronics
Luis Enrrique Erro No. 1, Sta. María Tonantzintla, Pue.; 72840; Mexico
{gabrielarr, mmontesg, villasen}@inaoep.mx

Abstract. Current text classification methods are mostly based on a supervised approach, which require a large number of examples to build models accurate. Unfortunately, in several tasks training sets are extremely small and their generation is very expensive. In order to tackle this problem in this paper we propose a new text classification method that takes advantage of the information embedded in the own test set. This method is supported on the idea that similar documents must belong to the same category. Particularly, it classifies the documents by considering not only their own content but also information about the assigned category to other similar documents from the same test set. Experimental results in four data sets of different sizes are encouraging. They indicate that the proposed method is appropriate to be used with small training sets, where it could significantly outperform the results from traditional approaches such as Naive Bayes and Support Vector Machines.

1 Introduction

The tremendous amount of digital documents available on the Web has motivated the development of different automatic mechanisms that facilitate their access, organization and analysis. One example of such mechanisms are text classification methods, which focus on the assignment of documents into a set of predefined classes or topics [1].

Over the years several methods and algorithms for text classification have been proposed. In particular, the leading approach considers the use of machine learning techniques such as bayesian models, support vector machines and prototype-based classifiers to mention some. Under this supervised approach it is necessary to have an adequate training set consisting of manually labeled documents. As expected, the more the labeled documents are, the better the classification model is [2]. Unfortunately, in many real-world applications training sets are extremely small, and, what is more, their generation is very expensive.

Regarding the above problem, current efforts have focused on the generation of high-performance classification models using few labeled training data. On the one hand, some methods take advantage of available unlabeled documents to,

iteratively, generate a robust classification model [3, 4]. On the other hand, there are some methods that use information about the similarity of the documents from the own test collection in order to improve their classification. Particularly, most of these methods employ clustering techniques to enrich the representation of documents by adding or replacing some attributes [2, 5, 6].

The approach proposed in this paper belongs to the second group of works; nevertheless, it does not aim to enrich the representation of test documents by including information extracted from other similar documents, instead, it attempts to improve their individual classifications by considering the categories assigned to their nearest neighbors (from to same test set). In other words, the idea behind our proposal may be expressed by the popular proverb "a man is known by the company he keeps".

Given that prototype-based classifiers are very simple and have demonstrated to consistently outperform others algorithms such as Naive Bayes, K-Nearest Neighbors and C4.5 in text classification tasks [1], we decided to implemented the proposed approach using this classification algorithm. In general, our prototype-based method decides about the category of a given document by determining the class which prototype is more similar to it and its nearest neighbors.

Experimental results in four data sets of different sizes are encouraging. They indicate that the proposed approach could significantly outperform the results from a traditional prototype-based method as well as the results achieved by Naive Bayes and Support Vector Machines. On the other hand, these results also demonstrate the appropriateness of the approach for dealing with small training sets.

The remainder of paper is organized as follows. Section 2 explains the prototype-based classification method. Section 3 introduces the proposed approach. Section 4 describes the experimental configuration and shows the results obtained in four document collections. Finally, Section 5 presents our conclusions and exposes some future work ideas.

2 Prototype-based Classification

This section describes the prototype-based classification method, which is used as base method in the proposed approach.

Prototype-based classification is one of the traditional methods for supervised text classification. This method may be summed up in a few words as follows. In the training phase, it considers the construction of one single representative instance, called prototype, for each class. Then, in a test phase, each given unlabeled document is compared against all prototypes and is assigned to the class having the greatest similarity score [1, 7–9]. Evidently there are several ways to calculate the prototypes as well as to measure the similarity between documents and prototypes. Next we describe the alternative used in this paper.

The definition of the prototype for each class c_i is based on the normalized sum model, where each class is represented by a vector which is the sum of all

document vectors from the class, normalized so that it has a unitary length [9, 10]:

$$P_i = \frac{1}{||\sum_{d \in c_i} d||} \sum_{d \in c_i} d \tag{1}$$

In this case, documents are represented by vectors in the term-space, $d = \{w_1, w_2, ..., w_m\}$, where m indicates the number of different terms in the whole training set.

On the other hand, the assignation of the category to a given unlabeled document d is based on the following criterion:

$$class(d) = \arg\max_i(sim(d, P_i)) \tag{2}$$

where,

$$sim(d, P_i) = \frac{d \cdot P_i}{||d|| \times ||P_i||} \tag{3}$$

In Formulas 1 and 3, $||z||$ denotes the 2-norm of z, and $v \cdot z$ denotes the dot product of v and z vectors.

3 The Proposed Method

Figure 1 shows the general schema of the proposed method. It consists of two main phases. The first focuses on the construction of the class prototypes using the traditional techniques. The second involves, on the one hand, the identification of the nearest neighbors for each unlabeled document, and, on the other hand, their classification considering information from their own as well as from their neighbors. Following we present a brief description of each one of these processes.

Prototype Construction. This process carries out the construction of the class prototypes. In particular, given a set of labeled documents (i.e., training set) organized in set of classes, it computes the prototype for each class using Formula 1. This process is performed only once at the training phase.

Nearest Neighbors Identification. This process focuses on the identification of the N nearest neighbors for each document of the test set. In order to do that it firstly computes the similarity between each pair of documents from the test set using the cosine formula (refer to Formula 3), and then, based on the obtained similarity values, selects the N nearest neighbors for each document.

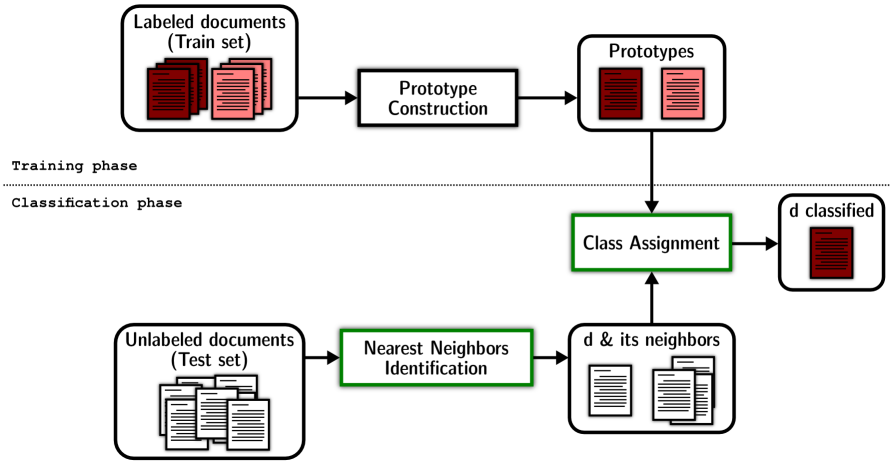


Fig. 1. Genaral scheme of the proposed text classification method.

Class Assignment. Given a document d from the test set in conjunction with its N nearest neighbors, this process assigns a class to d using the following formula:

$$class(d) = \arg\max_i \left(sim(d, P_i) + \lambda \frac{1}{N} \sum_{j=1}^{N} [inf(d, v_j) \times sim(v_j, P_i)] \right) \quad (4)$$

where,

- sim is the cosine similarity function defined in Formula 3.
- N is the number of neighbors considered to provide information about document d.
- λ is a constant used to determine the relative importance of both, the information from the own document (d) and the information from its neighbors. The greater the value of λ is, the greater the contribution of the neighbors, and vice versa.
- inf is an influence function used to weight the contribution of each neighbor v_j to the classification of d. The purpose of this function is to give more relevance to the closer neighbors. In particular, we define this influence in direct proportion to the similarity between each neighbor and d calculated using the cosine formula (refer to Formula 3).

In order to give more information about these processes, Figure 2 presents the algorithm of the proposed method.

Let L be the set of labeled documents from the training set, U the set of test documents, C the set of classes in the set L, V^d the set of N neighbors of d, T_L the terms obtained from L, T_U the terms obtained from U.

```
Represent each d ∈ L by a vector d = {t_1, t_2, ..., t_|T_L|}.
For each c_i ∈ C
    Compute the prototype P_i using Formula 1.

Represent each d ∈ U by a vector d = {t_1, t_2, ..., t_|T_U|}.
For each d ∈ U
    V^d ← ∅.
    repeat from 1 to N
        Search v ∈ {U − V^d − d} : sim(d, v) is the greatest, where sim
        is given by Formula 3.
        V^d ← {V^d + v}.

Represent each d ∈ U by a vector d = {t_1, t_2, ..., t_|T_L|}.
For each d ∈ U
   Assign a class using Formula 4.
```

Fig. 2. Algorithm of the proposed method.

4 Experimental Evaluation

4.1 Datasets

For the evaluation of the proposed method we considered the R8 collection. This collection was previously used by Cardoso-Cachopo and Oliveira [9], and it is formed by the eight largest classes from the Reuters-21578 collection, which documents belong to only one class. Table 1 shows some data about this collection, such as the number of documents per class in the training and test sets.

With the aim of evaluating the proposed method in situations having small training sets, we generated three smaller collections from the original R8 corpus: R8-reduced-41, R8-reduced-20 and R8-reduced-10, consisting of 41, 20 and 10 labeled documents per class respectively. Table 2 shows some data about these four collections, such as the number of documents in the training set and the number of terms from the vocabulary of each class. The number of documents in the test set were not included since they are the same for all collections (2189) and were previously presented in Table 1.

Table 1. The R8 collection

Class	Documents in training set	Documents in test set
acq	1596	696
crude	253	121
earn	2840	1083
grain	41	10
interest	190	81
money-fx	206	87
ship	108	36
trade	251	75
Total	5485	2189

Table 2. The four evaluation datasets

Collection	Documents in training set	Vocabulary
R8	5485	3711
R8-reduced-41	328	2887
R8-reduced-20	160	1807
R8-reduced-10	80	1116

4.2 Evalaution Measure

The evaluation of the performance of the proposed method was carried out by means of the F-measure. This measure is a linear combination of the precision and recall values from all class $c_i \in C$. It is defined as follows:

$$F - Measure = \frac{1}{|C|} \sum_{i=1}^{|C|} \left[\frac{2 \times Recall(c_i) \times Precision(c_i)}{Recall(c_i) + Precision(c_i)} \right] \tag{5}$$

$$Recall(c_i) = \frac{number\ of\ correct\ predictions\ of\ c_i}{number\ of\ examples\ of\ c_i} \tag{6}$$

$$Precision(c_i) = \frac{number\ of\ correct\ predictions\ of\ c_i}{number\ of\ predictions\ as\ c_i} \tag{7}$$

4.3 Baseline Results

In order to generate the baseline results we considered three of the most used methods for text classification, namely, Naive Bayes (NB) [11], Support Vector Machines (SVM) [12] and the prototype-based method (PBC) described in Section 2. Table 3 shows the results obtained by these methods in the four used

datasets. These results confirm the robustness of the prototype-based method for dealing with small training sets. Particularly, it is of special interest to notice that reducing the training set in 94% (R8-reduced-41 in relation to R8) only caused a decrement of 4.7% in the F-measure value.

Table 3. F-measure results from three classification methods

Collection	NB	SVM	PBC
R8	0.828	**0.886**	0.876
R8-reduced-41	0.747	0.812	**0.836**
R8-reduced-20	0.689	0.760	**0.803**
R8-reduced-10	0.634	0.646	**0.767**

4.4 Results

As described in Section 3, the main idea of the proposed method is to classify the documents by considering not only their own content but also information about the assigned category to other similar documents. Based on this idea, Formula 4 attempts to combine both kinds of information, being λ a constant that determines their relative importance.

Considering the above situation, we designed the experiments in such a way that we could evaluate the impact on the classification results caused by the selection of different values of λ. In particular we used $\lambda = 1, 2, 3$ in order to assign equal, double or triple relevance to the neighbors information in relation to the information from the document itself.

In addition, with the purpose of analyzing the impact caused by the inclusion of non-relevant neighbors into the class assignment process, we also considered different number of neighbors; we used $N = 1...30$.

Experiment 1. The objective of this experiment was to analyze the performance of the proposed method in collections having small training sets, which complicate the construction of accurate classification models. Table 4 shows the F-measure values achieved by the proposed method in three collections using different values of λ and N. Results in bold indicate that the method significantly outperformed the baseline result. We evaluated the statistical significance of results using the z-test with a confidence of the 95%.

The obtained results show that the method could improve the classification performance in all collections, but especially in those having smaller training sets. For instance, for the R8-reduced-10 collection the improvement was as high as 9.7%. There is also important to mention that the method demonstrated not to be very sensitive to the values of λ and N, achieving –in general– the best results with $\lambda = 3$ and $N < 10$.

Table 4. F-measure results of the proposed method on the three reduced datasets

	R8-reduced-41 (baseline=0.836)			R8-reduced-20 (baseline=0.803)			R8-reduced-10 (baseline=0.767)		
N	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$
1	0.843	0.845	0.825	0.813	**0.821**	0.804	**0.807**	**0.804**	0.780
2	**0.864**	**0.864**	**0.865**	0.819	**0.824**	**0.831**	**0.813**	**0.816**	**0.821**
3	**0.866**	**0.865**	**0.871**	**0.839**	**0.845**	**0.846**	**0.806**	**0.829**	**0.825**
4	**0.864**	**0.871**	**0.880**	**0.838**	**0.845**	**0.846**	**0.813**	**0.829**	**0.836**
5	**0.859**	**0.876**	**0.866**	**0.832**	**0.846**	**0.844**	**0.819**	**0.829**	**0.836**
6	**0.863**	**0.865**	**0.863**	**0.831**	**0.860**	**0.844**	**0.812**	**0.826**	**0.837**
7	**0.873**	**0.867**	**0.863**	**0.839**	**0.855**	**0.839**	**0.812**	**0.829**	**0.841**
8	**0.870**	**0.863**	**0.861**	**0.845**	**0.858**	**0.841**	**0.812**	**0.829**	**0.838**
9	**0.866**	**0.861**	**0.862**	**0.847**	**0.854**	**0.851**	**0.812**	**0.827**	**0.836**
10	**0.861**	**0.857**	**0.862**	**0.841**	**0.852**	**0.837**	**0.813**	**0.825**	**0.829**
11	**0.853**	**0.855**	**0.862**	**0.842**	**0.851**	**0.825**	**0.811**	**0.827**	**0.829**
12	0.850	**0.852**	**0.860**	**0.840**	**0.833**	**0.827**	**0.811**	**0.829**	**0.836**
13	0.851	**0.853**	**0.859**	**0.840**	**0.837**	**0.831**	**0.810**	**0.822**	**0.813**
14	0.849	**0.854**	**0.858**	**0.838**	**0.836**	**0.832**	**0.808**	**0.817**	**0.818**
15	0.849	**0.854**	**0.854**	**0.823**	**0.832**	**0.830**	**0.804**	**0.823**	**0.806**
16	0.842	**0.854**	0.844	**0.821**	**0.828**	**0.830**	**0.804**	**0.822**	**0.805**
17	0.848	**0.855**	0.840	**0.823**	**0.829**	**0.827**	**0.805**	**0.822**	**0.802**
18	0.850	**0.854**	0.830	**0.822**	**0.831**	0.819	**0.802**	**0.813**	**0.801**
19	**0.852**	**0.854**	0.830	0.817	**0.832**	0.818	**0.801**	**0.811**	**0.798**
20	0.851	**0.853**	0.842	0.816	**0.833**	**0.822**	**0.796**	**0.811**	**0.798**
21	**0.852**	**0.853**	0.845	0.816	**0.831**	**0.824**	**0.795**	**0.804**	**0.807**
22	**0.853**	**0.853**	0.845	0.814	**0.830**	**0.825**	**0.797**	**0.803**	**0.797**
23	0.851	**0.853**	0.844	0.816	**0.827**	**0.822**	**0.796**	**0.798**	**0.798**
24	0.848	0.845	0.840	0.806	0.817	**0.821**	**0.795**	**0.805**	**0.796**
25	0.849	0.845	0.839	0.805	0.817	0.819	**0.795**	**0.800**	**0.794**
26	0.846	**0.853**	0.839	0.807	0.818	**0.820**	**0.795**	**0.800**	**0.803**
27	0.846	**0.856**	0.839	0.807	0.817	**0.820**	**0.795**	**0.800**	**0.803**
28	0.846	**0.852**	0.839	0.811	0.819	**0.823**	**0.794**	**0.799**	**0.800**
29	0.843	**0.852**	0.839	0.810	0.820	**0.822**	**0.795**	**0.801**	**0.789**
30	0.843	**0.854**	0.836	0.810	0.820	**0.822**	**0.795**	**0.801**	**0.788**

Experiment 2. The objective of this second experiment was to evaluate the performance of the method in a traditional classification scenario, having available enough training examples. In particular, we used the R8 collection which allowed to generate accurate classification models as shown in Section 4.3. Table 5 shows the results from this experiment, indicating in bold numbers the cases where the proposed method significantly outperformed the baseline result. In general, these results indicate that our method could also obtain satisfactory results with a larger training set. However, in this case, most relevant results were achieved using $\lambda = 1$.

Comparative analysis of results. In order to get a better idea about the behavior of the proposed method, Figure 3 shows its performance in all collections using the different values of λ and N. From this figure it is possible to make the following observations regarding this method:

- It requires a relative small number of neighbors to achieve the highest performance value; in all collections it used less than 10 neighbors. Moreover, as intuitively expected, it is possible to notice that the lesser the number of training examples, the greater the number of neighbors required to achieve the maximum performance value. For instance, for R8 there were needed only four neighbors, whereas, for R8-reduced-10 there were seven.
- The lower the number of documents in the training set, the greater the improvement achieved by the proposed method (in comparison with the baseline). This fact demonstrates that this method is especially appropriate to

Table 5. F-measure results of the proposed method on the R8 collection

	(baseline=0.876)		
N	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$
1	**0.898**	0.880	0.865
2	**0.894**	**0.900**	0.879
3	**0.893**	**0.902**	0.889
4	**0.893**	**0.894**	**0.905**
5	**0.892**	**0.902**	**0.895**
6	**0.895**	0.888	0.884
7	**0.893**	**0.891**	0.885
8	**0.894**	0.884	**0.891**
9	**0.895**	0.887	0.877
10	**0.893**	0.883	0.878
11	**0.894**	0.886	0.881
12	**0.899**	0.883	0.877
13	0.886	0.875	0.872
14	0.885	0.879	0.876
15	0.887	0.869	0.870
16	0.887	0.869	0.869
17	0.879	0.870	0.863
18	0.878	0.870	0.859
19	0.880	0.868	0.858
20	0.880	0.879	0.858
21	0.881	0.879	0.856
22	0.880	0.880	0.858
23	0.880	0.881	0.858
24	0.881	0.881	0.857
25	0.880	0.880	0.857
26	0.879	0.881	0.852
27	0.878	0.878	0.854
28	0.879	0.877	0.856
29	0.881	0.877	0.857
30	0.880	0.879	0.857

be used with small training sets, where current approaches tend to generate poor classification models.

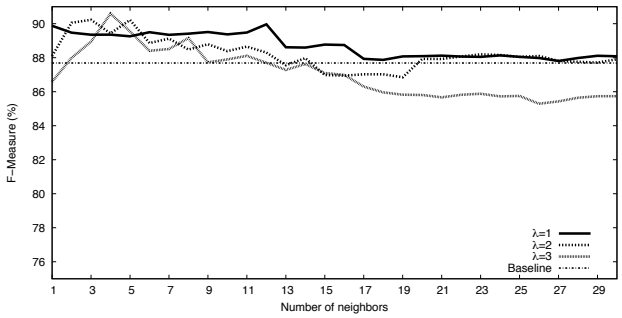
– In most cases the best results were achieved using $\lambda > 1$. Somehow this fact indicates that information from neighbors may be useful in practically any classification scenario, including or not sufficient training examples.

5 Conclusions

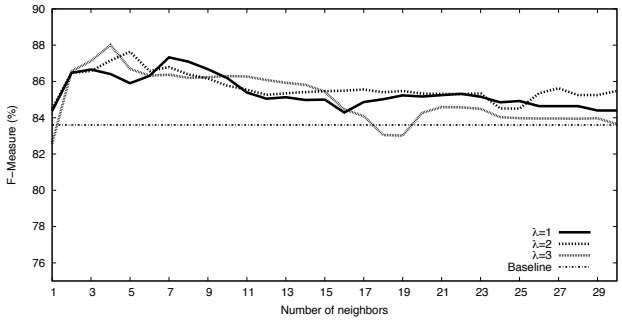
Inspired by the popular proverb "a man is known by the company he keeps", in this paper we proposed a new text classification method that carries out the classification of documents by considering not only their own content but also the information about the assigned category to their similar documents.

Experimental results in four collections with training sets of different sizes demonstrated the robustness of the proposed method, which could significantly outperformed the results from methods such as Naive Bayes, Support Vector Machines (SVM) and a traditional prototype-based classifier. In relation to this last point, it is important to point out that the proposed method, using only 2% of the labeled instances (i.e., R8-reduced-10), achieved a similar performance than Naive Bayes when it employed the complete training set (i.e., R8).

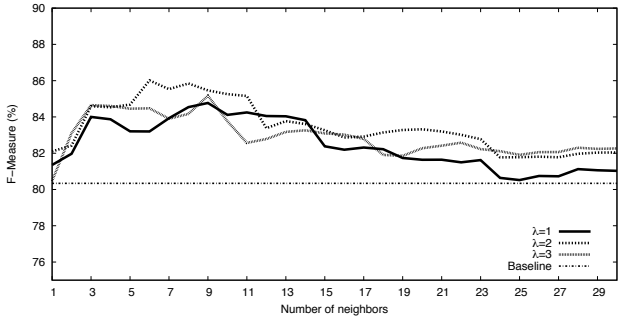
As described in Section 3, the proposed method has two main parameters: λ and N. Experimental results indicated that the method is not very sensitive to the selection of the value of these parameters. Nevertheless, it was observed that the lesser the number of training examples, the greater the values of λ and N required to achieve the maximum performance value.



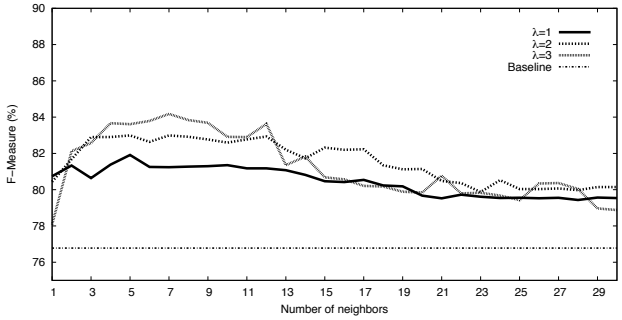
(a) R8



(b) R8-reduced-41



(c) R8-reduced-20



(d) R8-reduced-10

Fig. 3. Comparison of the results obtained in four collections using different values of λ and N.

As future work we plan to: (i) implement the proposed method using other algorithms as base classifiers such as Naive Bayes and SVM, (ii) evaluate the method in a cross-lingual text classification task as well as in a transfer-learning kind of problem, and (iii) define the influence function based on other similarity measures.

References

1. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, London, UK, Springer-Verlag (2000) 424–431
2. Derivaux, S., Forestier, G., Wemmert, C.: Improving supervised learning with multiple clusterings. In: Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications in conjunction with ECAI, Patras, Greece (2008) 57–60
3. Cong, G., Lee, W.S., Wu, H., Liu, B.: Semi-supervised text classification using partitioned em. In: 11 th Int. Conference on Database Systems for Advanced Applications (DASFAA). (2004) 229–239
4. Nigam, K., Mccallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using em. In: Machine Learning. (1999) 103–134
5. Kyriakopoulou, A., Kalamboukis, T.: Using clustering to enhance text classification. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, New York, USA, ACM (2007) 805–806
6. Fang, Y.C., Parthasarathy, S., Schwartz, F.: Using clustering to boost text classification. In: Workshop on Text Mining (TextDM'2001). (2001)
7. Lertnattee, V., Theeramunkong, T.: Term-length normalization for centroid-based text categorization. In: Knowledge-Based Intelligent Information and Engineering Systems. (2003) 850–856
8. Guan, H., Zhou, J., Guo, M.: A class-feature-centroid classifier for text categorization. In: WWW '09: Proceedings of the 18th international conference on World wide web, ACM (2009) 201–210
9. Cardoso-Cachopo, A., Oliveira, A.L.: Semi-supervised single-label text categorization using centroid-based classifiers. In: SAC '07: Proceedings of the 2007 ACM symposium on Applied computing, ACM (2007) 844–851
10. Tan, S.: An improved centroid classifier for text categorization. Expert Systems with Applications **35** (2008) 279–285
11. Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval, Springer Verlag (1998) 4–15
12. Joachims, T.: Text categorization with support vector machines: Learning with many relevant features, Springer Verlag (1998) 137–142