Text Classification by PNN-based Term Re-weighting

Atilla Elçi
Department of Computer and
Educational Technologies
Süleyman Demirel University
Isparta 32260 Turkey

ABSTRACT
Current approaches to feature selection for text classification aim to reduce the number of terms that are used to describe documents. Thus, documents can be classified and found with greater ease and precision. A key shortcoming of these approaches is that they select the topmost terms to describe documents after ranking all terms using a feature selection measure (scoring function). Lesser high-ranking terms below the topmost terms are discarded to reduce computational costs. Nevertheless, in many cases, they may have considerable discriminative power to enhance the text classification precision. In order to address this issue, we proposed a new feature weighting formalism that ties the topmost terms with lesser high-ranking terms using probabilistic neural networks. In the proposed method, probabilistic neural networks are formed using relative category distribution matrix and topmost terms are re-weighted and passed to Rocchio classifier. This is achieved without increasing the dimensionality of the feature space. Through experiments on datasets from Reuters news collection RCV1, we show that the proposed method is a significant supplement to the statistical feature selection measures for better text classification at extreme term filtering ranges.

Keywords
Term re-weighting, boosting, probabilistic neural networks, text classification, feature selection, Rocchio classifier.

1. INTRODUCTION
Traditionally, documents are examined and classified based on their subjects by employees at many organizations. A large amount of human resources is spent in carrying out such a task, nevertheless results often lack accuracy. Text classification methods have been developed to automate the assignment of text-based documents to various classes [16, 8]. Cluster analysis offers mainstay methodologies for multivariate data analysis [7]. Methods that consider documents as bag of words are made of several stages. Firstly, words that are invariant of topics such as prepositions, articles, and conjunctions are removed. Remaining words are stemmed to group those with same root together. The most discriminative words are selected to reduce the dimensionality of the vector space for document representation. Words are weighted using a particular weighting scheme for better document representation. Finally, one classifier is preferred from a set of classifiers for document categorization. Feature selection and weighting are crucial before passing document vectors to classifiers. A good feature selection measure coupled with appropriate feature weighting can dramatically decrease the size of input vectors. This in return increases classifiers' computation speed, and can help maintain high accuracy.

Numerous approaches have been proposed to identify important concepts in text documents. Some of them make use of lexical chains and WordNet. Lexical chains were introduced to capture concept relations in text documents [13]. A lexical chain holds a set of semantically related words of a text. WordNet is a lexical ontology in which nouns, verbs, adjectives and adverbs are organized into synonym sets. The synonym sets are related to other synonym sets by different types of relations. The most common relations in WordNet are the part of and the kind of relations. In a later study, lexical chains were constructed using WordNet for text summarization [2]. In another study, conceptual similarities among terms were computed using WordNet for term re-weighting and expansion to help document retrieval [18]. In a different work, concept clusters were defined using WordNet to lower term dimension in a document [6]. In semantic approaches, on the whole, knowledge bases such as WordNet must be available to create detailed semantic representation of a document. Besides, lexical chain construction is laborious. Words can have many senses; therefore word sense disambiguation is a must in order to build effective lexical chains.

Statistical approaches take corpora into consideration for better document classification. In an earlier work with distributional word clustering, the classification accuracy of word clusters on the 20 Newsgroups Dataset was notable [1]. In another study, words were clustered using a pure statistical method and word clusters outperformed word-based representation on the 20 Newsgroups Dataset in terms of categorization accuracy and representation efficiency [3]. In the same study, word-based representation (bag of words) outperformed word-cluster representation on the Reuters 21578 Dataset. It was discovered that some datasets can be categorized with optimal accuracy using a small set of words, whereas others required many hundreds more words to reach optimal accuracy. For a multi-class classification problem without taking hierarchical structure into account, pair-wise coupling was used in computing probabilities and comparing it to other approaches [19]. On the other hand, applying boosting to hierarchical text classification by taking into account the hierarchical structure of the Reuters Corpus Volume 1 news collection [10] increased recall, decreased precision, and increased F1-values [5].

Statistical feature selection measures such as expected cross entropy and Gini index greatly improve classifiers' performance by producing fair rankings of terms before categorization [17]. Only a subset that includes the topmost features is considered

for text categorization and lesser high-ranking features are excluded to ease computational burden at the cost of categorization accuracy. In order to address this shortcoming, in the present study we developed a new methodology to re-weight the topmost features given lesser high-ranking features. This is essentially a multi-class classification problem where one assigns each of the observations into one of k classes. In this paper we evolve a technique for multi-class classification by considering pair-wise comparisons of features. In carrying out pair-wise coupling we use a probabilistic neural network (PNN) in the backstage. Thus, feature selection and re-weighting are combined in order to significantly enhance existing feature selection methods at extreme term filtering ranges. Feed-forward neural networks, which offer an approach for very flexible modeling, have been a popular tool for classification [9], however in this work for the first time we employ PNN for boosting classification selection.

This paper is organized as follows. Section 2 briefly introduces prominent statistical feature selection measures and section 3 does the same for popular term weighting measures. Likewise the section 4 briefly describes Rocchio classifier. In section 5, we explain the proposed method. We discuss our experiments in section 6 where we also present the results of traditional feature selection measures with and without our supplementary methodology. Conclusions and future directions for term re-weighting in PNN are mentioned in section 7.

2. STATISTICAL FEATURE SELECTION MEASURES

The common characteristic of statistical feature selection measures is that they place terms that do not have enough discriminative power low in the term rankings; thus, the size of the feature space can be reduced by filtering low-ranking terms. This is generally achieved by using occurrence distribution, relative word distribution, and relative category distribution. Following examples of occurrence distribution as presented in Table 1, derived relative word distribution in Table 2, and relative category distribution in Table 3 are generated from a small subset of Yahoo sports pages. For the sake of simplifying computation, it is assumed that there are just 4 words and 5 categories in the given corpus.

Table 1. Example: Occurrence distribution matrix for words.

Word/Cat.	Cycling	Hockey	Baseball	Auto	Soccer
Shutout	0	14	9	0	2
Rider	71	0	0	0	0
Europ	0	3	0	0	22
Nascar	0	0	0	43	0

Table 2. Example: Relative word distribution matrix [P(Ci|Wj)].

Word/Cat.	Cycling	Hockey	Baseball	Auto	Soccer
Shutout	0	0.56	0.36	0	0.08
Rider	1.0	0	0	0	0
Europ	0	0.12	0	0	0.88
Nascar	0	0	0	1.0	0

Table 3. Example: Relative category distribution matrix [P(Wi|Ci)].

Word/Cat.	Cycling	Hockey	Baseball	Auto	Soccer
Shutout	0	0.82	1.0	0	0.09
Rider	1.0	0	0	0	0
Europ	0	0.18	0	0	0.91
Nascar	0	0	0	1.0	0

Both relative word distribution [P(Ci|Wj)] probability and relative category distribution [P(Wi|Ci)] probability play important parts in major statistical feature selection measures such as Gini Text and Expected Cross Entropy. These are briefly introduced below.

2.1 Gini Text

In a recent work, Gini index theory was applied to text feature selection and a new formula was constructed:

$$GT(W) = \sum_i P(W \mid C_i)^2 P(C_i \mid W)^2 \quad (1)$$

where *i* is the category number, P(W|Ci) is the probability of word W, given the occurrence of category *i* and P(Ci|W) is the probability of category *i* given the occurrence of word W [17].

2.2 Expected Cross Entropy

Expected cross entropy measure comes from information theory. It takes into account probability distribution of words over categories:

$$CET(W) = P(W)\sum_i P(C_i \mid W)\log\frac{P(C_i \mid W)}{P(C_i)} \quad (2)$$

where *i* is the category number, *P(W)* is the probability of word *W*, $P(C_i)$ is the probability of category *i* [12].

The two feature selection measures mentioned above score terms and rank them.

3. TERM WEIGHTING SCHEMES

In general, terms selected in the feature selection stage describe the content of the document to different extends. Thus, each term has to be assigned a weight to specify its level of significance in the document. Document classification can be achieved by using various weighting functions such as raw term frequency tf, log(tf+1) to reduce the effects of large differences in frequencies, the product of term frequency and inverse document frequency tf-idf where idf = log(|D|/|Df|), |D| is the number of documents in the corpus and |Df| is the number of documents in which term occurs. tf-idf has remained as one of the simplest and strongest feature weighting schemes to date. tf-idf and its logarithmic and normalized versions are default choices in text categorization because of their simple formulation and good performance on a number of various data sets [11].

4. ROCCHIO TEXT CLASSIFIER

Rocchio classifier is a simple and efficient linear classifier [15]. Normalized document vectors of a given category and normalized document vectors of all other categories are summed up. The prototype vector of a category is computed as follows;

$$\vec{w}_{+} = \alpha \frac{1}{|+|}\sum_{i\in+}\vec{d}_i - \beta \frac{1}{|-|}\sum_{j\in-}\vec{d}_j \qquad (3)$$

where α and β are impact parameters. |+| is the number of documents in the given category and |−| is the number documents in other categories. Negative elements of the prototype vector are set to 0. To classify a document, the cosine between the prototype vector of each category and document vector is computed. The category with the highest cosine score is chosen. The advantage of Rocchio's algorithm is that it is fast in training and testing.

5. PROPOSED METHOD

The method proposed and studied in this paper is composed of several stages. Terms are ranked using one of the feature selection measures. Although the exact number of high-ranking terms is a grey area, terms that rank in the top 10% are labeled as high-ranking in this study. The topmost terms are assigned to set A and remaining high ranking terms are assigned to set B. Terms in the top 1%, 2%, and 3% were discretely assigned to set A and remaining lesser high-ranking terms were assigned to set B in the experiments. The topmost terms in the training documents are weighted using tf-idf. Prototype vectors of each category are computed using Rocchio.

The probabilistic neural network depicted in Figure 1 consists of C input units where C is the number of categories. Thus, relative category distribution of each word in set B is considered as an input vector. Each input unit is connected to the pattern units. Pattern units are comprised of topmost terms from set A. Each pattern unit is connected to the corresponding output unit. Each output unit is initialized by the corresponding pattern unit's tf-idf weight in the testing document. The connections from the input units to each pattern unit represent weights. Those weights are acquired from the relative category distribution of each pattern unit. The number of output units is the same as the number of topmost terms. The number of input units is the same as the number of categories in a particular domain. Each pattern unit emits the inner product of its normalized weight vector (normalized relative category distribution) and the normalized input vector to form wTx where T stands for transpose operator. If wTx is greater than or equal to a threshold value, it adds the tf-idf value of the input vector (lesser high-ranking term) to the corresponding output unit. The process is repeated for each of the lesser high-ranking terms in the new document. Each input vector can contribute to zero, one or more than one output unit. At the end of re-weighting, a new document vector is obtained. In the network, a single pass through the pattern units (topmost terms) is sufficient. This procedure is repeated for each new document. It should be noted that the amount of memory for the PNN depends on the number of classes (|C|) and the number of topmost terms (|A|).

The cosine between the previously computed prototype vector of each category and the new document vector is then computed. The category with the highest cosine score is assigned to the document.

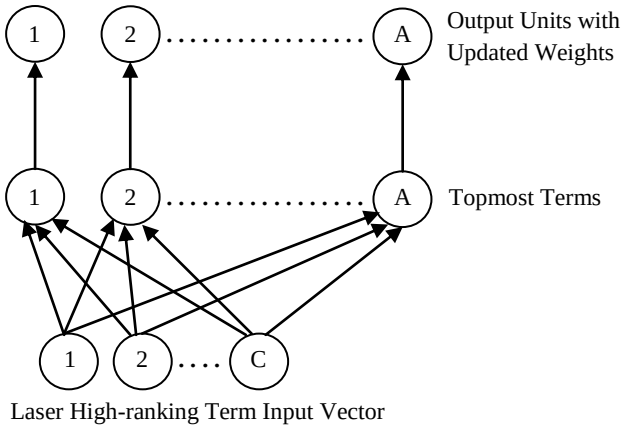

Fig 1: PNN Term Re-weighting Scheme

6. EXPERIMENTS

Reuters news collection RCV 1 [10] is comprised of 806,791 news articles between years 1996 and 1997. Each document may have more than one topic code depending on the material it covers. We formed three datasets with different characteristics from this corpus. We attempted to choose categories with high number of overlapping topic codes as Dataset 1 (3970 documents) and Dataset 2 (5133 documents) as seen in Tables 4 and 5 so that they would normally cause to produce relatively low classification accuracies. Dataset 3 (4052 documents) as summarized in Table 6 had low number of overlapping topic codes and thus higher classification accuracies may be expected. Each dataset had 11 categories.

Table 4. Dataset 1 selection from Reuters news collection RCV1.

Cat.#	Topic Codes	# of Docs
1	E11/E12/ECAT	715
2	E12/E13/E131/ECAT	602
3	E21/E211/E212/ECAT	492
4	E51/E511/E512/ECAT	398
5	E12/E21/E211/ECAT	370
6	E12/E21/E212/ECAT	360
7	E21/E212/E51/ECAT	330
8	E11/E13/E131/ECAT	199
9	E12/E51/E512/ECAT	176
10	E11/E21/E211/ECAT	165
11	E12/E51/ECAT	163

Table 5. Dataset 2 selection from Reuters news collection RCV1.

Cat.#	Topic Codes	# of Docs
1	C31/C311/CCAT/M14/M141/MCAT	715
2	M12/M13/M131/M132/MCAT	602
3	C31/CCAT/M14/M143/MCAT	492
4	C21/CCAT/M14/M142/MCAT	398
5	C31/CCAT/M14/M141/MCAT	370
6	M12/M13/M132/MCAT	360
7	C24/CCAT/M14/MCAT	330
8	M14/M141/M142/M143/MCAT	199
9	C24/CCAT/M14/M141/MCAT	176
10	C31/CCAT/M14/M142/MCAT	165
11	C24/CCAT/GCAT/GWEA/ M14/M141/MCAT	163

Table 6. Dataset 3 selection from Reuters news collection RCV1.

Cat.#	Topic Codes	# of Docs
1	C17/C171/C18/C183/CCAT	473
2	M14/M141/M142/M143/MCAT	402
3	E51/E511/E512/ECAT	398
4	C13/C33/CCAT	395
5	C31/C311/CCAT	379
6	E12/E21/E211/ECAT	370
7	C11/C41/C411/CCAT	368
8	GCAT/GENT/GPRO	353
9	M11/M13/M132/MCAT	311
10	GCAT/GDEF/GDIS	303
11	GCAT/GPOL/GREL	300

These datasets were divided into two equal-sized parts based on publication dates for training and testing. After collecting all the words in the training documents, the stop words were removed. Then, Porter stemming algorithm was applied to the remaining words [14]. The word frequencies in each category were used to compute the probability functions p(ci), p(wj), p(wj|ci), and p(ci|wj) for the feature selection measures. The statistical feature selection measures GT and CET were used to rank features. α was set to 16 and β was set to 4 for Rocchio classifier in this study as suggested by previous work [4]. The threshold value was set to 1 in the probabilistic neural network. As suggested by Lee 2007 [9], a flat prior is appealing as it allows the treatment of all class predictions equivalently for classification.

The results obtained using PNN-Rocchio and Rocchio classifiers at extreme term filtering ranges from 1% to 3 % are plotted in Figures 2 & 3 for Dataset 1; Figures 4 & 5 display them for Dataset 2; and, Figures 6 &7 for Dataset 3. In each couple of graphs, the first is for when using GT statistical feature selection measure and the other using CET.

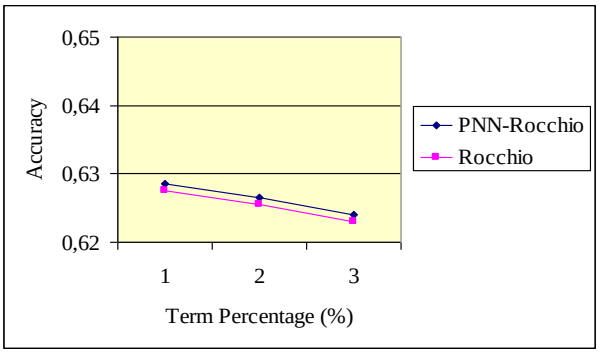

Fig 2: Gini Index accuracy rates for Dataset 1

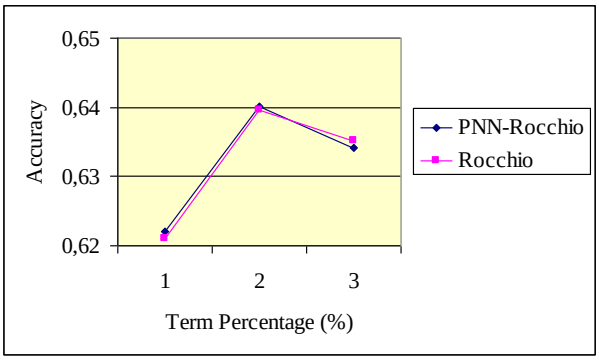

Fig 3: Cross Entropy accuracy rates for Dataset 1

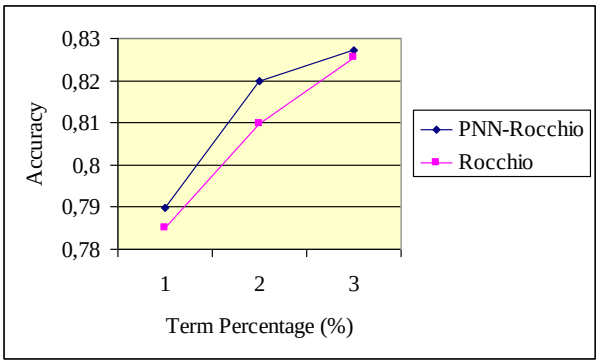

Fig 4: Gini Index accuracy rates for Dataset 2

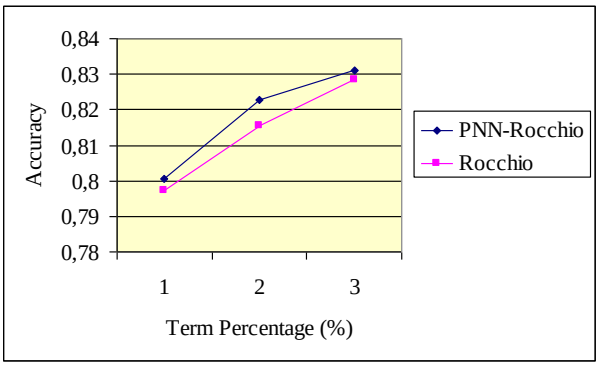

Fig 5: Cross Entropy accuracy rates for Dataset 2

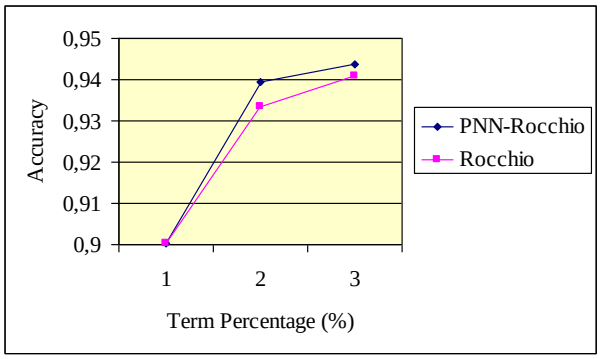

Fig 6: Gini Index accuracy rates for Dataset 3

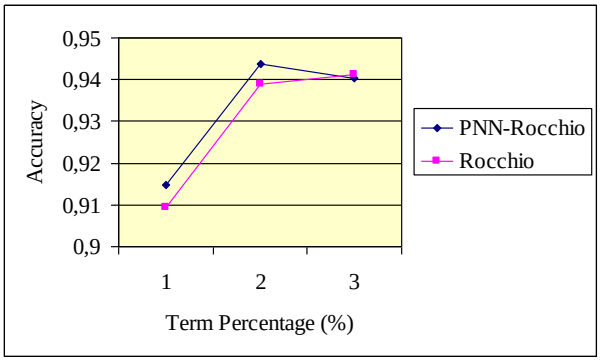

Fig 7: Cross Entropy accuracy rates for Dataset 3

In the cases of all datasets, both GT and CET terms at extreme filtering produce improved results for the most part employing PNN-Rocchio. For Dataset 1, although accuracies are low, our method gives a small boost to the existing classifier as seen in Figures 2 and 3; this result may as well be seen as confirmation of the fact that all that was possible to discover was already covered so there was no room for boosting. For Dataset 3, our method enhances already high accuracies as seen in Figures 6 and 7. In the cases of Dataset 2, our method, on the whole, boosts the accuracies compared to those in Dataset 1 and 3 as seen in Figures 4 and 5. This indicates that the use of PNN-Rocchio for datasets with moderate accuracies can be rewarding.

7. CONCLUSIONS AND FUTURE WORK

This paper shows that re-weighting of topmost terms conditioned by lesser high-ranking terms achieves better accuracy at extreme term filtering ranges. This method, employing two known feature selection algorithms, tf-idf term weighting scheme and probabilistic neural networks with Rocchio text classifier, performed marginally better.

In this paper we evolved and experimented with a new approach in order to discover possibly hidden cluster structures by employing PNN for boosting classification. We hope that this will be taken as a worthwhile example as called for by Kettenring 2006 [7].

For follow up work, the number of high-ranking terms may be increased thus including other words to the re-weighting process. This is not expected to be productive as the newly inducted words would likely have much lesser influence on classification accuracy. Term weighting schemes other than tf-idf may as well be considered. Furthermore, the effect of term re-weighting using PNN on accuracies of other text classifiers is yet to be studied. Likewise, the threshold of PNN may be adjusted. Indeed, for parameters are difficult or impossible to interpret, this inherent uncertainty plays into quantification of a coherent prior [9]. One may look into choices of different classes of priors for a fully Bayesian analysis.

8. ACKNOWLEDGEMENT

Zafer Erenel's contribution throughout this work is gratefully acknowledged.

9. REFERENCES

[1] Baker, L.D. and McCallum, A.K. (1998), "Distributional Clustering of Words for Text Classification", *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne, Australia, 96-103.

[2] Barzilay, R. and Elhadad, M. (1997), "Using Lexical Chains for Text Summarization", *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, Madrid, Spain, 10-17.

[3] Bekkerman, R., El-Yaniv, R., Tishby, N., and Winter, Y. (2003), "Distributional Word Clusters vs. Words for Text Categorization", *Journal of Machine Learning Research, 3*, 1183-1208.

[4] Buckley, C., Salton, G., and Allan, J. (1994), "The Effect of Adding Relevance Information in a Relevance Feedback Environment", *Proceedings of the 17th Annual International ACM-SIGIR Conference*, Dublin, Ireland, 293-300.

[5] Granitzer, A. and Auer, P. (2005), "Experiments with Hierarchical Text Classification", *Proceedings of the Artificial Intelligence Soft Computing (ASC 2005)*, Ed. del POBIL, A. P., Benidorm, Spain, *481*, 57-62.

[6] Kang, B.Y. and Lee, S.J. (2005), "Document Indexing: A Concept Based Approach to Term Weight Estimation", *Information Processing and Management, 41*, 1065-1080.

[7] Kettenring, J.R. (2006), "The Practice of Cluster Analysis", *Journal of Classification, 23*, 3-30, DOI: 10.1007/s00357-006-0002-6

[8] Kyriakopoulou, A. (2008), "Text Classification Aided by Clustering: A Literature Review", *Tools in Artificial Intelligence*, Ed. FRITZSCHE, P., Austria: In Tech, 233-252.

[9] Lee, H.K.H. (2007), "Default Priors for Neural Network Classification", *Journal of Classification, 24*, 53-70, DOI: 10.1007/s00357-007-0001-2

[10] Lewis, D.D., Yang, Y., Rose, T.G., and Li, F. (2004), "RCV1: A New Benchmark Collection for Text Categorization Research", *Journal of Machine Learning Research, 5*, 361-397.

[11] Liu, Y., Loh, H.T., and Sun, A. (2009), "Imbalanced Text Classification: A Term Weighting Approach", *Expert Systems with Applications, 36*, 690-701.

[12] Mladenic, D. and Grobelnik, M. (2003), "Feature Selection on Hierarchy of Web Documents", *Decision Support Systems, 35*, 45-87.

[13] Morris, J. and Hirst, G. (1991), "Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text", *Computational Linguistics, 17*, 21-48.

[14] Porter, M.F. (1980), "An Algorithm for Suffix Stripping", *Program, 14*, 130-137.

[15] Rocchio, J.J. (1971), "The SMART Retrieval System: Experiments in Automatic Document Processing". In Relevance Feedback in Information Retrieval, Ed. Salton, G., Englewood Cliffs, NJ: Prentice-Hall, 313-323.

[16] Sebastiani, F. (2002), "Machine Learning in Automated Text Categorization", ACM Computing Surveys, 34, 1–47.

[17] Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., and Wang, Z. (2007), "A Novel Feature Selection Algorithm for Text Categorization", *Expert Systems with Applications, 33*, 1-5.

[18] Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E.G.M., and Millios, E.E. (2005), "Semantic Similarity Methods in WordNet and their Application to Information Retrieval on the Web", *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, Bremen, Germany, 10-16.

[19] Wu, T.F., Lin, C.J., and Weng, R.C. (2004), "Probability Estimates for Multi-Class Classification by Pairwise Coupling", *Journal of Machine Learning Research, 5*, 975-1005.