Performance Evaluation of Text Categorization Algorithms Using an Albanian Corpus

Evis Trandafili[1(✉)], Nelda Kote[2], and Marenglen Biba[3]

[1] Department of Computer Engineering, Faculty of Information Technology, Polytechnic University of Tirana, Tirana, Albania
etrandafili@fti.edu.al

[2] Department of Fundamentals of Computer Science, Faculty of Information Technology, Polytechnic University of Tirana, Tirana, Albania
nkote@fti.edu.al

[3] Department of Computer Science, Faculty of Information Technology, New York University of Tirana, Tirana, Albania
marenglenbiba@unyt.edu.al

Abstract. Text mining and natural language processing are gaining significant role in our daily life as information volumes increase steadily. Most of the digital information is unstructured in the form of raw text. While for several languages there is extensive research on mining and language processing, much less work has been performed for other languages. In this paper we aim to evaluate the performance of some of the most important text classification algorithms over a corpus composed of Albanian texts. After applying natural language preprocessing steps, we apply several algorithms such as Simple Logistics, Naïve Bayes, k-Nearest Neighbor, Decision Trees, Random Forest, Support Vector Machines and Neural Networks. The experiments show that Naïve Bayes and Support Vector Machines perform best in classifying Albanian corpuses. Furthermore, Simple Logistics algorithm also shows good results.

1 Introduction

The digital word is expanding not only with data that users create themselves, but even with data created about these users. A study conducted by IDC stated that from now until 2020, the digital universe will double every two years [1]. The increase of accessible textual data has caused a flood of information instead of providing knowledge. In this situation there is an urgent need to explore and upgrade Text Mining algorithms and design new methods to exploit this avalanche of text.

Furthermore, we have to consider that most of the digital information is composed by unstructured text data; as a result the process of knowledge discovery and analysis is becoming an issue. The aim of Data Mining techniques and methods is to extract patterns and/or analyze databases, structured, well organized data. However, text is unstructured as it is based on language syntax and structure and therefore much more difficult to handle.

L. Barolli et al. (Eds.): EIDWT 2018, LNDECT 17, pp. 537–547, 2018.
https://doi.org/10.1007/978-3-319-75928-9_48

Text Mining is similar to Data Mining, but works with unstructured or semi-structured data sets (such as full-text documents or HTML files). Starting with a collection of documents, a text-mining tool retrieves a particular document and preprocesses it by checking its format and character sets. It then goes through text analysis, sometimes repeating techniques until the targeted information is extracted [2].

For this paper, we created a text corpus in Albanian language by collecting information from different online portals and grouping the documents in twenty categories. This corpus can also be used for future experiments and other purposes in Text Mining. As a first step, the corpus is passed through a language dependent preprocessing task composed of stop-word removal and stemming providing 'cleaned' datasets. On the output dataset is performed the training and testing of the algorithms. Since text categorization improves the organization level of the corpus, we focused our work on the evaluation of the performance of text classification algorithms. The classification problems are used in different domains of data mining and information retrieval and are implemented in publicly available software systems. We will evaluate the performance of the following classification algorithms: Naïve Bayes, Logistic Regression, k-Nearest Neighbor, Decision Trees, Random Forest, Support Vector Machines and Neural Networks. As studied on [3], the preprocessing task provides significant improvement on classification accuracy depending on the domain and language. Under this point of view, we can assume that the results of our work may not be the same if the corpus used is not composed of Albanian text documents and a different language is used.

The organization of the paper is as follows: Sect. 2 presents the background and the related work on text preprocessing and classification; Sect. 3 presents some basic information about the Albanian language structure; Sect. 4 presents the structure of the corpus and the preprocessing steps applied to it; Sect. 5 presents and analyzes the experiments; Sect. 6 analyses the classification algorithms taken in consideration; and we conclude in Sect. 7 with conclusions and future work.

2 Background and Related Works

The leading Text Mining approaches are listed by [4] such as: Information Retrieval, Natural Language Processing, Information Extraction from text, Text Summarization, Unsupervised Learning Methods, Supervised Learning Methods, Probabilistic Methods for Text Mining, Text Streams and Social Media Mining, Opinion Mining, Sentiment Analysis and Biomedical Text Mining. The process of selecting the appropriate technique optimizes the efforts of extracting the most valuable information [5].

The main technologies for Text Classification are supervised, semi-supervised, and unsupervised approaches. Supervised learning and semi-supervised learning are broadly used for text classification, while unsupervised learning is mainly used for clustering. Some studies show that a hybrid method which combines supervised and unsupervised methods outperforms the supervised support vector machine (SVM) in terms of both F1 performance and classification accuracy [6].

The Albanian language has not been much explored from the perspective of Natural Language Processing and Computational Linguistics. There are some trivial works

previously conducted by [7] who proposed a rule-based stemmer for Albanian language and [8] who enhanced the previous stemmer by supporting the composite words. The performance of the composite stemmer was tested by using text classification algorithms and showing that preprocessing the document with the stemmer of composite words significantly enhances the performance of the classifier.

Text classification algorithms are evaluated on text corpuses of different languages. For example, in [9] the authors tested some classification algorithms on Turkish written documents. Their experimental results estimated that the Random Forest classifier gives more accurate results than Naïve Bayes, Support Vector Machines, K-Nearest Neighbor and J48.

The authors in [10] evaluated the classification algorithms (decision trees, rule induction, naive Bayes, neural networks and support vector machines) for n-gram collocations in Croatian language and concluded that the best classifier for bigrams was SVM, while for trigrams the decision tree.

Another interesting work was conducted in [11] to compare the performance of Naïve Bayes and Support Vector Machines in literary domain. The algorithms were also combined with text pre-processing tools to study the impact on the classifiers' performance. NB and SVM achieved high accuracy in sentimental chapter classification, but the NB classifier outperformed the SVM classifier in erotic poem classification.

Furthermore, in [12] the authors investigated the preprocessing techniques that impact the performance of Support Vector Machines classification algorithm in English and European Portuguese languages. They treated the document representation as an optimization problem in terms of feature reduction, selection and term weighting.

Another text classification comparison is conducted by [13] to evaluate the performance of K-Nearest Neighbor and Naïve Bayes. The assessment is done on a corpus of XML documents and the optimal value of $k = 13$ that yield the best performance for K-NN was identified.

There are some other works which focus on Arabic languages. And interesting case study is conducted by [14] using an Arabic corpus and demonstrating that using an Artificial Neural Network model is effective in capturing the non-linear relationships between document vectors and document categories if used with feature reduction methods.

A novel approach is the exploitation of text classification methods for multilingual language classification with the use of Convolutional Neural Networks. The work carried out by [15] showed that the classifier does not require syntactic and semantic knowledge of the language and performs well even on new languages.

3 Albanian Language Structure

The Albanian language is considered the modern survivor of the Indo-European language family, mostly spoken in Albania, Kosovo and in other parts of the Balkans. Dacian and Illyrian have been considered its ancestors of ancient languages. There are two main dialects Gheg, spoken in the north, and Tosk, spoken in the south, which by now have been diverging to their most extreme and diverse forms. The official Albanian

language is written in Roman alphabet and from 1909 till World War II was based on the south Gheg dialect. Since then it has been modeled on Tosk dialect [16]. For the purpose of this paper, the official Albanian language is used.

The official Albanian language has 7 vowels and 29 consonants. The vowels are represented by single Latin letters (*a, e, ë, i, o, u, y*), and the consonants by single letters (*b, c, ç, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, x, z*), and combination of different letters (*dh, gj, ll, nj, rr, sh, th, xh, zh*).

There are some words in Albanian which do not carry any meaning. These are the stop-words which for the purpose of this paper are identified and removed from the corpus. Some of the most useful stop-words in Albanian are: *"dhe", "sepse", "kur", "edhe", "në", "prej", "apo", "ose"*, etc.

The main grammatical categories of Albanian are: nouns which show gender, number, case and are inflected with suffixes to show definite or indefinite meaning, e.g. *tavolinë – "table", tavolina – "the table"*. A large number of noun plurals have irregular stem formation; Adjectives follow the noun and are preceded by a particle that agrees with the noun, e.g. in *një njeri i fortë*, "a strong man," *burrë* "man" is modified by the adjective *fortë* "strong," preceded by *i*, which agrees with the noun "man"; verbs have a great variety of forms and are quite irregular in forming their stems. As a conclusion, the grammar and formal distinctions of Albanian are inherited by the Romance languages and of Modern Greek [16].

4 Data Collection and Preprocessing

It is difficult to work with text processing algorithms with Albanian texts because it doesn't exist a formal corpus where you can rely. So, as part of our work we had to create a text corpus of Albanian written documents. We collected text data regarding 20 domains as follows: Animals, Art, Astronomy, Biology, Charity, Chemistry, Culture, Curiosities, Economy, Environment, Fashion, Food, History, Literature, Medicine, Politics, Religion, Sport, Technology and Tourism. Each category has 40 documents made up by textual information chosen randomly on the web, respectively from the fields chosen previously.

Before running the experiments, the corpus of documents was passed through a preprocessing phase which consists of the tokenization, stop-word removal and the stemming algorithm [7]. The aforementioned algorithms are implemented in java programming language and the whole implementation is based on different rules comprising the Albanian language structure. After the preprocessing step the text files look like a bag of word and do not have language structure any more. This structure is then used as input for text classification algorithms. There are a variety of publicly available software systems which implement different machine learning algorithms and data mining tasks like WEKA [17] and MALLET [18]. For the purpose of this paper we chose the Weka software. For our text documents to be classified by Weka we converted the file format from .txt to .arff which is an ASCII text file that describes a list of instances sharing a set of attributes and stands for Attribute-Relation File Format [17]. For this purpose we used the textDirectoryLoader class implemented in Weka. Then the file was

passed through the StringToWordVector filter which transforms all the string attributes into a vector that represents the word occurrence information from the text in strings.

5 Classifier Selection

Text classification is the process of assigning predefined categories to text documents. The classification problem is defined as follows: on a training set of documents, $D = \{d_1, d_2, \ldots d_n\}$, such that each document d_i is labeled with l_i from the set of categories $L = \{l_1, l_2, \ldots, l_k\}$. The goal is to find a classification function (classifier) f where $f(d) = l$ which assigns the correct category label to a new document d not previously used for training [4]. There are different methods for the classification task which are applied in domains such as quantitative or categorical data. Text data is modeled as quantitative data regarding frequencies and word attributes so most of the classification methods can be applied directly on text [19].

The classification techniques are divided in five main categories: Regression, Distance, Decision Trees, Rules and Neural Networks [20]. In order to handle text classification in breadth we selected five key classifiers, one for each classification technique, respectively: Logistic Regression, K-Nearest Neighbor, C4.5, Naïve Bayes and Back Propagation. Furthermore, Support Vector Machines (SMO) and Random Forest algorithms are also included in our experiments due to their popularity and the results obtained in similar works.

5.1 Logistic Regression

Logistic regression is a statistical machine learning algorithm that uses a logistic function, also called sigmoid function, to compute the probability for each class and then choosing the class with the maximum probability. It is considered a linear model because it assumes a linear additive relationship between the predictors and the log odds of a classifier. A key difference with the linear regression is that the output value being modeled is a binary value rather than a numeric value [21]. For the purpose of this paper the Simple Logistic algorithm in Weka is used.

5.2 Naïve Bayes

The Naive Bayes classifier is the most popular among generative classifiers and as the name suggests is based on Bayes rules. The algorithm computes the posterior probability of a class, based on the distribution of the words in the document by ignoring the actual position of the words in the document, and working with the "bag of words" assumption [19]. Despite the simplicity of this algorithm, it is fast and does not have big storage requirements.

5.3 K-Nearest Neighbor

The k-Nearest Neighbor classifier algorithm compares new items with all members in the training set based on the distance of the k most similar neighbors to predict the class of the new unlabeled document, X. The classes of the neighbors are weighted using the similarity of each neighbor to X. The similarity is measured by Euclidean distance or the cosine value between two document vectors. KNN does not rely on prior probabilities, since the main computation is the sorting of training documents in order to find the k nearest neighbors for the new document. It is computationally expensive to find the k nearest neighbor in high dimensions. KNN is implemented in Weka as IBk [17], (Instance Based Learner).

5.4 Decision Tree (C4.5)

C4.5 belongs to the category of statistical classifiers. It is an improvement of ID3 classification algorithm. This algorithm creates a decision tree by using the entropy to determine which attribute of a given instance will optimize the classification of the instances in the dataset and which values of these ranges will provide the best classifying results. Rules can be generated for each path in the tree. After building the tree from a training dataset, the algorithm receives new data and classifies it. In Weka this algorithm is implemented as J48 [17].

5.5 Neural Network with Back Propagation

Backpropagation is an algorithm based on supervised learning for training an Artificial Neural Network Classifier, ANN. Backpropagation is very efficient in recognizing complex patterns and performing nontrivial mapping functions. During the training phase, the connection weights of the neural network are given randomly initialized values. These training examples are then delegated to the ANN classifier which adjusts the connection weights using the back propagation algorithm. The procedure is repeated until the desired learning error is reached [22]. Multilayer Perceptron is the implementation of the Back Propagation algorithm in Weka software.

5.6 Support Vector Machines

Support vector Machines is a classifier based on statistical information theory and structural risk minimization. Sequential minimal optimization, SMO, is used for training a support vector classifier in weka. SMO breaks the quadratic programming problem into small quadratic programming problems and solves them analytically. The memory required by SMO is linear in the training set size; thereby the algorithm can handle large training sets. SMO is fastest for linear SVMs and sparse data sets. On real world sparse data sets, SMO can be more than 1000 times faster than the chunking algorithm [23].

5.7 Random Forest

Random forest algorithm creates the forest from a set of decision trees, each created by selecting random subsets of training data. The final class of the new object is assigned to the class with the highest value and is achieved as an outcome of all trees in the forest. Tree ensembles are a divide-and-conquer approach used to improve the performance. Random inputs and features yield good results in classification, run fast and are able to deal with missing data, but it is less beneficial in regression. It can't predict beyond the training range, resulting in an over fit in noisy data sets [24].

6 Experiments

In order to perform the classification experiments, a corpus of 20 different categories, each with 40 text Albanian documents is created. The classification can be affected by the type of categories and the similarity between them. For this purpose we created sub corpuses of different sizes and content of categories.

Furthermore, we also created corpuses of the same sizes and number of categories, differing from category names and text content inside the documents. We expect the same algorithm to slightly vary in performance based on the content of documents and type of class used. All the classification experiments are run on the corpuses listed in Table 1.

Table 1. Experimental corpus

Corpus code	Interpretation
C1	Corpus of 20 categories (Animals, Art, Astronomy, Biology, Charity, Chemistry, Culture, Curiosities, Economy, Environment, Fashion, Food, History, Literature, Medicine, Politics, Religion, Sport, Technology, Tourism) each with 20 documents.
C2.1	Corpus of 10 categories (Animals, Charity, Environment, Fashion, Food, Medicine, Politics, Religion, Technology, Tourism) each with 20 documents.
C2.2	Corpus of 10 categories (Art, astronomy, chemistry, economy, Food, literature, Politics, sport, Technology, Tourism) each with 20 documents.
C3	Corpus of 10 categories (astronomy, biology, chemistry, culture, curiosities, economy, history, literature, sport) each with 40 documents.
C4.1	Corpus of 5 categories (Environment, Fashion, Medicine, Technology, Tourism) each with 20 documents.
C4.2	Corpus of 5 categories (Animals, Charity, Fashion, Politics, sport) each with 20 documents.
C5.1	Corpus of 3 categories (chemistry, sport, Tourism) each with 20 documents.
C5.2	Corpus of 3 categories (Art, Food, literature) each with 20 documents.
C6.1	Corpus of 2 categories (Medicine, Tourism) each with 20 documents
C6.2	Corpus of 2 categories (Charity, Technology) each with 20 documents.

Each corpus is used for training a model and testing the chosen classification algorithms: Naïve Bayes, IBk, J48, SMO, Random Forest, Simple Logistic and Multilayer Perceptron. The Table 2 shown below gives the results of the experiments. We have highlighted in red the best percentage of correctly classified instances of each corpus.

Table 2. Experimental results with the classification accuracy for each algorithm.

Algorithm	C1	C2.1	C2.2	C3	C4.1	C4.2	C5.1	C5.2	C6.1	C6.2
Simple Logistic	64%	76%	76%	61%	81%	87%	96%	81%	100%	85%
Naïve Bayes	66%	82%	77%	57%	89%	93%	98%	81%	97%	87%
K-Nearest Neighbor (IBk)	18%	36%	21%	30%	52%	24%	41%	55%	85%	85%
Decision Tree (J48)	43%	52%	50%	55%	65%	67%	95%	72%	87%	80%
SVM (SMO)	65%	78%	78%	60%	91%	82%	93%	93%	97%	92%
Random Forest	58%	67%	69%	58%	77%	77%	95%	88%	92%	78%
ANN (Multilayer Perceptron)	5%	9%	9%	12%	19%	19%	48%	33%	82%	85%

To rate the algorithm from the best performant to the least performant, for each corpus we assessed each algorithm with a score from 6 to 0 based on the percent of correctly classified instances. For example, corpus C1 performs best with Naïve Bayes algorithm achieving 66% of correctly classified instances, so we rate this experiment with 6 points. The second best performant algorithm for C1 is SMO with 65% of correctly classified instances, so we rate SMO with 5 points. Next comes Simple Logistic with 64% scoring 4 points; Random Forest with 58% correctly classified instances scores 3 points; J48 scores 2 points with 43% correctly classified instances; IBk scores 1 point with 18% correctly classified instances and Multilayer Perceptron scores 0 points as the less performant of all. An equivalent scoring scheme is applied to every corpus listed in Table 1 and the result is shown in Table 3. The last column, Total, is calculated summing the scores in the rows and is considered as the score achieved by the algorithm.

Table 3. Algorithms evaluation scheme

Algorithm	C1	C2.1	C2.2	C3	C4.1	C4.2	C5.1	C5.2	C6.1	C6.2	Total
Simple Logistic	4	4	4	6	4	5	5	4	6	4	46
Naïve Bayes	6	6	5	3	5	6	6	4	5	5	**51**
K-Nearest Neighbor (IBk)	1	1	1	1	1	1	1	2	2	4	15
Decision Tree (J48)	2	2	2	2	2	2	4	3	3	3	25
SVM (SMO)	5	5	6	5	6	4	3	6	5	6	**51**
Random Forest	3	3	3	4	3	3	4	5	4	2	34
ANN (Multilayer Perceptron)	0	0	0	0	0	0	2	1	1	4	8

From the table shown below we can see that the best performing algorithms on our Albanian corpus are Naïve Bayes and Support Vector Machines, scoring both 51 points. Simple Logistics is the next best choice, and then comes Random Forest, J48, IBk and the worst performance is achieved by Multilayer Perceptron.

Based on the above results, we decided to evaluate the best performing algorithms to show if they have any statistical differences with one another. For this purpose we used the Experimenter in Weka for each dataset with Naïve Bayes, Support Vector Machines and Simple Logistics using the *Percent_correct* as comparison field. In this way we compared the percent of correctly classified instances of Naïve Bayes with Support Vector Machines and Simple Logistics. The summarized results are shown in the following Table 4.

Table 4. Results of the statistical evaluation

Corpus	*Naïve Bayes*	*SVM (SMO)*	*Simple Logistic*
C1	66.49	64.33	64.16
C2.1	81.86	78.60	76.67
C2.2	76.40	76.73	76.69
C3	57.24	59.43	61.20
C4.1	88.31	88.85	81.27
C4.2	92.07	83.22 *	86.18
C5.1	99.17	93.33	96.50
C5.2	81.83	92.33 v	87.83
C6.1	98.50	98.25	97.25
C6.2	90.40	92.15	82.60

The statistical comparison showed no significant differences except for corpuses C4.2 and C5.2. When the comparison is run on corpus C4.2, the result for SMO has a "*" next to it, meaning that SMO is statistically different with Naïve Bayes and the later performs better. From the other side, when the comparison is run on corpus C5.2, the result for SMO has a "v" next to it, meaning that SMO statistically outperforms Naïve Bayes. As a conclusion, we cannot determine statistically the best algorithm among Naïve Bayes, Support Vector Machines and Simple Logistics based on the percent of correctly classified instances.

7 Conclusions and Future Work

The main goal of this paper was the overall comparison of performance of text classification algorithms for Albanian language. For this purpose we reviewed the state of the art for text classification problems in different languages. Since there isn't any public corpus previously created for Albanian, we created a general corpus of 20 classes, each with 40 text documents and divided it in 10 different sets appropriate for our experiments. The corpus was preprocessed with stop words removal and JStem algorithm. We used Weka software to test Naïve Bayes, IBk, J48, SMO, Random Forest, Simple Logistic and Multilayer Perceptron algorithms. For each algorithm we tested the performance on

10 sets of documents. The best performing algorithms on our Albanian corpus are Naïve Bayes and Support Vector Machines both with the same score. From an overall projection of the results we can say that in general all the algorithms perform quiet well in classification problems when the number of classes is relatively small (2 or 3).

As a future work, it is of great interest to measure the effect of preprocessing phase on the performance of text classification. The stemming phase in Albanian language is a rule based algorithm that needs further improvements, and we believe that this will also improve the overall performance of the classification algorithms.

Furthermore, a public bigger corpus for Albanian needs to be created for further experiments in Natural Language Processing and Text Mining. The same algorithms ca also be compared using another bigger corpus.

References

1. Gantz, J., Reinsel, D.: The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. Technical Report 1. IDC, 5 Speen Street, Framingham, MA 01701 USA (2012)
2. Fan, W., Wallace, L., Rich, S., Zhang, Z.: Tapping the power of text mining. Commun. ACM **49**(9), 76–82 (2006)
3. Uysal, A.K., Gunal, S.: The impact of preprocessing on text classification. Inf. Process. Manage. **50**, 104–112 (2014)
4. Allahyari, M., et al.: A brief survey of text mining: classification, clustering and extraction techniques. In: Proceedings of KDD Bigdas, Halifax, Canada, 13 p., August 2017
5. Talib, R., et al.: Text mining: techniques, applications and issues. Int. J. Adv. Comput. Sci. Appl. **7**(11) (2016)
6. Zewen, X.U., et al.: Semi-Supervised Learning in Large Scale Text Categorization. Shanghai Jiao Tong University and Springer, Heidelberg (2017)
7. Sadiku, J., Biba, M.: Automatic stemming of Albanian through a rule-based approach. J. Int. Res. Publ. Lang. Individ. Soc. **6** (2012). ISSN 1313-2547
8. Biba, M., Gjati, E.: Boosting text classification through stemming of composite words. In: ISI 2013, pp. 185–194 (2013)
9. Kılıncx, D., et al.: TTC-3600: a new benchmark dataset for Turkish text categorization. J. Inf. Sci., 1–12 (2015)
10. Karan, K., Snajder, J., Basic, B.D.: Evaluation of classification algorithms and features for collocation extraction in Croatian. In: LREC 2012, Eighth International Conference on Language Resources and Evaluation (2012). ISBN 978-2-9517408-7-7
11. Yu, B.: An evaluation of text classification methods for literary study. Literary Linguist. Comput. **23**(3), 327–343 (2008)
12. Gonçalves, T., Quaresma, P.: Using IR techniques to improve automated text classification. In: Meziane, F., Métais, E. (eds.) Natural Language Processing and Information Systems, NLDB 2004. LNCS, vol. 3136. Springer, Heidelberg (2004)
13. Rasjida, Z.E., Setiawan, R.: Performance comparison and optimization of text document classification using k-NN and Naïve Bayes classification technique. In: 2nd International Conference on Computer Science and Computational Intelligence 2017, ICCSCI 2017, vol. 1314, Bali, Indonesia, October 2017

14. Al-Zaghoul, F., Al-Dhaheri, S.: Arabic text classification based on features reduction using artificial neural networks. In: UKSim 15th International Conference on Computer Modelling and Simulation. IEEE (2013)
15. Zaid Enweiji, M., Lehinevych, T., Glybovets, A.: Cross-language text classification with convolutional neural networks from scratch. Eureka: Phys. Eng., 24–33 (2017). https://doi.org/10.21303/2461-4262.2017.00304
16. Hamp, E.P.: Albanian Language, Encyclopedia Britannica (2016)
17. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann (2005)
18. McCallum, A.K.: Mallet: A Machine Learning for Language Toolkit (2002)
19. Aggarwal, C., Zhai, C.X.: Mining Text Data. Springer (2012)
20. Dunham, M.H.: Data Mining: Introductory And Advanced Topics. Pearson Education (2006)
21. Moreaux, M.: Text Classification with Generic Logistic-Regression Classifier (2015)
22. Ramasundaram, S., Victor, S.P.: Text categorization by backpropagation network. Int. J. Comput. Appl. (0975 – 8887) **8**(6), October 2010
23. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning (1998)
24. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)