Winter 1-28-2021

Delineating Knowledge Domains in Scientific Domains in Scientific Literature using Machine Learning (ML)

Abhay Maurya
Mizoram University, abhaymaurya@mail.mzu.edu.in

Smarajit Paul Choudhury Mr.
Indian Institute of Technology, Benaras Hindu University, Varanasi, Uttar Pradesh, India,
smarajit.pc@gmail.com

Kshitij Jaiswal Mr.
Indian Institute of Technology, Benaras Hindu University, Varanasi, Uttar Pradesh, India,
kshitijjaiswal.min18@itbhu.ac.in

Delineating Knowledge Domains in Scientific Domains in Scientific Literature using Machine Learning (ML) Techniques

Abhay Maurya[1] *Smarajit Paul Choudhury*[2] *Kshitij Jaiswal*[3]

1 Research Scholar, Mizoram University, Mizoram, India
2, 3 Indian Institute of Technology, Benaras Hindu University, Varanasi, Uttar Pradesh, India

---

ABSTRACT

The recent years have witnessed an upsurge in the number of published documents. Organizations are showing an increased interest in text classification for effective use of the information. Manual procedures for text classification can be fruitful for a handful of documents, but the same lack in credibility when the number of documents increases besides being laborious and time-consuming. Text mining techniques facilitate assigning text strings to categories rendering the process of classification fast, accurate, and hence reliable. This paper classifies chemistry documents using machine learning and statistical methods. The procedure of text classification has been described in chronological order like data preparation followed by processing, transformation, and application of classification techniques culminating in the validation of the results.

Keywords

Text classification, text mining, random forest, support vector machines, naïve Bayes, xgboost

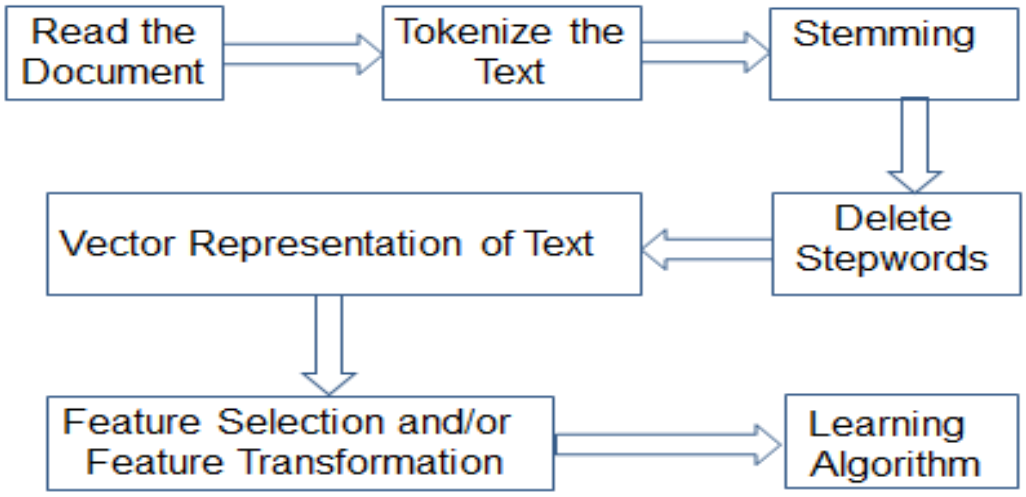
---

INTRODUCTION

Text classification or text categorization is the art of classifying a text into discrete groups. It is a complex process that involves the training of models besides certain additional processes which inter alia include processing of data, noise reduction, and transformation. Text classification is a topic of research dovetailing the latest techniques and their utility in complex systems. Researchers are also developing certain novel techniques for a better classification culminating in the yield of better results [1][2][3]. Since the advent of documents in digital form, text classification has been the most widely used application. Text classification has been necessitated due to a large number of digital text documents that confront us every day.

Text classification can be subdivided into topic-based and genre-based. While the former classifies documents as per their topics [4], the latter relies upon various genres like reviews, articles, etc., for classification of the documents [5]. The word genre implies the modus operandi of the creation of a document and the intended audience. Previous research on the subject draws a clear distinction between the two forms of classification [5]. Normally, data for classification is retrieved from a wide variety of sources and suffer from various impediments like the variation in formats, vocabulary, writing styles, etc., which give them a heterogeneous character.

From a technical perspective, if d_i is any document belonging to the entire set of documents D and $\{c_1, c_2, c_3, \ldots\ldots\ldots\ldots, c_n\}$ is the set of all categories, then the process of text classification would assign a category c_j to document d_i. Like all other machine learning processes, text classification also requires a preliminary set of documents. Any document in the set of documents may be assigned numerous categories but the scope of the present study tries to assign distinct categories. Numerous research indicates the domain classification of texts [6]. A graphical representation of the process of text classification is produced below:



The construction of a classifier is similar to several other machine learning problems sans representation of the document [7]. One peculiarity of text classification is the presence of a large number of features denying the use of sophisticated learning algorithms. In any text classification exercise, the removal of redundant features is a complex procedure.

This calls for the introduction of the procedure of dimension reduction entailing either selecting a subset from the set of original features [8] or computing new features from old ones [9]. Dimension reduction procedures involve feature extraction and feature selection. Feature extraction involves the extraction of features from the low dimensional feature space, like principal component analysis [10], and linear discriminant analysis [11]. There are two main models of feature selection: the filter and the wrapper [12]. While wrapper models generate new data sets using specific classifiers for selection and generation of features [13], filter models emphasize evaluation algorithms over classifiers [14]. Due to high efficiency and faster processing speeds, filter models find utility in the scaling of large data sets [15].

Chemistry is a branch of science that has a scope between physics and biology and deals with the structure, properties, and composition of matter. Chemistry helps in understanding the other branches of science like botany, pharmacology, geology, etc. The history of chemistry has been both challenging and interesting which has developed over the centuries through trial and error. The foundation of chemistry has been laid when Robert Boyle began his research which led to discovering the behaviour of gases. Boyle also put the results of his research in a mathematical form lending credibility to his findings. After the lapse of considerable time, Dalton put forward the atomic theory.

The scope of this study lies in the classification of documents on chemical research derived from Scopus into 10 classes and compare various machine learning algorithms to arrive at the best predictive model. The classification model has been built considering three important features which include title, abstract and initial keywords. This is followed by the data cleaning process which involves the removal of punctuation, splitting the text into individual words, stemming of split words, etc.

VECTOR SPACE DOCUMENT REPRESENTATION

Any document is a collection of different words arranged in sequential order [16]. So all the words present in any training set may be called vocabulary or feature set. So any document can be expressed as a binary vector assigning the value 1 if a particular word is present in the document or 0 if the word is absent from the document. This implies positioning the document in a space $R^{|V|}$ where $|V|$ denotes the size of the vocabulary V. All documents contain certain words that find no use training the classifier and are hence removed as a part of the pre-processing work. Such words are referred to as Stopwords [17]. Another common pre-processing task is stemming which entails the reduction in the size of the initial feature set by removing misspelled words etc., using a stemming algorithm. Stemming amplifies the performance of the classifiers though aggressive use of stemming is a matter of debate [18].

Feature engineering is defined as the representation of the value of a feature [19]. This value is the Boolean indicator of the sufficiency of the presence of any word in the document. Other definitions include the frequency of the presence of a word in the document normalized by the length of the document. Normalization of the count is vital for documents having varying lengths. However, in the case of short documents where the chances of repetition are minimized, Boolean indicators can prove beneficial. This step assumes importance in terms of lessening the time and cost of training the resources.

FEATURE SELECTION

The method of feature selection reduces the dimensionality of the dataset by removing features that are considered unnecessary for classification [20]. Besides decreasing the dimensionality of the dataset leading to a decrease in the cost of computing and increased accuracy, feature selection also reduces overfitting. The process makes use of the evaluation function for every word [21]. Feature selection involves either of the two different types of processes: Best Individual Features (BIF) which is based upon the frequency of terms in any document, odds ration, mutual information, the strength of the terms [20][21][22][23][24] and Sequential Forward Selection (SFS) which selects a word based on the criteria and adds new words till the total number of words reach the required number [25]. As opposed to BIF methods, SFS methods rely on the dependencies between the various words appearing in a document making the method more reliable in terms of results. However, the large cost of computation and the large size of vocabulary makes it redundant in the application. Although text classification using machine learning techniques are better in performance, its inefficiency can be seen while training large datasets.

To speed up the process, certain researchers propose a pruning exercise to fine-tune the Training data set [26]. The use of this method reduces the size of the Training dataset maintaining the level of performance close to that without pruning. Some studies have also gelled Feature Selection and Instance Selection for text classification with better results [27] using a two-step process. The first step selects features having a high precision thereby dropping those words that do not conform to any of the features, while the second step searches those features that predict the complement of the target class from the initial dataset together with selecting these additional features.

FEATURE TRANSFORMATION

Both feature selection and feature transformation serve the purpose of trimming the size of the feature dataset but with certain inherent differences [28]. Feature transformation does not discard words with lower weights rather compacts the words as per the feature

requirement. Principal Component Analysis aims at reducing the complexities involved in classification by decreasing the size of the feature dataset without compromising the accuracy of the result. Studies show that the accuracy of text classification by the use of standard KNN over Latent Semantic Indexing yields a better result besides being less costly in terms of the involved computation cost [29].

MACHINE LEARNING ALGORITHMS

After completion of the process of feature selection and transformation, the data can be presented in a form understood by ML algorithms. Several studies recommend various algorithms which differ in their approach to the problem. Despite the several approaches, automatic classification of texts lacks credibility and needs further research for improvement. Simplicity and effectiveness make Naïve Bayes the most widely used text classifier [30] though it does not model the text efficiently. Studies conducted by Schneider show that certain corrections can rectify the problems [31]. Various studies show that Bayesian multinet classifiers based on the tree-like Bayesian network can handle text classification of a hundred thousand variables speedily and maintaining a high level of accuracy [32].

In the realm of text classification, support vector machines can provide accurate results though the algorithm lacks good recall. Studies suggest that the recall can be improved by adjusting the threshold of the SVMs [33]. In another study wherein a fast decision tree algorithm was developed to deal with the sparsity of data, Johnson et. al. converted the decision tree into a rule set [34]. Improvement in KNN based text classification using certain well-established parameters have also been shown in certain studies [35]. The well-established parameters can be found out using various decision functions, k-values, etc.

Training a binary classifier involves the use of all documents whether relevant or irrelevant present in the training set. In case a large number of categories are allocated to a limited number of documents, the problem of imbalanced data persists which can be sorted using a cost-sensitive learning mechanism [36]. Certain authors have proposed the system of parallelizing and distribution of text classification which has enhanced both accuracy and time complexity [37]. Recent studies propose combining classifiers towards improving the performance of the classifiers. In this context, studies indicate that the use of a combination of classifiers can improve the accuracy of classification [38][39]. Studies conducted towards comparing the efficacy of the best individual classifier versus the combination of classifiers show that the combined method surpasses the individual classifiers [40]. Some studies also propose the use of algorithms to boost automatic text classification with favorable outcomes [41].

REVIEW OF EXTANT LITERATURE

The studies on the subject in the public domain which could be accessed are unanimous regarding the steps involved in the process of text classification: (a) pre-processing of the document, (b) modeling of the document, (c) feature selection, (d) construction of a classification model using machine learning algorithm, and (e) evaluation. Certain previous studies prescribe the following steps for the purpose: (a) pre-processing, (b) creation of a vector space model, (c) feature selection, (d) training of the Training dataset, and (e) determination of the performance [42]. In their study, the authors used several plans for feature weighting besides explaining three major feature selection methods and one feature projection method. The study also dealt with details six machine learning methods. This study has been reciprocated by other studies with a larger number of examples [43]. This study also commented upon the accuracy of the classification process and observed certain things related to the performance of the linear classifiers and prescribing solutions. The other studies reciprocating the same have bee conducted by T.S.Guzella and W.M.Caminhas [44] and Garcia Adeva and others [45]. While the study conducted by T.S.Guzella and W.M.Caminhas focused mainly on spam filtering together with giving a detailed comparison of the various spam filtering methods, the study conducted by Garcia Adeva and others deals with the elements of classification systems.

Recent studies suggest that the process of text classification involves a complex exercise than previously thought of and describe text classification as a six-step process involving (a) acquisition of data, (b) labeling of data, (c) feature construction, (d) feature selection, (e) training of the model, and finally (f) evaluation of the results [46]. It may be inferred from related literature that any training model may employ various algorithms for training a classification model into specific classes. Several machine learning algorithms can achieve the objectives with accurate results like ANN, KNN, Decision Tree, Rule-based classifiers, Naive Bayes, Selective Naive Bayes, and SVM, etc. [47].

METHODOLOGY

Environment Configuration: This study was conducted using an Intel(R) Core(TM) i5-7200U processor having a CPU clock rate of @ 2.50GHz and 2.70GHz and the main memory of 8.00 GB RAM.

Building a Data Frame: From the data extracted from scopus.com, a data frame has been built selecting 2000 most relevant research papers taken from each sub-category of chemistry: Analytical Chemistry, Biochemistry, Environmental Chemistry, Industrial Chemistry, Inorganic Chemistry, Organic Chemistry, Physical Chemistry, Polymer Chemistry, Theoretical Chemistry, and Thermochemistry.

For this study, we have followed a particular sequence of operations represented by

Importing the libraries Importing the dataset Data cleaning Feature Engineering Splitting the dataset into the Training set and the Test set Training various classification models on the Training set Result prediction Finding the accuracy and classification matrix.

Libraries Used: For this study, numpy, pandas, re and nltk libraries were imported.

Importing the dataset: The dataset for the research paper that has been created above was imported, and then the same was explored through necessary steps to get an insight of the various features in the available data.

Data Cleaning: From the available data, a text classification model was built using three major features including Title, Abstract, and Index Keywords. All the data obtained were subjected to the process of data cleaning which involved

- Removal of punctuation
- Removal of capitalization of words
- Splitting the texts into individual words
- Stemming the split words

Feature Engineering: In this step, we have converted the cleaned text documents into a matrix of token counts using the Bag of Words Model, which is regarded as the most common way to convert any text into vectors in any NLP. The BoW model applies a count vectorizer to the cleaned texts to create vectors out of the text. Each document is represented as a vector. Each vector can now be used as feature vectors for building a model. We also performed Label encoding on the Topic column to convert the categorical categories into numerical values by assigning a different integer to all the 10 subtopics.

Splitting the Dataset: In this step, we have split the data into two sets: the Training set and the Testing set in the ratio of 7:3.

Training Classification Models on the Training dataset: After having split the dataset into two components, we have trained the Training dataset using various classification algorithms from the Scikit Learn Library. The following algorithms have been used for training the Training dataset: Multinomial Naïve Bayes, Linear Support Vector Machine, Decision Tree Algorithm, Random Forest Classifier Algorithm, and XGB Classifier Algorithm. To create a nice baseline for the task, we started with the Multinomial Naive Bayes and then proceeded to the other algorithms to increase the accuracy of our prediction, the results of which are discussed below.

Result Prediction: We have obtained the following levels of accuracy on the Testing dataset with the various classification algorithms. XGB Classifier Algorithm showed the

highest level of accuracy at 80.3% followed by Decision Tree Algorithm which had an accuracy of 71.6%. The third best accuracy of 70.1% was by using the Random Forest Classifier Algorithm. In the order of decreasing level of accuracy, Linear Support Vector Machine and Multinomial Naïve Bayes show an accuracy at 60.5% and 50.9% respectively.

Tuning the hyperparameters: We have tuned the hyperparameters in all the algorithms using the Randomized Search Cross-Validation technique. Cross-validation validates the model and splits the entire data into multiple Testing and Training dataset.

RESULT

It is very important to get the prediction results and compare the efficiency of the different algorithms used to get an idea of the best predictive model**.** In this section, we compare the accuracy levels of the various classification algorithms obtained from the Scikit Learn Library. For this study, five different classification algorithms have been used. Various classification metrics can be used for the task. We used the classification report of the algorithms that have been detailed below, which tells us about the precision, recall, and the f1-score of all the different algorithms used in this study. Precision is defined as the ratio of the correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of the correctly predicted positive observations to all the observations in the actual class. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

Multinomial Naïve Bayes Algorithm

	precision	recall	f1-score	support
0	0.78	0.94	0.85	615
1	0.35	0.15	0.22	613
2	0.44	0.32	0.37	624
3	0.55	0.62	0.58	570
4	0.50	0.26	0.34	628
5	0.48	0.26	0.34	604
6	0.74	0.77	0.75	595
7	0.44	0.63	0.52	583
8	0.42	0.66	0.51	566
9	0.35	0.51	0.41	602
accuracy			0.51	6000
macro avg	0.50	0.51	0.49	6000
weighted avg	0.50	0.51	0.49	6000

As observed from the table above, the accuracy of the multinomial Naïve Bayes algorithm is 0.5086666666666667 or 50.9%.

Linear Support Vector Machine Algorithm

	precision	recall	f1-score	support
0	0.95	0.98	0.97	615
1	0.40	0.27	0.32	613
2	0.54	0.38	0.45	624
3	0.54	0.74	0.63	570
4	0.57	0.62	0.60	628
5	0.64	0.46	0.54	604
6	0.81	0.92	0.86	595
7	0.55	0.46	0.50	583
8	0.71	0.70	0.71	566
9	0.36	0.54	0.43	602
accuracy			0.61	6000
macro avg	0.61	0.61	0.60	6000
weighted avg	0.61	0.61	0.60	6000

The accuracy of the linear support vector machine algorithm has been calculated at 0.6053333333333333 or 60.5% which is around 10% more than the accuracy achieved using the multinomial naïve Bayes algorithm.

Decision Tree Algorithm

	precision	recall	f1-score	support
0	1.00	1.00	1.00	615
1	0.70	0.40	0.51	613
2	0.69	0.55	0.61	624
3	0.78	0.72	0.75	570
4	0.74	0.79	0.76	628
5	0.70	0.77	0.73	604
6	0.57	0.93	0.71	595
7	0.70	0.63	0.66	583
8	0.79	0.74	0.77	566

9	0.60	0.64	0.62	602
accuracy			0.72	6000
macro avg	0.73	0.72	0.71	6000
weighted avg	0.73	0.72	0.71	6000

This algorithm shows an accuracy of 0.7166666666666667, rounded off to 72% which is 11.5% more than its predecessor and 21.1% more than the first algorithm used.

Random Forest Classifier Algorithm

	precision	recall	f1-score	support
0	0.98	1.00	0.99	615
1	0.56	0.42	0.48	613
2	0.56	0.55	0.56	624
3	0.72	0.79	0.75	570
4	0.74	0.68	0.71	628
5	0.69	0.62	0.65	604
6	0.96	0.95	0.95	595
7	0.62	0.67	0.64	583
8	0.72	0.80	0.76	566
9	0.47	0.55	0.51	602
accuracy			0.70	6000
macro avg	0.70	0.70	0.70	6000
weighted avg	0.70	0.70	0.70	6000

The level of accuracy of random forest classifier algorithm is 0.7011666666666667 or 70% which is 2% less than the random forest classifier algorithm.

XGB Classifier Algorithm

	precision	recall	f1-score	support
0	1.00	1.00	1.00	615
1	0.81	0.52	0.63	613
2	0.82	0.61	0.70	624
3	0.79	0.90	0.84	570

4	0.76	0.91	0.83	628
5	0.69	0.82	0.75	604
6	0.99	0.95	0.97	595
7	0.78	0.74	0.76	583
8	0.79	0.92	0.85	566
9	0.66	0.67	0.67	602
accuracy			0.80	6000
macro avg	0.81	0.81	0.80	6000
weighted avg	0.81	0.80	0.80	6000

At 0.8035 or 80.3%, the accuracy level achieved using the XGB classifier algorithm outperforms the decision tree algorithm by 8.3% making it the best among the various classifier algorithms used as a part of the study.

CONCLUSION

The purpose of this study is to compare the different machine learning algorithms used for the classification of texts and arrive at the best predictive model. We have chosen published documents on Chemistry and have tested various classification algorithms on the dataset containing documents on various topics. 2000 documents each from the major subtopics of chemistry came under the ambit of the study. After the usual processes of data cleaning and converting the cleaned data to a vector form, the data was split into Training and Test data sets using the Randomized Search Cross-Validation technique. The Training dataset was subjected to various classification algorithms and the experimental results on the accuracy levels obtained on these Training datasets indicate that the XGB Classifier Algorithm shows the highest level of accuracy followed by the Decision Tree Algorithm. The lowest accuracy level was recorded at 50.9%. It can, therefore, be concluded that the XGB Classifier Algorithm is the best among the classifier algorithms used in the context of this study.

Reference

[1]B. Altınel, M. Can Ganiz and B. Diri, "A corpus-based semantic kernel for text classification by using meaning values of terms", *Engineering Applications of Artificial Intelligence*, vol. 43, pp. 54-66, 2015. Available: 10.1016/j.engappai.2015.03.015.

[2]R. Pinheiro, G. Cavalcanti and I. Tsang, "Combining dissimilarity spaces for text categorization", *Information Sciences*, vol. 406-407, pp. 87-101, 2017. Available: 10.1016/j.ins.2017.04.025.

[3]D. Wang, J. Wu, H. Zhang, K. Xu and M. Lin, "Towards enhancing centroid classifier for text classification—A border-instance approach", *Neurocomputing*, vol. 101, pp. 299-308, 2013. Available: 10.1016/j.neucom.2012.08.019.

[4]Y. Yang, *Information Retrieval*, vol. 1, no. 12, pp. 69-90, 1999. Available: 10.1023/a:1009982220290

[5]B. Kessler, G. Numberg and H. Schütze, "Automatic detection of text genre", 2020.

[6]F. Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1-47, 2002. Available: 10.1145/505282.505283.

[7]E. Leopold and J. Kindermann, *Machine Learning*, vol. 46, no. 13, pp. 423-444, 2002. Available: 10.1023/a:1012491419635

[8]J. Brank, M. Grobelnik, N. Milic-Frayling and D. Mladenic, "Interaction of Feature Selection Methods and Linear Classification Models", *International Workshop on Text Learning, in Conjunction with International Conference on Machine Learning*, pp. 1-6, 2002.

[9]X. Han, G. Zu, W. Ohyama, T. Wakabayashi and F. Kimura, "Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination", *Content Computing*, pp. 463-468, 2004. Available: 10.1007/978-3-540-30483-8_57

[10]J. Shlens, "A tutorial on Principal Component Analysis", 2014. Available: https://arxiv.org/abs/1404.1100

[11]A. Izenman, "Linear Discriminant Analysis", *Springer Texts in Statistics*, pp. 237-280, 2013. Available: 10.1007/978-0-387-78189-1_8.

[12]R. Kohavi and G. John, "Wrappers for feature subset selection", *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997. Available: 10.1016/s0004-3702(97)00043-x

[13]S. Das, ""Filters, wrappers and a boosting-based hybrid for feature selection,"", *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 74-81, 2001.

[14]L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy,"", *Journal of Machine Learning Research,*, vol. 5, pp. 1205-1224, 2004.

[15]L. Yu and H. Liu, "Feature selection for high-dimensional data: a fast correlation-based filter solution", *DBLP Conference: Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003)*, vol. 2, pp. 856-863, 2003.

[16]E. Leopold and J. Kindermann, *Machine Learning*, vol. 46, no. 13, pp. 423-444, 2002. Available: 10.1023/a:1012491419635.

[17]R. Madsen, S. Sigurdsson, L. Hansen and J. Larsen, "Pruning the vocabulary for better context recognition", *Proceedings of the 17th International Conference on Pattern Recognition,*

2004. ICPR 2004., Cambridge, 2004, pp. 483-488, 2004. Available: doi: 10.1109/ICPR.2004.1334270.

[18]F. Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys (CSUR)*, vol. 34, no. 1, pp. 1-47, 2002. Available: 10.1145/505282.505283.

[19]E. Leopold and J. Kindermann, *Machine Learning*, vol. 46, no. 13, pp. 423-444, 2002. Available: 10.1023/a:1012491419635.

[20]G. Forman, "An Extensive Empirical Study of Feature Selection Metrics for Text Classification", *Journal of Medicine Learning Research*, vol. -3, 2003.

[21]P. Soucy and G. Mineau, "Feature Selection Strategies for Text Categorization", *Advances in Artificial Intelligence. Canadian AI 2003. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence)*, vol. 2671, 2003. Available: https://doi.org/10.1007/3-540-44886-1_41.

[22]D. Mladenić, J. Brank, M. Grobelnik and N. Milic-Frayling, "Feature selection using linear classifier weights", *Proceedings of the 27th annual international conference on Research and development in information retrieval - SIGIR '04*, 2004. Available: 10.1145/1008992.1009034.

[23]K. Naidu, A. Dhenge and K. Wankhade, "Feature Selection Algorithm for Improving the Performance of Classification: A Survey," *2014 Fourth International Conference on Communication Systems and Network Technologies*, Bhopal, 2014, pp. 468-471, doi: 10.1109/CSNT.2014.99.

[24]K. Torkkola, "Discriminative features for text document classification", *Formal Pattern Analysis & Applications*, vol. 6, pp. 301-308, 2004. Available: https://doi.org/10.1007/s10044-003-0196-8.

[25]E. Montañés, J. Quevedo and I. Díaz, "A Wrapper Approach with Support Vector Machines for Text Categorization", *Computational Methods in Neural Modeling*, pp. 230-237, 2003. Available: 10.1007/3-540-44868-3_30.

[26]J. Guan and S. Zhou, "Pruning Training Corpus to Speedup Text Classification. In: Hameurlain A., Cicchetti R., Traunmüller R. (eds) Database and Expert Systems Applications. DEXA 2002.", *Lecture Notes in Computer Science, vol 2453. Springer, Berlin, Heidelberg.*, 2002. Available: https://doi.org/10.1007/3-540-46146-9_82.

[27]D. Fragoudis, D. Meretakis and S. Likothanassis, "Integrating feature and instance selection for text classification", *Conference: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002. Available: DOI: 10.1145/775047.775120.

[28]X. Han, G. Zu, W. Ohyama, T. Wakabayashi and F. Kimura, "Accuracy Improvement of Automatic Text Classification Based on Feature Transformation and Multi-classifier Combination", *Content Computing*, pp. 463-468, 2004. Available: 10.1007/978-3-540-30483-8_57.

[29]W. Qiang, W. XiaoLong and G. Yi, "A Study of Semi-discrete Matrix Decomposition for LSI in Automated Text Categorization", *Natural Language Processing – IJCNLP 2004. IJCNLP 2004. Lecture Notes in Computer Science*, vol. 3248, 2005. Available: https://doi.org/10.1007/978-3-540-30211-7_64.

[30]S. Kim, H. Rim, D. Yook and H. Lim, "Effective Methods for Improving Naive Bayes Text Classifiers", *Lecture Notes in Computer Science*, pp. 414-423, 2002. Available: 10.1007/3-540-45683-x_45.

[31]K. Schneider, "Techniques for Improving the Performance of Naive Bayes for Text Classification", *Computational Linguistics and Intelligent Text Processing*, pp. 682-693, 2005. Available: 10.1007/978-3-540-30586-6_76.

[32]M. Klopotek and M. Woch, "Very Large Bayesian Networks in Text Classification", *Conference: Computational Science - ICCS 2003, International Conference, Melbourne, Australia and St. Petersburg, Russia*, vol. 2657, no., pp. 397-406, 2003. Available: DOI: 10.1007/3-540-44860-8_41.

[33]J. Shanahan and N. Roma, "Improving SVM Text Classification Performance through Threshold Adjustment", *Machine Learning: ECML 2003*, pp. 361-372, 2003. Available: 10.1007/978-3-540-39857-8_33.

[34]D. Johnson, F. Oles, T. Zhang and T. Goetz, "A decision-tree-based symbolic rule induction system for text categorization", *IBM Systems Journal*, vol. 41, no. 3, pp. 428-437, 2002. Available: 10.1147/sj.413.0428.

[35]H. Lim, "Improving kNN Based Text Classification with Well Estimated Parameters", *Neural Information Processing*, pp. 516-523, 2004. Available: 10.1007/978-3-540-30499-9_79.

[36]N. Chawla, K. Bowyer, L. Hall and W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique", *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002. Available: 10.1613/jair.953.

[37]V. Lertnattee and T. Theeramunkong, "Parallel Text Categorization for Multi-dimensional Data", *Parallel and Distributed Computing: Applications and Technologies*, pp. 38-41, 2004. Available: 10.1007/978-3-540-30501-9_10.

[38]Y. Bao and N. Ishii, "Combining Multiple K-Nearest Neighbor Classifiers for Text Classification by Reducts", *Discovery Science*, pp. 340-347, 2002. Available: 10.1007/3-540-36182-0_34.

[39]S. Cho and J. Lee, "Learning Neural Network Ensemble for Practical Text Classification", *Intelligent Data Engineering and Automated Learning*, pp. 1032-1036, 2003. Available: 10.1007/978-3-540-45080-1_145.

[40]Y. Bi, D. Bell, H. Wang, G. Guo and K. Greer, "Combining Multiple Classifiers Using Dempster's Rule of Combination for Text Categorization", *Modeling Decisions for Artificial Intelligence*, pp. 127-138, 2004. Available: 10.1007/978-3-540-27774-3_13.

[41]P. Nardiello, F. Sebastiani and A. Sperduti, "Discretizing Continuous Attributes in AdaBoost for Text Categorization", *Lecture Notes in Computer Science*, pp. 320-334, 2003. Available: 10.1007/3-540-36618-0_23.

[42]K. Aas and L. Eikvil, "Text Categorisation A Survey", *Norwegian Computer Centre*, 1999.

[43]C. Aggarwal and C. Zhai, "A Survey of Text Classification Algorithms", *Mining Text Data. Springer, Boston, MA*, 2012. Available: https://doi.org/10.1007/978-1-4614-3223-4_6.

[44]T. Guzella and W. Caminhas, "A review of machine learning approaches to Spam filtering", *Expert Systems with Applications*, vol. 36, no. 7, pp. 10206-10222, 2009. Available: 10.1016/j.eswa.2009.02.037.

[45]J. García Adeva, J. Pikatza Atxa, M. Ubeda Carrillo and E. Ansuategi Zengotitabengoa, "Automatic text classification to support systematic reviews in medicine", *Expert Systems with Applications*, vol. 41, no. 4, pp. 1498-1508, 2014. Available: 10.1016/j.eswa.2013.08.047.

[46]M. Mirończuk and J. Protasiewicz, "A recent overview of the state-of-the-art elements of text classification", *Expert Systems with Applications*, vol. 106, pp. 36-54, 2018. Available: 10.1016/j.eswa.2018.03.058.

[47]C. Aggarwal, "Data Mining", 2015. Available: 10.1007/978-3-319-14142-8.