JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 19 (2021) 144–150
Received: 20 April 2021
Accepted: 04 May 2021

Model of the text classification system using fuzzy sets

Model systemu klasyfikacji tekstu z wykorzystaniem zbiorów rozmytych

Dmytro Salahor*, Jakub Smołka

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract
Classification of work's subject area by keywords is an actual and important task. This article describes algorithms for classifying keywords by subject area. A model was developed using both algorithms and tested on test data. The results were compared with the results of other existing algorithms suitable for these tasks. The obtained results of the model were analysed. This algorithm can be used in real-life tasks.

Keywords: text classification; "fuzzy" sets; classification; fuzzy rules; fuzzy logic

Streszczenie
Klasyfikacja tematyki pracy według słów kluczowych jest aktualnym i ważnym zadaniem. W artykule opisano algorytmy klasyfikowania słów kluczowych według obszaru tematycznego. Model został opracowany przy użyciu dwóch algorytmów i przetestowany na danych testowych. Uzyskane wyniki porównano z wynikami innych istniejących algorytmów odpowiednich do tego zadania. Uzyskane wyniki modelu analizowano. Algorytm ten może być stosowany w zadaniach rzeczywistych.

Słowa kluczowe: klasyfikacja tekstu; zbiory rozmyte; klasyfikacja; reguły rozmyte: logika rozmyta

*Corresponding author
Email address: s97218@pollub.edu.pl (D. Salahor)

1. Introduction

With the increase of text information amount, it is very important to understand which area the text belongs to. When working with databases containing scientific papers and articles, getting the subject area of the work without looking through its entire text is need. In this case, work attributes such as keywords can be used. However, manually assigning keywords or subject areas will take too much time and resources. Therefore, automating the definition of text attributes will save time and resources. Also, when selecting attributes for text information, in reality, you can get belonging not to one class, but to several. Modern classification systems must support this capability.

The purpose of the article is to develop, describe and test a text classification model. The model must have high accuracy and non-binary classification capability.

This article proposes an algorithm for recognizing the subject area of work by its keywords. Two versions of the classifier are described in detail, models are implemented and tested on real data. A comparison was made with other algorithms such as Decision tree [1-5], Support-vector machine [6-10], Neural network [11-14].

2. Review of text analysis methods

In the [15] a comparative analysis of the best way to classify complaint texts for a state online complaint service in Indonesia was made. The following algorithms participated in the comparison: Naive Bayes [16-17], Maximum Entropy [18-19], K-Nearest Neighbours [20], Random Forest [21-22], and Support Vector Machines, and two ensemble strategies - hard voting and soft voting. The results also indicate that generally all the ensemble methods performed better than the individual classifiers.

In [23] a keyword recognition system for texts using neural networks and hierarchical taxonomies was described. For each category in taxonomy own pre-trained neural network is used. The neural network uses function "logistic" and solver "adam". Using the combined hierarchical system of neural networks, with a sample of 2843 documents, an accuracy of 77.87% was achieved.

In [24] a new approach to categorizing text for category recognition for online newspapers was described. The approach has strict rules that determine a possibility of using a particular database for classification. To classify the system, the methods Support Vector Machine (SVM), Hidden Markov was chosen. Using this approach gives high accuracy. When using Hidden Markov Model (HMM) [25-26], the best accuracy was obtained.

In [27] an automatic text classification system based on the genetic algorithm classifier has been developed. Before classification with genetic algorithm, the text data was pre-processed, translated into correct representation, and the selection of features was made. When testing the system, 20291 documents of 6 categories were used. Using 1000 selected words, a performance of 0.748 was achieved, which is more than using kNN and decision tree classifiers.

In [28] a method for investigating the temporal patterns using keywords in the comments of online newspapers was proposed. The system should return a conclusion based on the content of the comment. Text data has been cleaned and categorized. As a result, the following conclusions were drawn: the drop in activity is associated with the end of the event and the time before

this event; lexeme frequency maps were obtained; using temporal analysis, you can get a clear picture of the sentiment of changes.

In [29] a system for extracting keywords and sentiment from Twitter posts was developed. The Archivist API was used to collect information. The sample consisted of 40 000 tweets. The using the developed system in [30] with knowledge enhancer and synonym binder allowed to improve the results from 0.1% to 55% in comparison with the usual keyword search.

In [30] semi-automatic context analysis and text correction using specialized linguistic graphs was used and a system for text correction and context analysis has developed as a part of logic of web application. Developing the model, graphs composed of special neurons of different types was used. Comparison with similar programs for text correction has performed. The results of [30] are: an effective graph model for writing words and punctuation marks appearing in the examined text; developed methods to obtain texts from several sources for graph construction; text analysis and contextual adjustment methods developed and implemented; web application (website) was made that allows to use of implemented algorithms.

3. Classification model using fuzzy-sets

There are many ways to classify texts. This article will consider a method for classifying texts that is suitable for classifying scientific papers in different subject areas. The model is based on «fuzzy sets» [31,32].

Fuzzy set is a set, each element of which is matched with a real number in the range [0; 1], which indicates how much the element belongs to set [31].

The idea of such a classifier is to assign a real number within the range [0..1] for each possible variant. This classification method is more flexible than the usual classification method, where each variant is assigned an integer from the set {0, 1}. When writing the article, two versions of such a classifier were used and compared: using key phrases and using keywords.

The classifier model can be divided into two parts: training and direct classification. The model uses a normalized fuzzy set during training and classification. Values are normalized upon classification.

3.1. Training model description

Let there be a set containing *N* subject areas, among which the classification should be performed. As input data for training, the subject area and keywords that relate to the subject area are given in a form of a set:

$$T = \{P, \{Tp1, Tp2..Tpa..Tpk\}\} \quad (1)$$

where *P* – subject area for key phrases, *Tp1* .. *Tpk* – key phrases.

Depending on *Tpa*, subject areas *P* are added to the corresponding set *Ma* (2).

A set *Ma* has structure as below:

$$Ma = \{Ipa, \{\{P1, i1\}, \{P2, i2\}..\{Pa, ia\}..\{PN, iN\}\} \quad (2)$$

where *Ipa* – key phrase, *P1..PN* – subject areas, *i1..iN* – integer number, indicating the frequency of use of the key phrase in the subject area. It should be noted that the set does not allow duplicates.

The set of *Ma* sets, each of which describes its own keyword, makes up the set Mp, which is the result of training:

$$Mp = \{M1, M2..Ma..Mm\} \quad (3)$$

where *M1..Mm* – a set containing information about in which subject areas and with what frequency the key phrase occurs.

As a result, the more training material is available for training, the more power the set *Mp* has, and the more accurate results the algorithm can provide.

3.2. Model of classifier description

The input data for the finished classifier is the set *S*, containing the key phrases of the work, the classification of which must be carried out:

$$S = \{s1, s2..sa..sN\} \quad (4)$$

where *s1* .. *sN* - strings containing key phrases.

The key phrases have the form of strings, each of which comprises words separated by whitespace.

The output of the classifier is the set *Out*:

$$Out = \{\{P1, f1\}, \{P2, f2\}..\{Pa, fa\}..\{PN, fN\}\} \quad (5)$$

where *f1* .. *fN* – real numbers in the range [0..1], which show the value of belonging to a particular subject area, *P1* .. *PN* – subject areas.

Also done normalizing each *fa* multiplying it by the normalization factor *j*:

$$j = 1/\max(a1..aN) \quad (6)$$

where *max(a1..aN)* – maximal value in set *{a1..aN}*.
The classification algorithm is as follows:

Algorithm 1: Classification

```
Input : N = count of subject areas;
Set Mp;
Output: Out with N elements;
for Each keyword do
    Create set Ma from Mp by keyword;
    Get sum of ia in all Pa sets;
    Write sum of ia in fa in set Out;
    if sum of ia not equal 0 then
        Normalize fa: divide fa by sum of all ia in fa;
    end
end
for Each fa in Out do
    Normalize Out: divide fa by max of fa;
end
return Out;
```

3.3. Classifier modification

To increase recognition by the classifier, a modification of the classifier is proposed. The modification consists in using not key phrases, but their components – keywords for training and classification. Keywords can be obtained by splitting key phrases by words.

In the modified model of the classifier, during training in (1), keywords are given as *Tp1..Tpk*; when classified in (4), keywords are given as *S1..SN*.

4. Model and algorithm realization

The model using the above algorithms was implemented in the Java language. HashMap and ArrayList were used as sets. The input data was submitted as file with CSV format (comma-separated values). Every row in the file has values: authors, title, link of work, author keywords, index keywords, subject area. Output data printed in console.

4.1. Input data preparing

The Scopus database was chosen as the input data source [33]. The database is free and allows you to export the required attributes of the works to a file of the required format. All papers from Lublin University of Technology were taken as exported data. A total of 6543 works were selected, of which 655 were used for testing. The resulting data sample was divided into two parts, 90% and 10%. The first part will be used for training, the second for testing. The classification was made between 27 subject areas.

Next subject areas was chosen:

- engineering,
- materials science,
- physics and astronomy,
- mathematics,
- computer science,
- environmental science,
- chemistry,
- chemical engineering,
- energy,
- agricultural and biological sciences,
- social sciences,
- earth and planetary sciences,
- biochemistry, genetics and molecular biology,
- medicine,
- economics, econometrics and finance,
- business, management and accounting,
- decision sciences,
- multidisciplinary,
- pharmacology, toxicology and pharmaceutics,
- arts and humanities,
- health professions,
- neuroscience,
- immunology and microbiology,
- psychology,
- dentistry,
- nursing,
- veterinary.

4.2 Evaluating classification results method

Since the classification is done by fuzzy sets, you need to decide what values at the output of the classifier are considered sufficient to assign a keyword search. In evaluating using next variables:

The model leaves in set all values greater than 0.5. Choosing the sorting of 0.5 due to the fact that if the *fa* is smaller than 50% in the fuzzy set, such variant under repeated less probable than others.

Then *Out* is compared with the set *R*:

$$R = \{Pa1..Pam\} \quad (7)$$

where *Pa1..Pam* are real subject areas for the *S* keyword list.

The goal of the classifier is to get an *Out* that is as close as possible to *R*. It is also necessary to evaluate the classification accuracy. The model uses the following algorithm:

Algorithm 2: Accuracy evaluation

Input : *Out*, *R*;
Output: *accuracy*;
if $|Out| \geq |R|$ **then**
  $accuracy = |(R \cap Out)| / |Out|$;
else
  $accuracy = |(R \cap Out)| / |R|$;
end
return *accuracy*;

In algorithm next statements used:

$$|(R \cap Out)|/|Out| \quad (8)$$

where $|(R \cap Out)|$ - number of right keywords/ key phrases in *Out*, $|Out|$ - number of right keywords/ key phrases in *Out*.

$$|(R \cap Out)|/|R| \quad (9)$$

where $|(R \cap Out)|$ - number of right keywords/ key phrases in *Out*, $|R|$ - number of right keywords/ key phrases in *R*.

5. Text classification using popular classificators

For comparison with the work of the developed algorithm, classifiers of the following types were tested: Decision tree, Support-vector machine, Backpropagation neural network. The following results were compared:

- using first keyword,
- using first three keywords,
- using last three keywords,
- using random three keywords,
- using all keywords.

Since the ability of the model described in the article to perform multiple classification is already superior to the models used in comparison, which perform only binary classification, therefore, to be able to compare

them, the model developed in the article will also perform binary classification.

Model building was performed in KNIME - open-source software for data analysis. KNIME allows you to work with data, perform data processing and visualize the results. A visual editor is used to work in KNIME. It is also possible to use code inserts in programming languages such as Java, Python, Javascript.

When creating the required models, the following elements were used:

- for neural network - Rprop MLP Learner and Multi-LayerPerceptron Predictor [34, 11-13],
- for support-vector machine - SVM Learner and SVM Predictor [35,36],
- for decision tree - Decision Tree Learner and Decision Tree Predictor. [37,38].

When selecting keywords, inserts in the Java language were used.

6. Analysis of the obtained results

The analysis is presented separately for the results of comparing binary and non-binary models. In comparison of binary models, the analysis of the results obtained during the operation of the two algorithms described in the article, as well as the results of the operation of the Decision tree, Support-vector machine, Backpropagation neural network models, is given. In comparison of non-binary models, the analysis of the results obtained by the operation of the two algorithms described in the article are presented.

6.1. Analysis of the binary models

In Table 1, the simulation results for binary classifiers are presented.

Table 1: Accuracy values of various algorithms

	Standard	Modified	Decision tree	SVM	NN
Only 1 keyword	0.538	0.586	0.665	0.585	0.585
Only 3 first keywords	0.564	0.611	0.626	0.586	0.586
Only 3 last keywords	0.505	0.611	0.63	0.586	0.585
Random 3 keywords	0.541	0.614	0.611	0.586	0.586
All keywords	0.750	0.771	0.586	0.585	0.585

When using a small count of keywords, the best results are obtained with the Decision tree. The results in the Support-vector machine and Neural network are almost unchanged when the number of keywords is changed. When using the maximum count of keywords, the best results are obtained with the modified algorithm described in the article. A visual representation of the accuracy of these models is shown in Figure 1.

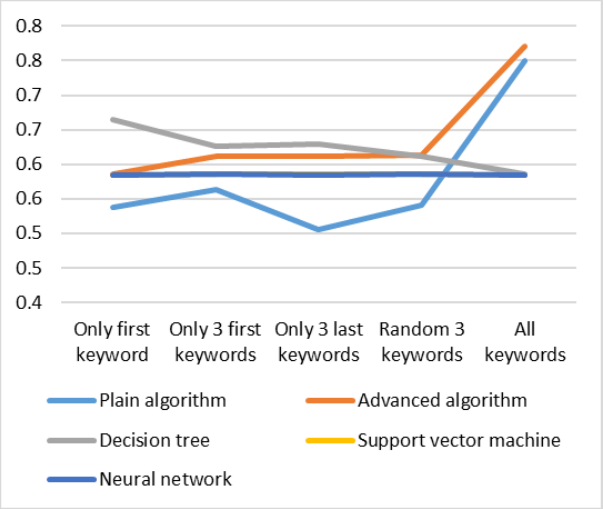

Figure 1: Accuracy values of various algorithm.

6.2. Analysis of the non-binary models

Two different models were implemented: for the standard version of the algorithm, and for the modified one.

For each model, 6554 work data sets were used for training, and 655 work data sets for testing.

When considering the successful operation of the system subject to at least one true subject area model is successful in 79% for the standard algorithm. The arithmetic mean for the standard model is 47 (Table 2).

Table 2: Accuracy values of standard algorithm

	count of sets	%
All classified	118	18.02
At least one classified	520	79.39
No one classified	135	20.61
Summary	655	100
Average value	0.47	
Median value	0.5	

For the modified algorithm, more averaged values were obtained, the total number of positive classifications was 78%. The arithmetic mean for the modified model is 40% (Table 3).

Table 3: Accuracy values of modified algorithm

	count of sets	%
All classified	56	8.55
At least one classified	513	78.32
No one classified	142	21.68
Summary	655	100
Average value	0.40	
Median value	0.33	

The difference of arithmetic mean for the models is due to a more blurred result of the modified algorithm. Considering that is not necessary that for each set of data has been full compliance, it is a satisfactory result.

Figures 2 and 4 show the classification accuracy for standard (Figure 2) and modified (Figure 4) algorithms.

Probability values are grouped for clarity. It shows that in contrast to the standard model in which clearly shows the standard probability distribution, in the modified model blur occurs to low probability values.

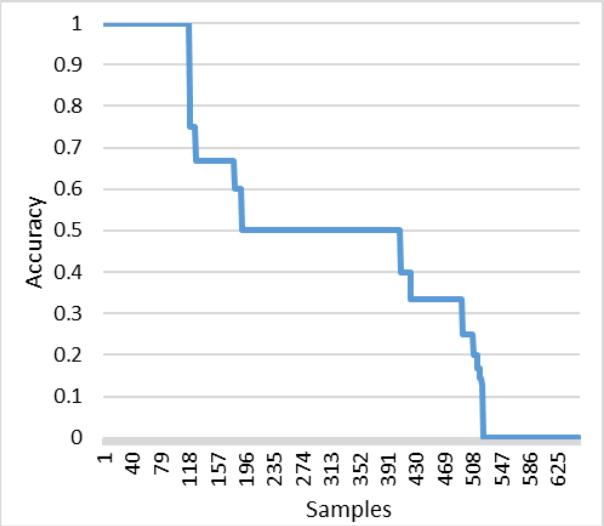

Figure 2: Sorted accuracy values of standard algorithm.

Figures 3 and 5 show the number of correctly defined subject areas for standard (Figure 3) and modified (Figure 5) algorithms. Count of classified values also are grouped for clarity.

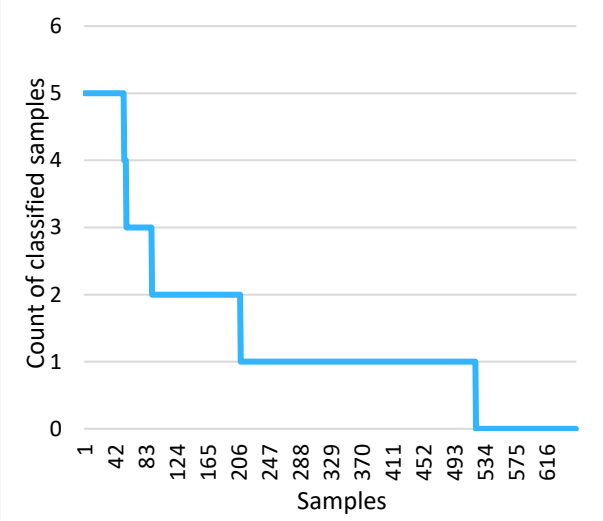

Figure 3: Sorted classified count of subject areas of standard algorithm.

For the standard model, there is a shift in the mean probability of 50%. The majority of correctly classified samples have only one correctly classified subject area.

Similarly, for a number of suitable domains, unlike the modified model in softer distribution - there are less complete correspondences (Figure 4,5). For the modified model, there is a shift in the average probability below 50%. The majority of correctly classified samples (as in the standard model) have only one correctly classified subject area.

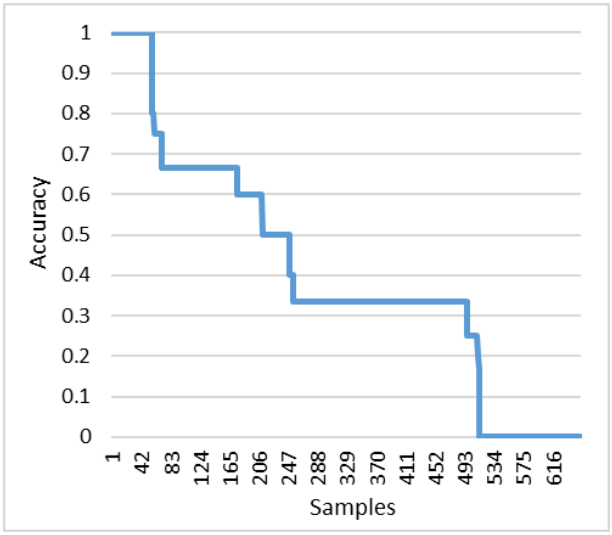

Figure 4: Sorted accuracy values of modified algorithm.

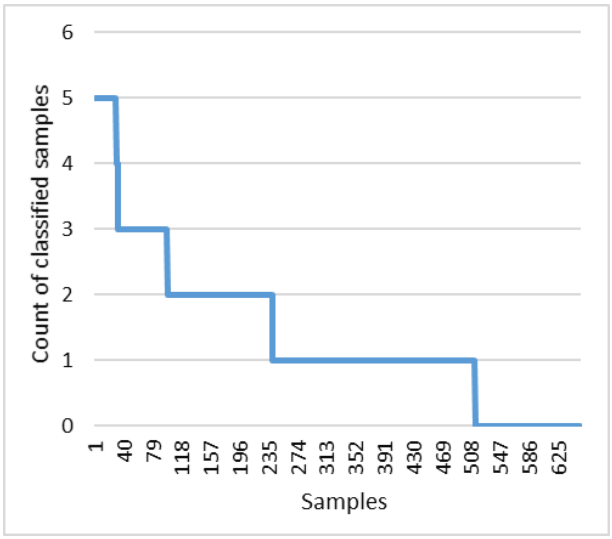

Figure 5: Sorted classified count of subject areas of modified algorithm.

7. Conclusions

The described classification algorithm is suitable for determining the subject area by work's keywords and key phrases. A model has been implemented that uses both versions of the algorithm and tested on real data. The obtained accuracy estimate of 79% allows one to determine with great accuracy not only the main, but also additional areas to which the work is only partially related. The standard version of the algorithm is more successful with large amounts of data, the modified version is more flexible and can work with new and incomplete data. The developed models can be used as binary classifiers with good accuracy. The developed algorithm can be used in databases of scientific papers and other types of text data.

References

[1] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees, Wadsworth & Brooks, Pacific Grove, 1984.

[2] G. V. Kass, An exploratory technique for investigating large quantities of categorical data, Applied Statistics 29 (1980) 119–127.

[3] E. B. Hunt, J. Marin, P. J. Stone, Experiments in induction, Academic, New York, 1966.

[4] R. S. Michalski, J. G. Carbonell, T. M. Mitchell, Machine learning. An artificial intelligence approach (1983) 463–482.

[5] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.

[6] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, Proceedings of annual conference computational learning theory, ACM Press, Pittsburgh (1992) 144–152.

[7] C. Cortes, V. Vapnik, Support vector networks, Machine Learning 20 (1995) 273–297.

[8] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, M. Anthony, Structural risk minimization over data-dependent hierarchies, IEEE Transactions on Information Theory 44 (1998) 1926–1940.

[9] J. Shawe-Taylor, N. Cristianini, Margin distribution and soft margin, Advances in large margin classifiers, MIT Press, Cambridge (2000) 349–358.

[10] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, Proceedings of the European conference on machine learning, Springer, Berlin (1998) 137–142.

[11] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychological Review 65 (1958) 386–408.

[12] J. Schmidhuber, Deep Learning in Neural Networks: An Overview, Neural Networks 61 (2015) 85–117.

[13] S. E. Dreyfus, Artificial neural networks, back propagation, and the Kelley-Bryson gradient procedure, Journal of Guidance, Control, and Dynamics 13 (1990) 926–928.

[14] E. Mizutani, S. E. Dreyfus, K. Nishio, On derivation of MLP backpropagation from the Kelley-Bryson optimal-control gradient formula and its application, IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium 2 (2000) 167–172.

[15] M. A. Fauzi, Automatic Complaint Classification System Using Classifier Ensembles, Telfor Journal 10 (2018) 123-128.

[16] D. Lewis, Naive Bayes at forty: the independence assumption in information retrieval, Proceedings of the 10th European Conference on Machine Learning, Springer, Berlin (1998) 4–15.

[17] A. McCallum, K. Nigam, A comparison of event models for Naive Bayes text classification, AAAI-98 Workshop on Learning for Text Categorization, AAAI Press, California (1998) 41–48.

[18] R. Lau, R. Rosenfeld, S. Roukos, Adaptive language modelling using the maximum entropy principle. Proceedings of the ARPA Human Language Technology Workshop, San Francisco (1993) 108–113.

[19] A. L. Berger, S. A. Della Pietra, V. J. Della Pietra, A maximum entropy approach to natural language processing, Computational Linguistics 22 (1996) 39–71.

[20] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, The American Statistician 46 (1992) 175–185.

[21] T. K. Ho, Random Decision Forests, Proceedings of the 3rd International Conference on Document Analysis and Recognition 14–16, Montreal (1995) 278–282.

[22] T. K. Ho, The Random Subspace Method for Constructing Decision Forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (1998) 832–844, http://dx.doi.org/10.1109/34.709601.

[23] A. Ciapetti, R. Di Florio, L. Lomasto, G. Miscione, G. Ruggiero, D. Toti, NETHIC: A System for Automatic Text Classification using Neural Networks and Hierarchical Taxonomies, Proceedings of the 21st International Conference on Enterprise Information Systems 1 (2019) 296-306.

[24] G. Krishnalal, S. Rengarajan, K. Srinivasagan, A New Text Mining Approach Based on HMM-SVM for Web News Classification, International Journal of Computer Applications 1 (2010) 98-104. DOI. 10.5120/395-589

[25] L. E. Baum, T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, The Annals of Mathematical Statistics 37 (2019) 1554–1563.

[26] L. E. Baum, G. R. Sell, Growth transformations for functions on manifolds, Pacific Journal of Mathematics 27 (1968) 211–227.

[27] M. I. Khaleel, I. I. Hmeidi, H. M. Najadat, An Automatic Text Classification System Based on Genetic Algorithm, Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on Social Informatics 31 (2016) 1–7.

[28] N. Medagoda, S. Shanmuganathan, Keywords based temporal sentiment analysis, 12th International Conference on Fuzzy Systems and Knowledge Discovery (2015) 1418-1425.

[29] R. Batool, A. M. Khattak, J. Maqbool, S. Lee, Precise tweet classification and sentiment analysis, 12th International Conference on Computer and Information Science (2013) 461-466.

[30] M. A. Gadamer, A. Horzyk, Semi-automatic contextual analysis and correction of texts by specialized linguistic graphs, AGH University of Science and Technology, 2019.

[31] L. A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338-353.

[32] P. Karczmarek, Selected problems of face recognition and decision-making theory, Wydawnictwo Politechniki Lubelskiej, 2018.

[33] The website for Elsevier B.V., Open database, https://www.scopus.com, [01.04.2021].

[34] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, Proceedings of the IEEE International Conference on Neural Networks 16 Piscataway (1993) 586-591.

[35] J Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, Advances in Kernel Methods: Support Vector Learning (1999) 185-208.

[36] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy, Improvements to Platt's SMO Algorithm for SVM Classifier Design, Neural Computation 13 (2001) 637-649, http://dx.doi.org/10.1162/089976601300014493.

[37] S. L. Salzberg. C4.5: Programs for Machine Learning by J. Ross Quinlan, Machine Learning 16, Morgan Kaufmann Publishers (1994) 235–240, http://dx.doi.org/10.1007/BF00993309.

[38] J. Shafer, R. Agrawal, M. Mehta, SPRINT: A scalable parallel classifier for data mining, VLDB, 2000.