A Conceptual Text Classification Model Based on Two-Factor Selection of Significant Words

Olesia Barkovska, Vladyslav Kholiev, Anton Havrashenko, Dmytro Mohylevskyi and Andriy Kovalenko

Kharkiv National University of Radio Electronics, Nauki ave., 14, Kharkiv, 61166, Ukraine

Abstract

The aim of the study is to develop a text classification conceptual model based on a combined method of two-factor selection of significant words in a frequency dictionary. The task is relevant due to the increase in the amount of textual information in electronic form, which requires organization and classification, for example, in the automatic processing of news flow, distribution of news texts in catalogs or analysis of different publications in the scientific field. Efficient processing of text arrays and the quality of searching for materials require an accurate correlation of the publication with other types of publications related to particular scientific field. It confirms the relevance of research in the field of automatic text documents' classification. Achieving this goal was possible due to the analysis of the dependence of the classification accuracy of the Reuters-21578, NSF and MiniNg20 datasets on the choice of significant words of the frequency dictionary on the basis of the TF-IDF.

The first study of the selection of topic-related words for classification based on such factor, as frequency of topic-related words showed that for the analyzed data set the most informative words are those that occur at least 10 to 15 times in the data set. The second study of the selection of topic-related words based on such factor, as the reduction of the frequency vector by determining the threshold of the frequency dictionary showed that using the range of significant words from 2000 to 4000 for all datasets gives more successful results than using all words in the feature vector. The proposed combined method of two-factor selection of topic-related words (on the base of frequency of topic-related words together with the threshold of the frequency dictionary) outperforms previous methods for all three datasets and increases the accuracy of text document classification from 2 to 4 percent.

Keywords

Text classification, text representation, TF-IDF, frequency dictionary, acceleration, accuracy

1. Introduction

With the increase in the amount of textual information in electronic form the task of automatic text classification continues to increase in relevancy. This task arises during the automatic processing of news flow and distribution of news texts in catalogs (Figure 1). For the convenience of users, directories are organized in a hierarchical structure: a directory consists of several subdirectories, etc.

The task of classification is especially important in the scientific field, where tens of thousands of monographs, articles, preprints, and other types of publications are being added annually in each discipline. Effective processing of such arrays and the quality of searching for materials relevant to a particular research area require an accurate correlation of each publication with its thematic category [1] for different languages, including Ukrainian [2, 3].

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine

EMAIL: olesia.barkovska@nure.ua (O. Barkosvka); vladyslav.kholiev@nure.ua (V. Kholiev); anton.havrashenko@nure.ua (A. Havrashenko); dmytro.mohylevskyi@nure.ua (D. Mohylevskyi); andriy.kovalenko@nure.ua (A. Kovalenko)
ORCID: 0000-0001-7496-4353 (O. Barkosvka); 0000-0002-9148-1561 (V. Kholiev); 0000-0002-8802-0529 (A. Havrashenko); 0009-0003-2889-6208 (D. Mohylevskyi); 0000-0002-2817-9036 (A. Kovalenko);

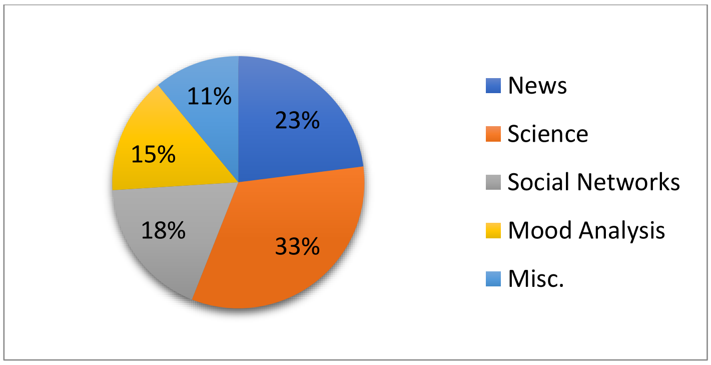

Figure 1: Practical applications of text array classifiers

After converting documents into vector form, this task is suitable to be solved by machine learning methods. Currently, TextMining is actively developing: research is being conducted, projects and competitions are being launched to identify the best algorithms in terms of accuracy.

Many different methods are used to solve the problem of classifying text documents [4, 5]. The k-nearest neighbors method and its modifications are widely used, where the classified object is assigned to the class that the k other closest objects in the training set “around” it belong to. Another algorithm is Bayesian classification, which works to calculate the posterior error probabilities of classes. A representative of linear classifiers is the support vector method, which involves constructing a hyperplane that separates the sample objects in the most optimal way. Recently, neural networks have been increasingly used to solve the classification problem [6, 7]. On average, the accuracy of various text classification algorithms varies from 70% to 90% and depends not only on the classification algorithms but also on the quality of the source data.

2. Related Works

Many existing methods of text classification are based on terminological proximity. The text is represented as a vector in Euclidean space, where the coordinate axes are terms, n-grams or lexemes that are extracted from the text, and the coordinate along the axis is statistical information about them. Thus, the text can be represented as frequency vectors of word occurrences based on TF, TF*IDF, C-TF*IDF, and other schemes [8, 9].

Another important parameter in text classification is the proximity measure calculated between vectors. Its choice has an impact on the quality of classification. The well-known metrics are: Euclidean distance, Minkowski distance, Otiai coefficient, Jaccard coefficient, projection distance, etc.

Consider in more detail the main methods used in text classification. These methods are related to supervised machine learning methods (table 1).

Metric classification methods include the k-nearest neighbors’ method, where the classified object is assigned to the class to which the objects in the training set closest to it belong. The classic k-nearest neighbors algorithm has many modifications. This is due to the high computational complexity of the algorithm and the low classification speed. One study compares the results of classifying Fudan University texts using five methods: the classical k-nearest neighbors’ method, k weighted nearest neighbors, fuzzy k-nearest neighbors, k-nearest neighbors based on Dempster-Shafer theory, and k-nearest neighbors based on fuzzy integral [10]. It is shown that the best accuracy of 86% is shown by the algorithm based on the fuzzy integral, while the accuracy of the classical k-nearest neighbors’ algorithm is only 78%.

Another group of classifiers is probabilistic. A widely used algorithm belonging to this class is naive Bayesian classification. It represents the simplest variation of Bayesian classifiers - a naive Bayesian classifier based on the assumption of feature independence. Since the classical approach to naive Bayesian classification often does not include the weights of the learned features in the conditional probability estimation, Liangxiao Jiang and co-authors in their study propose a naive Bayesian classification with deep feature weighting, which calculates weighted features by frequencies based on the training data, and then these weights are taken into account when calculating the probability [11]. In that paper, naive Bayesian classification is used to determine the authorship of texts. Depending on the representation of the text, for example, in the form of n-grams, the accuracy of the method in applying to this task showed results from 40% (with trigrams and tetragrams) to 96.67% (with terms). The study revealed a problem in the process of parameter estimation that can affect the accuracy of naive Bayesian text classification. To eliminate this problem, the authors propose to normalize the text for each document and use the feature weighting method. To improve the performance of the naive Bayesian classification, the method of auxiliary functions is also used, the Kullback-Leibler distance is calculated between words, naive Bayesian trees are built, polynomial naive Bayesian classification, Bernoulli naive Bayesian classification, Gaussian naive Bayesian classification, etc. The study shows that polynomial naive Bayesian classification gives a better result when classifying texts (although its accuracy is only 73.4%) than Bernoulli's naive Bayesian classification (its accuracy is 69.15%). When comparing the three methods based on naive Bayesian classification, it is shown that Bernoulli's naive Bayesian classification is comparable in terms of results to the classical one, while the Gaussian naive Bayesian classifier gives the best classification accuracy.

One of the examples of linear classifiers is the support vector machine, which consists in constructing a hyperplane that separates the sample objects in the most optimal way.

There is also a classification based on graph theory methods. It includes, for example, the random forest method. It consists in building an ensemble independent decision trees learning in parallel [12]. A number of studies have suggested ways to improve the performance of the random forest method. Thus, to solve multi-class problems for calculating the weights of objects, it is proposed to use the method of XI-squares [13]. By using a new feature weighting method for subspace sampling and a tree selection method, the subspace size is effectively reduced and classification performance is improved. Depending on the dataset, the method can demonstrate classification accuracy from 72% to 92%. The semantics-aware random forest algorithm on trees of different sizes shows an accuracy of 73-78%, while the accuracy of the classical algorithm is 57-60% [14].

Recently, neural networks have seen increased usage to solve the classification problem. In their work, Siwei Lai and co-authors propose to use recurrent convolutional neural networks to solve the text classification problem [15]. The authors conclude that the use of neural networks in the classification of text documents will help to avoid the problem of sparse data, as well as collect more contextual information about entities compared to traditional methods. Convolutional neural networks have shown high accuracy (83.98%) in the classification of patent documents.

Table 1
Classification methods

Methods	Accuracy	Scope	Computational complexity	Classification speed
k-nearest neighbors	78% – 86%	86% – 91%	High	Low
Support vector machine	63% – 90%	83% – 87%	Low	Low
Naive Bayesian classification	40% – 83%	80% – 90%	Low	Low
«Random forest»	57% – 78%	75% – 82%	High	High
Convolutional neural networks	83,98%	70% – 85%	High	High

There are many studies aimed at comparing the accuracy of text document classification using different methods. Thus, when comparing three methods: k-nearest neighbors based on fuzzy integral, support vector machine and Bayesian classification, the support vector machine showed the best accuracy of 90%. When classifying tweets in Turkish, the methods showed different classification results depending on the size of the training sample. The best results, from 63% to 83%, in all three cases were demonstrated by Bayesian classification. When classifying books, the Bayesian classifier also showed the best accuracy, 81%. However, when classifying Indian and English tweets, despite the fact that Bayesian classification was the most effective, its accuracy did not exceed 63%. The study uses five classifiers to classify data from news websites: k-nearest neighbors, random forest, polynomial naive Bayesian classifier, logistic regression, and support vector machine. The most effective algorithm was the support vector machine, which demonstrated not only a high accuracy of 91%, but also the fastest running time: at least one and a half times lower than the other algorithms studied [16, 17].

Combinations of different classification algorithms are also used to improve classification accuracy [18]. For example, the combination of k-nearest neighbors and support vector machine algorithms makes the classification accuracy higher by 1 to 2% than when these classifiers are used separately. The combination of k-nearest neighbors, the Rocchio algorithm, and the least squares method reduced the number of classification errors by 15%.

Thus, on average, the accuracy of various text classification algorithms varies from 70% to 90%. At the same time, the classification accuracy depends not only on the chosen classification algorithm, but also on the source data and preprocessing methods [19, 20].

That is, the analysis and development of a method that would rationally process the source data and classify it with a lower error rate is a popular and relevant task.

3. Aims and Tasks of the Work

The aim of the work is to create a conceptual model of two-factor text classification on the example of standardized training and test text data sets.

To achieve this goal, the following tasks have to be solved:

- to make the overview of existing methods of text data classification;
- to analyze methods of pre-processing and preparation of input text data;
- to develop a text classification model based on two-factor selection of topic-related words;
- to research the frequency vector reduction impact on the text classification accuracy;
- to analyze the results obtained.

4. Results and Discussion

When choosing a specific text classification algorithm, one should take into account the features of each of them. As before, the issue of determining the set of classifying features, their number, and how to calculate weights remains unresolved. In deep learning algorithms, the classification accuracy depends on the availability of a training set of appropriate size. Preparing such a set is a very time-consuming process. The problem of selecting the parameters of some algorithms at the training stage is still open.

Figure 2 shows a general scheme of the classification process, taking into account the main stages and options for their implementation.

After analyzing the existing methods of automatic text classification, a new two- factor classification model was developed. It is shown in Figure 3 in the form of IDEF0 notation.

In the proposed model, feature selection is performed using a two- factor approach based on TF-IDF and C-TF-IDF methods.

C-TF-IDF is a class-based TF-IDF procedure that can be used to create objects from text documents based on the class they are in.

The goal of a class-based TF-IDF is to provide all documents within a class with the same vector class. To do this, we must start thinking about TF-IDF in terms of classes rather than individual documents.

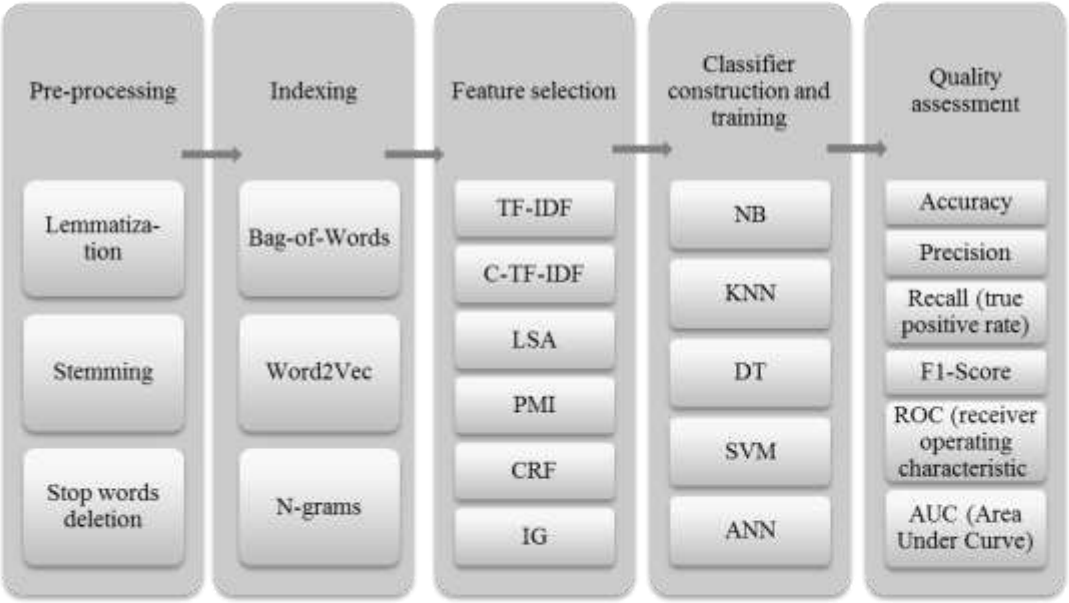

Figure 2: Accepted stages of the automatic text classification process

C-TF-IDF can be best explained as a TF-IDF formula adopted for multiple classes by combining all documents for each class. This way, each class is transformed into a single document rather than a set of documents.

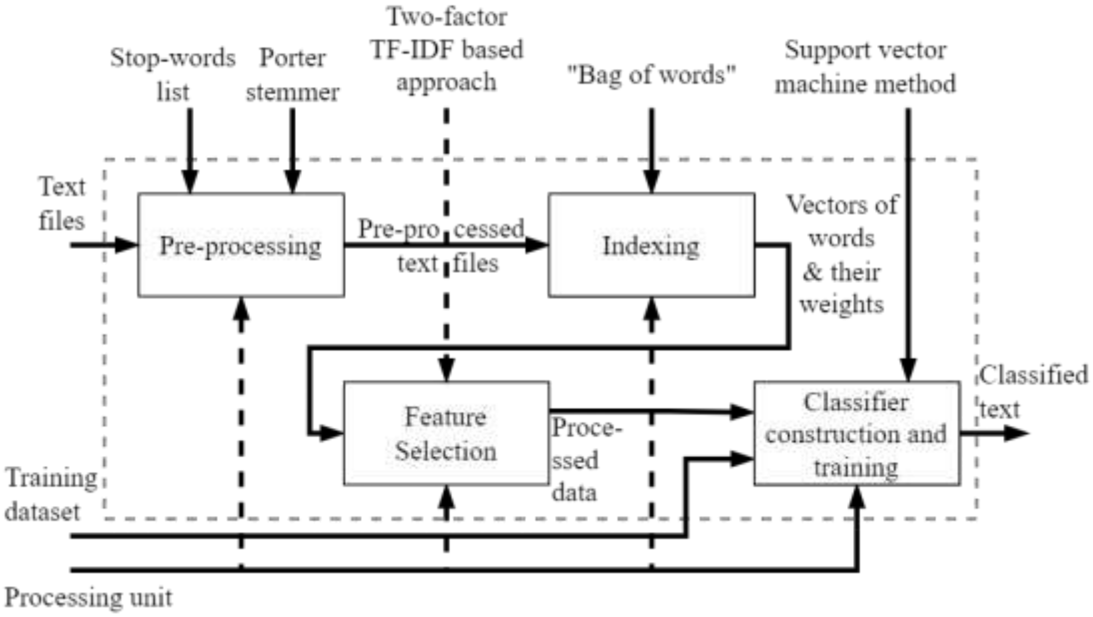
Stop-words list
Porter stemmer
Two-factor TF-IDF based approach
"Bag of words"
Support vector machine method
Text files
Pre-processing
Pre-pro cessed text files
Indexing
Vectors of words & their weights
Feature Selection
Proce-ssed data
Classifier construction and training
Classified text
Training dataset
Processing unit

Figure 3: A conceptual model of text classification based on two-factor selection of topic-related words

In this paper, we mainly implement four main methods: all-words (AW), all-words with corpus-based abbreviation (AWP), all-words with class-based keyword selection (AWK), and two-stage feature selection with both abbreviation and keyword selection (AWPK).

The AW method is a basic method that uses the standard bag of words with all the words in the feature vector.

The bag of words is a useful tool that is used for various purposes, such as classifying texts as spam/not spam, determining the similarity of texts, and as a simplified way to represent texts for various machine learning tasks in a pre-processing stage. The bag of words shows words founded in the text, but it does not take into account their order and semantics. It can be regarded as a shortcoming of the method. Text arrays' classification taking into account the semantics of the text is available to the majority of modern intellectual models, but, their use requires a powerful local computing resource or a certain cost of renting a computing server.

AWP takes into account all the words in the document collection, but filters them using a pruning process. This method filters out terms that occurs less than a certain threshold value in the entire training set. We call this threshold value the pruning level (PL). PL = n (n≥1) indicates that terms that appear at least n times in the training set are used in the decision vector, while the rest are ignored. Note that PL=1 corresponds to the AW method (i.e., no pruning). We perform parameter tuning by analyzing different values for each dataset to achieve optimal PL values for the AWP method. We conduct experiments with different levels of cropping from 2 to 30: 2, 3, 5, 8, 13, 20, i 30.

In the AWK method, separate keywords are selected for each class. This method gives equal weight to each class during the keyword selection phase. We experiment with five different numbers of keywords (250, 500, 1000, 2000, and 4000) and compare the results with AW, which includes all words as objects in the decision vector.

The AWPK method is designed to be an optimal combination of AWP and AWK by varying the level of pruning and the number of keyword parameters. The values of the parameters that give the best results in the basic methods are used for AWPK experiments.

4.1. Performing the experiment

Based on the methods discussed in the previous section, in this one we determine the optimal parameter values (pruning level and number of keywords) for the methods in all datasets. The experiments were evaluated and the methods were compared with respect to the micro-average F-measure (MicroF), which is the average success rate of documents, and the macro-average F-measure (MacroF), which is the average success rate of categories [21].

For the three datasets, we analyzed the relationships between:

- keyword frequency vector and classification accuracy;
- size of text collections and quality of classification;
- classification methods and text data sets.

As a result of the experiments, the impact of reducing the vector of keyword frequencies in the text on the accuracy of text classification should be assessed, and the impact of choosing a range of keywords according to the TF-IDF metric on the quality of classification should be analyzed.

The study of the influence of the choice of keyword rank according to the TF-IDF metric on the quality of classification is the second experiment to be conducted on three text collections.

In this paper, we use three well-known datasets from the UCI Machine Learning Repository: Reuters-21578 (Reuters), National Science Foundation Research Award (NSF) abstracts, and Mini 20 newsgroups (MiniNg20). These datasets have different characteristics that can be crucial for classification performance. Skewness is one of the key properties of a dataset, which is defined as the distribution of the number of documents across classes. A dataset that has a low skewness coefficient indicates that it is a balanced dataset with approximately the same number of document samples for each class. The validity of multiple classes for documents (indicating that a document can belong to more than one topic), document length (e.g., short abstracts or long news articles), split proportions (training and test sets), level of formality (e.g., formal journal documents or informal Internet forum posts) are other properties of datasets.

In our experiments, we use standard partitions of the Reuters dataset (the dataset contains structured information about news feed articles that can be categorized into several classes, which creates a multiple label problem. The collection consists of 21,578 documents) and MiniNg20 (informal, with many grammatical errors, allows only one topic per text, and is a balanced dataset containing the same number of messages for each topic. The MiniNg20 dataset consists of 2000 messages). For NSF (the NSF dataset consists of 129,000 abstracts describing NSF awards for basic research from 1990 to 2003. The level of formality of the dataset is high. The NSF is not a perfectly balanced dataset, but its skewness coefficient is also not as high as Reuters. The length of the document is short due to its abstract content) data related to the year 2001 were randomly selected, and five sections were chosen from this year (four sections for training and one section for testing). We create five different splits, repeat all tests with them, and take the average as the final result.

4.2. A method for selecting topic-related words based on word frequency

In this experiment, the AWP method was implemented with several PL values (PL=1 corresponds to AW) for the three datasets. Table 2 shows the feature number and the "micro" and "macro" success rates for each reduction level. The first column of the table shows the method and the value of the PL parameter, separated by a comma. As it can be seen, the reduction process improves the success rate of the classifier, and the best results (high accuracy with low feature numbers) are obtained at approximately PL=13 consistently across all three datasets with two different performance parameters.

Table 2
AWP success rates (optimal results are highlighted in bold)

Method, Parameter	Reuters			NSF			MiniNg20		
	Featu-re#	MicroF	MacroF	Featu-re#	MicroF	MacroF	Featu-re#	MicroF	MacroF
AW	20292	85.58	43.83	13424	64.46	46.11	30970	46.42	43.44
AWP,2	12959	85.55	43.84	8492	64.41	46.21	13102	49.73	47.13
AWP,3	9971	85.52	43.93	6328	64.62	46.42	9092	49.64	47.19
AWP,5	7168	85.51	44.56	4528	64.86	46.49	6000	51.26	48.52
AWP,8	5268	85.73	44.91	3376	64.66	46.38	4169	52.48	49.90
AWP,13	**3976**	**85.84**	**44.85**	**2478**	**64.58**	**46.49**	**2863**	**53.62**	**51.02**
AWP,20	3046	86.02	44.55	1875	64.23	46.67	2025	53.78	51.02
AWP,30	2237	81.29	43.59	1419	63.84	46.21	1384	52.89	50.46

Following the generalization that words that occur less than 10 to 15 times in a dataset are likely not a good indicator for text classification, we found PL=13 in the reduction-based experiments. This result indicates that the common belief in the literature that a reduction level of 2 to 3 times is sufficient to eliminate uninformative terms is not true.

4.3. A method for selecting topic-related words based on determining the threshold of a frequency dictionary

In this experiment, the performance of the AWK method was analyzed using different parameters of the keyword (function) number. The results are shown in Table 3. The success rates for AW are also included in the table for comparison.

Table 3
AWK success rates (optimal results are highlighted in bold)

Method, Parameter	Reuters		NSF		MiniNg20	
	MicroF	MacroF	MicroF	MicroF	MacroF	MicroF
AWK,250	83.69	51.15	62.04	49.51	56.65	55.72
AWK,500	84.71	50.92	62.92	49.31	56.16	55.01
AWK,1000	85.16	51.72	64.69	49.33	53.68	52.17
AWK,2000	**85.58**	**52.03**	**65.19**	**49.31**	**54.04**	**52.10**
AWK,4000	**85.84**	**52.10**	**65.71**	**49.35**	**55.25**	**53.73**
AW	85.58	43.83	64.46	46.11	46.42	43.44

In general, the AWK method with the number of keywords from 2000 to 4000 increases the success rate in all datasets compared to the AW method. Therefore, it can be concluded that using a specific set of keywords for each class gives more successful results than using all the words in the feature vector.

When we analyze the results of AWP and AWK together, we see that the improvement of AWP over AW is clear in the balanced dataset (MiniNg20), while the improvement in the distorted datasets (Reuters and NSF) is smaller. On the other hand, the improvement of AWK over AW is more significant than the improvement of AWP in all datasets. This performance gain is more pronounced in the MacroF measure. In corpus-based approaches, documents of rare classes tend to be misclassified because the words of the predominant classes dominate the feature vector.

Figure 4 and Figure 5 show the results of microF and macroF, respectively, for the class-based and corpus-based approaches with TF-IDF representation of the document using all words and keywords in the range from 10 to 2000.

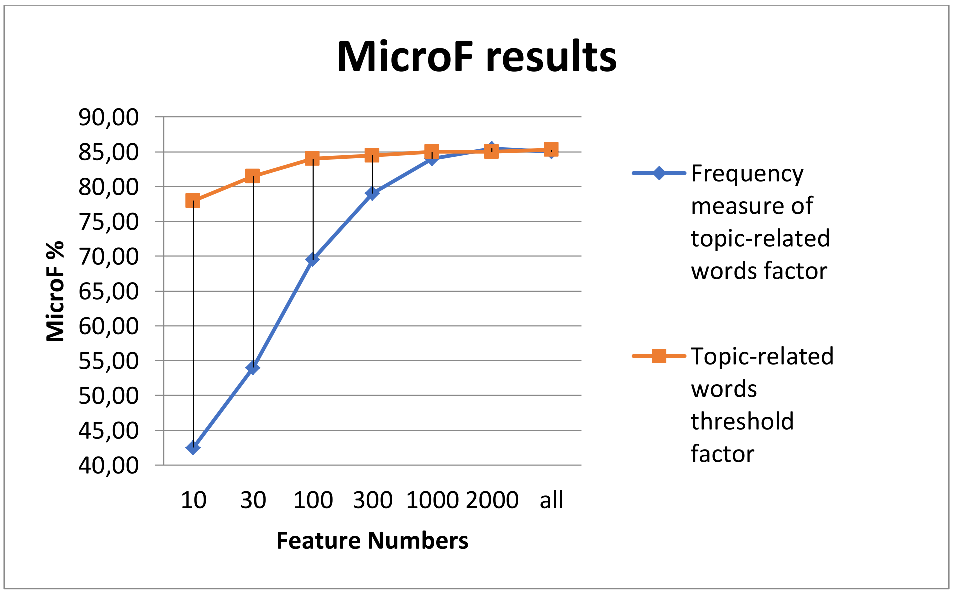

Figure 4: MicroF for two factors – frequency measure of topic-related words and topic-related words threshold in the frequency dictionary

Regarding the microF results (Figure 4), we can conclude that the class-based feature selection achieves a higher microF than the corpus-based approach for a small number of keywords. In text classification, most of the learning takes place with a small but important portion of keywords for a class. Class-based feature selection, by definition, focuses on this small portion; on the other hand, the corpus-based approach finds common keywords that apply to all classes. So, with a small number of keywords, the class-based approach is much more successful in finding more important class keywords. The corpus-based approach is not successful with such a small portion, but has a steeper learning curve that reaches a peak value of 86% in the experiments with 2000 corpus-based keywords.

For the macroF results (Figure 5), we analyzed that the class-based feature selection provides consistently higher macroF performance than the corpus-based approach. High asymmetry in the distribution of classes in the dataset negatively affects the macroF value, since macroF gives equal weight to each class rather than to each document, and documents of rare classes are more likely to be misclassified. Accordingly, the average value of correct class classifications drops sharply for datasets with many rare classes. Class-based feature selection is very useful for this asymmetry. As mentioned above, even with a small fraction of the words (e.g., 100), the class-based TF-IDF method achieves a 50% success rate, which is much better than the 43.9% success rate of TF-IDF with all words.

Rare classes are successfully characterized by class-based feature selection because each class has its own keywords for the categorization problem. The corpus approach performs worse because most of the keywords are selected from the predominant classes, which does not allow rare classes to be fairly represented by their keywords.

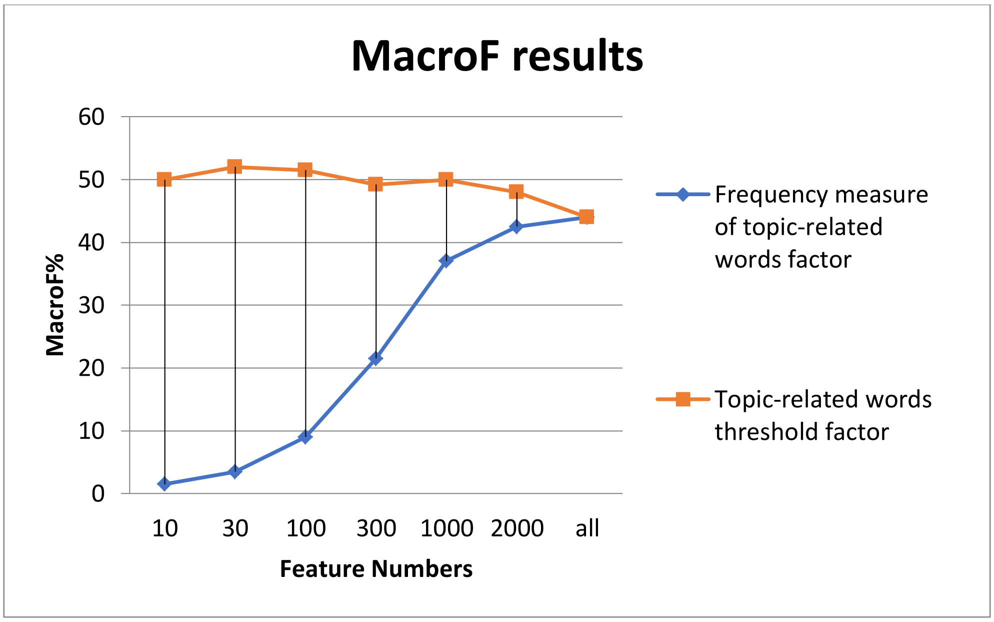

Figure 5: MacroF for two factors – frequency measure of topic-related words and topic-related words threshold in the frequency dictionary

MacroF gives equal weight to each class when determining the success of a classifier. Thus, especially for highly distorted datasets, where rare classes are poorly represented by the selected features, the average value of correct classifications for rare classes drops significantly. This is true for both AW and AWP in skewed datasets that use a common feature set for all classes. However, with class-based keyword selection, since each class has its own keywords during classification, sparse classes are characterized more successfully. Thus, we observe a significant increase in the success rate (MacroF) with AWK in the skewed datasets.

4.4. Combined method of two-factor selection of topic-related words

The AWPK method combines the optimal patterns of using the AWP and AWK approaches. Therefore, the parameters of the method are the reduction level and the number of keywords. In this experiment, we use the optimal values of these parameters determined in the previous analyzes for each dataset: a reduction level of 13 and the number of keywords 2000 and 4000. The results are shown in Table 4. The table also shows the best performing AW, AWP, and AWK for comparison.

As can be seen from the table, the two-factor feature selection approach outperforms the previous approaches. Selecting the best 2000-4000 keywords for each class with an initial reduction step significantly improves the best AWP (with PL=13) and AWK (with 2000-4000 keywords) performance in all three datasets.

Table 4
AWPK success rates (optimal results are highlighted in bold)

Method, Parameter	Reuters		NSF		MiniNg20	
	MicroF	MacroF	MicroF	MacroF	MicroF	MacroF
AWPK,13,2000	**86.40**	**53.95**	**66.06**	**50.11**	**57.43**	**55.66**
AWPK,13,4000	**86.70**	**53.98**	**66.10**	**50.12**	**57.43**	**55.66**
AW	85.58	43.83	64.46	46.11	46.42	43.44
AWP,13	85.84	44.85	64.58	46.49	53.62	51.02
AWK,2000	85.58	52.03	65.19	49.31	54.04	52.10
AWK,4000	85.84	52.10	65.71	49.35	55.25	53.73

The diagram (Figure 6) shows a comparison of the three methods by micro-averaged F-measure, with the AWPK method performing better than the others, especially in the Reuters set.

When comparing the three methods by macro-average F-measure in Figure 7, it can be seen that the AWPK method is better than the others, but the best performance is already with the MiniNg20 dataset. Also, due to the heterogeneity of the data, we can see different micro-average and macro-average F-measure values for different datasets.

Hence, we can conclude that the additional effect of corpus-based shortening is extended when it is combined with the class-based TF-IDF keyword selection metric. As a consequence, the method proposed in this paper, AWPK, gives the best performance. The significance of the results for the three methods was measured using a statistical feature test. We noticed that in general, each method outperforms its predecessor. In this sense, AWP and AWK are significantly better than the standard AW method, and AWPK is significantly better than both AWP and AWK. Thus, the most advanced method in this study (AWPK) is the optimal method with two-factor feature selection analysis.

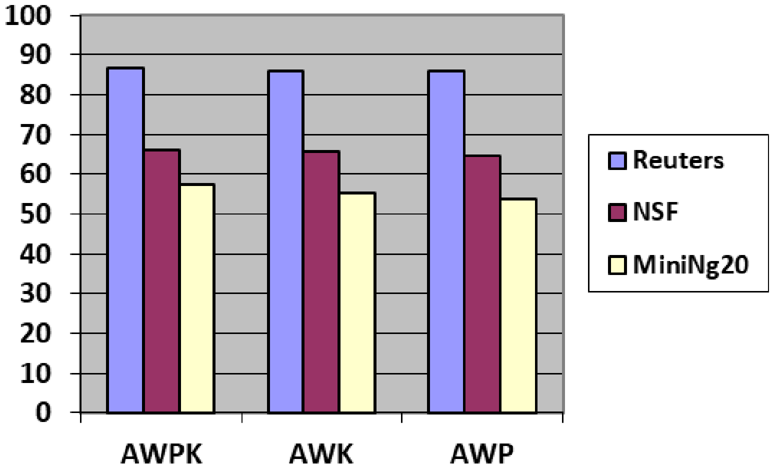

Figure 6: MicroF comparison chart of the three methods

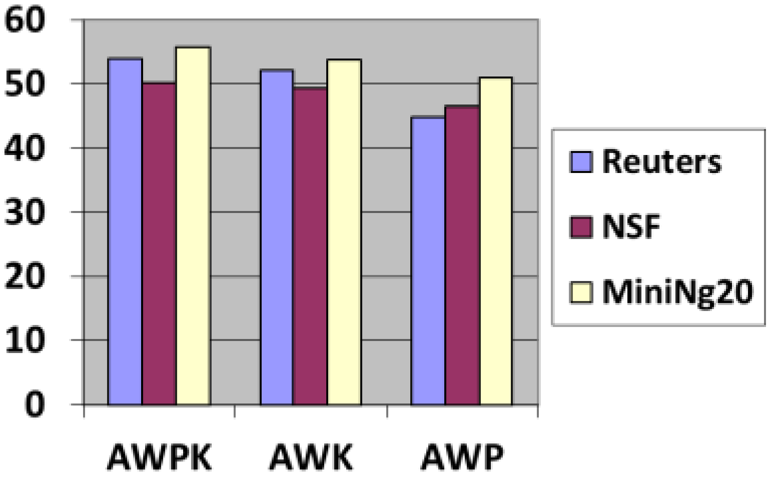

Figure 7: MacroF comparison chart of the three methods

The scientific novelty of the research is in creating a flexible method for extracting topic-related words from a frequency dictionary to increase further SVM-classification accuracy and to reduce the redundancy of the sample. To do this, a study of the classification accuracy was carried out using different values of the occurrence of terms in the dictionary (the value of the occurrence of words in the sample, equal to 13, was determined empirically), as well as a different threshold of words in the dictionary (the threshold, equal to 4000 words, was determined empirically). By combining the performed studies, a modified method for determining topic-related words was proposed, increasing the classification accuracy by 4%.

5. Conclusion

The paper proposes a conceptual model of text classification based on accepted stages of the automatic text classification process with modification in a feature selection module. The method of topic-related words selection was improved by combining two factors – frequency measure of topic-related words and topic-related words threshold in the frequency dictionary.

Achieving this goal was possible due to the analysis of the dependence of the classification accuracy of the Reuters-21578, NSF and MiniNg20 datasets on the choice of topic-related words of the frequency dictionary built on the basis of the TF-IDF method.

The first study of the selection of topic-based words for classification based on the frequency of words showed that for the analyzed data set the most informative words are those that occur at least 10 to 15 times in the data set. The second study of the selection of topic-related words based on the reduction of the frequency vector by determining the threshold of the frequency dictionary showed that using the range of topic-related words from 2000 to 4000 for all datasets gives more successful results than using all words in the feature vector.

Then, after determining the optimal parameter values for each method (with the highest micro-F and macro-F measures), a new two-factor method was proposed, which is a combination of these two approaches. The proposed combined method of two-factor selection of topic-related words outperforms the previous approaches for all three datasets and increases the accuracy of text document classification from 2 to 4 percent.

Possible future work is to apply the two-factor feature selection approach to more semantically oriented text classification methods, such as methods that use language models, linguistic features, or lexical dependencies. Integrating the concepts of keyword reduction and keywords number selection into these methods as two serial steps can lead to higher classification performance.

6. References

[1] O. Barkovska, Information Object Storage Model with Accelerated Text Processing Methods, in: D.Pyvovarova, V. Kholiev, H. Ivashchenko, D. Rosinskyi (Eds.), Proceedings of the : 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS 2021), volume 1 : Main Conference, Lviv Ukraine, 2021, pp. 286-299.

[2] D. Panchenko, Ukrainian News Corpus as Text Classification Benchmark. In: D. Maksymenko, O. Turuta, M. Luzan, S. Tytarenko, O. Turuta (Eds.), ICTERI 2021 Workshops. ICTERI 2021. Communications in Computer and Information Science, vol 1635. Springer, Cham., Kherson Ukraine, 2021, pp. 550-559. https://doi.org/10.1007/978-3-031-14841-5_37.

[3] E. Erdem. Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning. Journal of Artificial Intelligence Research 73 (2022) 1131-1207. doi: 10.1613/jair.1.12918

[4] M. Mirończuk, J. Protasiewicz, A Recent Overview of the State-of-the-Art Elements of Text Classification. Expert Systems with Applications 106 (2018) 36-54. doi:10.1016/j.eswa.2018.03.058.

[5] W. Cunha, V. Mangaravite, C. Gomes, S. Canuto, E. Resende, C. Nascimento, Cecilia & F. Viegas, C. França, W. Martins, T. Couto, L. Rocha, M. Gonçalves, On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study, in: Information Processing & Management, 58, 2021. doi:10.1016/j.ipm.2020.102481

[6] M. Malekzadeh, P. Hajibabaee, M. Heidari, S. Zad, O. Uzuner and J. H. Jones, Review of Graph Neural Network in Text Classification, in: IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 2021, pp. 0084-0091, doi:10.1109/UEMCON53757.2021.9666633.

[7] M. Shervin, Nal Kalchbrenner, E. Cambria, Narjes Nikzad, Meysam Asgari Chenaghlu, Jianfeng Gao, Deep Learning based Text Classification, ACM Computing Surveys (CSUR), 54, 2020: doi:10.1145/3439726.

[8] C. Liu, Y. Sheng, Z. Wei and Y. Yang., Research of Text Classification Based on Improved TF-IDF Algorithm, in: 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE), IEEE, Lanzhou, China, 2018, pp. 218-222, doi:10.1109/IRCE.2018.8492945.

[9] A. I. Kadhim, Term Weighting for Feature Extraction on Twitter: A Comparison Between BM25 and TF-IDF, 2019 International Conference on Advanced Science and Engineering (ICOASE), Zakho - Duhok, Iraq, 2019, pp. 124-128, doi:10.1109/ICOASE.2019.8723825.

[10] T.V. Batura, Automatic text classification methods, Software & Systems. 1(30) 2017: 85–99. doi:10.15827/0236-235X.117.085-099

[11] Jiang, Mingyang, Yanchun Liang, Xiaoyue Feng, Xiaojing Fan, Zhili Pei, Yu Xue and Renchu Guan. “Text classification based on deep belief network and softmax regression.” Neural Computing and Applications 29 (2016): 61-70. doi:10.1007/s00521-016-2401-x

[12] Chen W., Xie X., Wang J., Pradhan B., Hong H., Bui D.T., Duan Z., Ma J. “A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility.” CATENA 151 (2017) 147–160. doi:10.1016/j.catena.2016.11.032

[13] L.M. Manevitz, M. Yousef. “One-class SVMs for document classification.” Journal of Machine Learning Research 2 (2001): 139–154.

[14] Gary Marchionini. “Exploratory search: from finding to understanding.” Communication of the ACM 49, 4 (2006): 41–46. doi:10.1145/1121949.1121979.

[15] B. Choudhary, Text clustering using semantics, In: P. Bhattacharyya (Ed.), Proceedings of the 11th International World Wide Web Conference, 2002, pp. 1-4.

[16] M. R Utomo, AText classification of british english and american english using support vector machine, In: Y. Sibaroni, 7th International Conference on Information and Communication Technology (ICoICT), IEEE, 2019. pp. 1-6. doi: 10.1109/ICoICT.2019.8835256.

[17] J. Cervantes, F Garcia-Lamont, L. Rodríguez-Mazahua, A. Lopez. “A comprehensive survey on support vector machine classification: Applications, challenges and trends.” Neurocomputing, 408 (2020): 189-215. doi:10.1016/j.neucom.2019.10.118.

[18] K.Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, Text classification algorithms: A survey.” Information 10.4 (2019. doi:10.3390/info10040150.

[19] A. P. Pimpalkar, R. J. R Raj. “Influence of pre-processing strategies on the performance of ML classifiers exploiting TF-IDF and BOW features.” ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal 9.2 (2020): 49-68. doi: 10.14201/ADCAIJ2020924968.

[20] Y.Cohen-Kerner, D. Miller, Y.Yigal. “The influence of preprocessing on text classification using a bag-of-words representation.” PloS one, 15.5 (2020). doi:10.1371/journal.pone.0232525.

[21] J.C. Lamirel, P. Cuxac, A.S. Chivukula et al. “Optimizing text classification through efficient feature selection based on quality metric.” Journal of Intelligent Information Systems 45 (2015): 379–396. doi:10.1007/s10844-014-0317-4.