Text Classification using Term Co-occurrence Matrix

Tetiana Kovaliuk, Iryna Yurchuk, Kseniia Dukhnovska, Oksana Kovtun and Anastasiia Nikolaienko

Taras Shevchenko National University of Kyiv, Bohdan Hawrylyshyn str. 24, Kyiv, UA-04116, Ukraine

Abstract

Among modern classification methods, the support vector machine occupies a leading place due to its strict theoretical validity. This method is used in the theory of pattern recognition, in data mining, and it is also widely used to build search engines at the stage of classifying text documents.

The article considers the support vector machine method by using a kernel based on a term co-occurrence matrix in a corpus of text documents. From fuzzy set theory, the relationship between two terms in a collection of text documents can be defined. From here it is possible to build the kernel for the method support vectors. The study proves that the co-occurrence matrix can be the kernel for the method of support vectors.

The work shows that the quality of classification for the support vector machine method on the kernel of the term co-occurrence matrix in a collection of text documents exceeds the quality for the SVM method.

Keywords [1]

Document classification, text corpus, text documents, term co-occurrence matrix, kernel function, support vector machines.

1. Introduction

Text documents that are saved in electronic storages of global or corporate networks are the basis for decision-making in government, scientific, and educational institutions, and determine the success of their work as a whole. The intensive growth in the amount of text information, their universal accessibility and high dynamism leads to an excess of information and overflow with it. To overcome these problems, integrated banks of text documents are being built. During the formation of such repositories, preliminary processing of these documents is carried out for the purpose of their intellectual analysis. One of the most important stages of preliminary processing of text documents is their classification.

Among modern classification methods, the support vector machine occupies a leading place due to its strict theoretical validity. This method is used in the theory of pattern recognition, in data mining, and it is also widely used to build search engines at the stage of classifying text documents. But algorithms from this family are characterized by the problem of scalability – high resource consumption of memory and computation time at the training stage. This paper presents a strategy for improving the performance of support vector machines by applying a kernel based on a term co-occurrence matrix across multiple text documents. As a result of this strategy, the weight of more informative features and more informative feature combinations increases, which makes the classifier faster and less resource-intensive.

The purpose of the research is to improve the performance of the support vector machine method by using a kernel based on a term co-occurrence matrix in a corpus of text documents.

Dynamical System Modeling and Stability Investigation (DSMSI-2023), December 19-21, 2023, Kyiv, Ukraine

EMAIL: tetyana.kovalyuk@knu.ua (T. Kovaliuk); i.a.yurchuk@gmail.com (I. Yurchuk); duchnov@ukr.net (K. Dukhnovska); kovok@ukr.net (O. Kovtun); n_nastja@ukr.net (A. Nikolaienko)

ORCID: 0000-0002-1383-1589 (T. Kovaliuk); 0000-0001-8206-3395 (I. Yurchuk); 0000-0002-4539-159X (K. Dukhnovska); 0000-0003-0871-5097 (O. Kovtun); 0000-0002-2402-2947 (A. Nikolaienko)

2. Related works

Artificial intelligence systems are based on research in text mining. However, the linguistic topic is inherently complex, necessitating the development of new models of text documents and innovative approaches to classification and clustering of text documents. In the scientific study [1], it is shown that a text document can be represented as a vector.

A new model for text documents is introduced in [2]. This model is based on the concept of fractals to deepen the complexity of language. This approach reduces potential noise. Additionally, the authors introduce an innovative activation function to enhance the performance of the neural network. The results of this study are validated through real technical reports.

Text analysis is the process of extracting and interpreting information from a collection of text documents to identify and describe their key characteristics and features. It is a crucial tool in the field of linguistics, enabling the comprehension of text structure, content, features, and main ideas. The purpose of text analysis is to uncover the content, context, and linguistic and stylistic features of the text. The meaning of text analysis is to identify entities (terms). As stated in [3], the purpose of entity recognition is to automatically identify expected knowledge from text. For example, in [4, 5] algorithms for detecting technical terms are considered.

In [6], a text document is considered as a vector of features:

$$T = [c_1, c_2, \dots, c_n], \tag{1}$$

and entities are recognized in the form of:

$$[I_b, I_e, \omega, \lambda], \tag{2}$$

where ω denotes a specific entity; b and e are indexes that highlight the span from I_b to I_e, thus specifying the location of the entity; λ represents the type of the entity.

Another task of text analysis is the classification of text documents. The classification of text documents can be performed using various algorithms and models. For example, in [7, 8], it is proposed to assign a category label to the text document T, such as "acceptable, tolerable, investigated, and corrected" for risk assessment.

In the work [9], an ensemble classification method is proposed for multi-label classification of text documents. The method combines the random forest algorithm and the semantic vector space of hidden semantic core co-occurrence. The work shows that random word segmentation increases the diversity of integration and obtains another orthogonal projection of the low-dimensional space of hidden semantics.

Latent Dirichlet Allocation (LDA) is often used to define classes. In the article [10], an algorithm for clustering short emotional texts based on the LDA algorithm is proposed. The text document is presented in TF-IDF format. In addition, thematic word pairs and topic relation words are extracted and inserted into the LDA model for clustering. This approach allows for more accurate semantic information to be found. The results of this work can be successfully used to analyze texts published on social media.

The Support Vector Machine (SVM) method is widely used for classification. In the work [11] proposes a novel hybrid approach that leverages a gray level co-occurrence matrix and the SVM classifier to achieve highly accurate segmentation and classification of malignant and benign cells in breast cytology images under severe noise conditions. In [12], SVM is investigated for sentiment analysis. In order to improve the classification accuracy for the SVM method, this work uses the particle swarm method and genetic algorithm.

In [13], the author of a text document is determined using classification. Here, several methods for classifying text documents written by several authors are compared. The work compares the results of classification based on the following methods: artificial neural networks, multi-expression programming, k-nearest neighbor, support vector machines, and decision trees with C5.0.

Multi-view support vector machines are investigated in [14] to address the problems of multi-view image classification. The work proposes to introduce a fuzzy assessment to assign a weight to each sample from multiple images. This assessment combines membership and non-membership functions, which provides an efficient mechanism for assigning weight coefficients to collections of images with multiple representations.

Recently, researchers have been investigating term-document matrices. For example, in [15, 16], the relationships between terms and their use in text documents are studied. Here, the input data is a sequence of events when terms coincide in documents. Taking into account the term-document matching stream, latent vectors of terms and documents are studied. The goal of this study is search optimization. For this purpose, the work proposed a dimensionality reduction algorithm for adaptively learning the latent semantic index of terms and documents in a collection. The results of this study demonstrate improvements in search performance compared to the baseline method.

The application of network analysis to corpus linguistics is introduced in [17]. The authors conducted a comprehensive initial study involving various practical analyses, including frequency, keyword, collocation, and cluster analysis. The work proposes a novel procedure capable of extracting diverse intertextual and intratextual aspects from the analyzed documents. This procedure captures the existing connections between different elements of the corpus, enabling a deeper understanding of the relationships and dynamics within the sets of texts.

The term-document matrix was used in [18] to detect and track topics in a collection of text documents. In this work, a hierarchical non-negative matrix factorization method was proposed for creating topic hierarchies from text collections. The proposed method can dynamically adjust the topic hierarchy to adapt to emerging, developing, and fading processes [19]. The work proves that such an approach can achieve better performance with competitive time savings.

3. Problem definitions

Any text document can be described as a tuple:

$$D_i = \langle t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{im}, p_{i1}, p_{i2}, \dots, p_{ir}, \dots, p_{ih} \rangle,\ \ i = \overline{1,n},\ \ j = \overline{1,m},\ \ r = \overline{1,h}, \tag{3}$$

where n is the number of text documents in the corpus, t_{ij} is a statistical measure of the importance of the j-term of the i-document, m is the power of the dictionary, p_{ir} is additional attributes in the description of this text document, h is the number of additional attributes.

Corpus is a collection of text documents. Terms (concepts) are the names of mental images that are transferred in the process of information exchange. The terms are contained in the dictionary. A statistical measure of the importance of a term is the ratio of the number of occurrences of a term in a text document to the number of all terms in the text document. Additional features (p_{ir}) can include the creation date of the text document, its author, address, links to other text documents, etc.

The vector D_i represented by the tuple (3) is called the profile of this text document. The classification of text documents consists in dividing a set of these resources into non-intersecting groups in order to ensure the minimum difference between the resources of one group corresponding to a certain content topic, and the maximum difference between the resources of other groups.

Let there be a set of text documents: $D = \{D_i |\ i = \overline{1,n}\}$ and set of classes are given $Q = \{Q_k |\ k = \overline{1,n_c}\}$. Each class Q_k is described by some structure $F_k = \{D_{k1}, D_{k2}, \dots, D_{kl}\}$. The classification procedure f consists of performing some transformations on the profile of a text document, after which a conclusion is made about the correspondence of the resource D_i to one of the structures $F_k, f: D \to Q$.

3.1. The relationship between terms in a document

If we neglect additional parameters, then the set of text document profiles can be represented as follows:

$$\begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & t_{22} & \cdots & t_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{n1} & t_{n2} & \cdots & t_{nm} \end{pmatrix}. \tag{4}$$

From fuzzy set theory, the relationship between two terms in a collection of text documents can be defined as:

$$c_{ij} = \frac{\tilde{n}_{ij}}{\tilde{n}_i + \tilde{n}_j - \tilde{n}_{ij}}, \tag{5}$$

where $\tilde{n}_i$ is the number of text documents that contain the i-term, $\tilde{n}_j$ is the number of text documents that contain the j-term, $\tilde{n}_{ij}$ is the number of text documents that contain both terms.

However, (5) neglects the frequency of term occurrence in the document. The relationship coefficient c_{ij} can have the same value for terms that carry the main content of the document and for unimportant terms. To overcome this drawback in (5), we will introduce a normalized term frequency in the document t_{ij}:

$$c_{ij} = \frac{\frac{1}{2}\sum_{i_1=1}^{\tilde{n}_{ij}} \left(t_{ii_1} + t_{ji_1}\right)}{\sum_{i_1=1}^{\tilde{n}_i} \left(t_{ii_1}\right) + \sum_{i_1=1}^{\tilde{n}_i} \left(t_{ji_1}\right) - \frac{1}{2}\sum_{i_1=1}^{\tilde{n}_{ij}} \left(t_{ii_1} + t_{ji_1}\right)}, \tag{6}$$

where $\tilde{n}_i$ is the number of text documents that contain the i-term, $\tilde{n}_j$ is the number of text documents that contain the j-term, $\tilde{n}_{ij}$ is the number of text documents that contain both terms.

As a result, the matrix of relationships between terms will take the form:

$$C = \begin{pmatrix} 1 & c_{12} & \cdots & c_{1m} \\ c_{21} & 1 & \cdots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \cdots & 1 \end{pmatrix}. \tag{7}$$

In essence, it is a correlation matrix of two terms in a corpus, also known as a term co-occurrence matrix. Its coefficients indicate a statistical dependence of the frequencies of two terms, and changes in the values of one or more of these quantities lead to a systematic change in the values of another or other quantities.

3.2. Property of term co-occurrence matrix

Property 1. The term co-occurrence matrix C (7) in a corpus is positive definite.

Proof: According to Sylvester's criterion: a symmetric matrix is positive definite if and only if its leading minors are positive. The proof is based on the use of the Gauss method and reducing matrix (7) to triangular form (taking into account that the matrix is symmetric and its elements are positive numbers). With such transformations, the values of the main minors will not change and will be equal to the product of their diagonal elements.

Property 2. The co-occurrence matrix of terms in a corpus is quadratic.

Statement 1. For the term co-occurrence matrix, there is a matrix B, which is represented as $B = C^{\frac{1}{2}}$.

Proof: From Schur's lemma it follows that if a matrix C is symmetric, then there is an orthogonal matrix S, the columns of which are eigenvectors of the matrix C, and a diagonal matrix V, the elements of which are the eigenvalues of the matrix C, such that:

$$V = S^{-1} \cdot C \cdot S, \tag{8}$$

All eigenvalues of a positive definite matrix are positive. Hence, all non-zero elements of the diagonal matrix V are greater than 0, which means there exists $V^{\frac{1}{2}}$:

$$V = \begin{pmatrix} v_{11} & 0 & \cdots & 0 \\ 0 & v_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & v_{nn} \end{pmatrix}, \quad B = V^{\frac{1}{2}} = \begin{pmatrix} \sqrt{v_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{v_{22}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{v_{nn}} \end{pmatrix}. \tag{9}$$

If both sides of expression (8) are multiplied on the left by S, and on the right by S^{-1}(this is possible as $SS^{-1}=S^{-1}S=E$– identity matrix), then:

$$C = S \cdot (S^{-1} \cdot C \cdot S) \cdot S^{-1} = S \cdot V \cdot S^{-1} = S \cdot V^{\frac{1}{2}} \cdot V^{\frac{1}{2}} \cdot S^{-1} = \\ = \left(S \cdot V^{\frac{1}{2}} \cdot S^{-1}\right) \cdot \left(S \cdot V^{\frac{1}{2}} \cdot S^{-1}\right) = B \cdot B = B^2. \quad (10)$$

A function $K: XxX \to R$ is called a kernel if it is represented in the form $K(x, x') = [\varphi(x), \varphi(x')]$ under some mapping $\varphi: X \to F$, where F is the space with scalar product.

Let $\varphi(d) = C^{\frac{1}{2}} \cdot d$, then its kernel:

$$K(d, d_1) = d^T \cdot C \cdot d_1, \quad (11)$$

Statement 2. The function defined by expression (11) is the kernel.

Proof:

$$K(d, d_1) = d^T \cdot C \cdot d_1 = d^T \cdot \sqrt{C}^T \cdot \sqrt{C} \cdot d_1 = [\varphi(d), \varphi(d_1)].$$

The term co-occurrence matrix is the kernel.

3.3. Support vector machines

There are many approaches to solving classification problems – this is a probabilistic approach (for example: the naive Bayes method and its modifications), an algebraic approach (through various measures of proximity of text document profiles: Euclidean distance and its modifications, Manhattan distance, Mahalanobis distance, etc.). Today, the support vector machine is a classification method, the results of which are rated as one of the most effective. It should be noted that the support vector machine considers the problem of binary classification. If there are a large number of classes in a corpus, then the classification problem can be solved in a way in which each class is separated from all the others. In this case, each binary problem does not depend on the others, and they can be solved in parallel on different machines.

Support vector machines are a set of supervised learning classification algorithms. To implement this method, each text document is represented as a point in N-dimensional space. The classes will be determined by the clusters of these points. A separating hyperplane is drawn between these classes. That is, a hyperplane is constructed such that the distance between the two closest points from different classes is maximum. If such a hyperplane exists, then it is called an optimal separating hyperplane.

The equation of the hyperplane has the form:

$$w \cdot x - b = 0, \quad (12)$$

where w is a support vector, i.e. a perpendicular drawn from the class point to the separating hyperplane, b is the distance from the hyperplane to the origin (Figure 1).

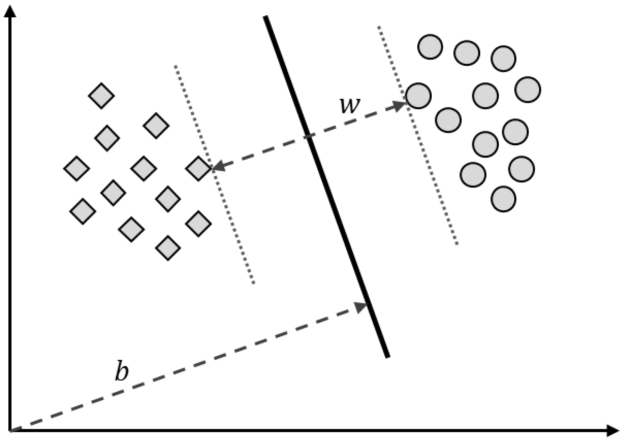

Figure 1: Visualization of the support vector machine in two-dimensional space

With respect to this hyperplane, all points of the same class lie on the same side. If we construct a hyperplane parallel to the given one and passing through the class point closest to the optimal separating hyperplane, then its equation will be $w \cdot x - b = 1$. The equation of the same hyperplane for another class is $w \cdot x - b = -1$.

Between these hyperplanes a strip is formed, which must be free from points of one and another class. To exclude all points from the strip, you need to check the condition:

$$\begin{cases} w \cdot x_i - b \geq 1, & k_i = 1 \\ w \cdot x_i - b \leq 1, & k_i = -1. \end{cases} \tag{13}$$

The problem of constructing an optimal separating hyperplane is reduced to the problem of minimizing the length of the support vector w. This is a quadratic optimization problem, which is represented as follows:

$$\begin{cases} \|w\|^2 \to \min \\ k_i(w \cdot x_i - b) \geq 1. \end{cases} \tag{14}$$

Task (14) is a mathematical programming problem. If we rewrite it in general form, we can get:

$$\begin{cases} f(x) \to \min \\ \varphi(x) \geq 0. \end{cases} \tag{15}$$

To find a solution to such a problem, the Lagrange function is composed:

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i \cdot \varphi(x), \tag{16}$$

where λ_i are the Lagrange multipliers.

According to the Kuhn-Tucker theorem, problem (14) will take the form:

$$\begin{cases} L(x, b, \lambda) = \dfrac{1}{2}\|w\|^2 - \displaystyle\sum_{i=1}^{m} \lambda_i \cdot (k_i(w \cdot x_i - b) - 1) \ \to \ \min_{w,b} \max_{\lambda} \\ \lambda_i > 0, \quad 1 < i < m. \end{cases} \tag{17}$$

Moreover, it reduces to an identical problem that contains only dual variables:

$$\begin{cases} -L(\lambda) = -\displaystyle\sum_{i=1}^{m} \lambda_i + \dfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i \lambda_j k_i k_j [x_i, x_j] \ \to \ \min_{\lambda} \\ \lambda_i > 0, \quad 1 < i < m \\ \displaystyle\sum_{j=1}^{m} k_i \lambda_j = 0. \end{cases} \tag{18}$$

If the problem is solved, then w and b can be found using the formulas:

$$w = \sum_{i=1}^{m} \lambda_i \cdot k_i \cdot x_i\,, \qquad b = w \cdot x_i - k_i. \tag{19}$$

As a result, the classification algorithm can be written as:

$$a(x) = \operatorname{sign}\left(\sum_{i=1}^{m} \lambda_i k_i [x_i, x] - b\right). \tag{20}$$

Modified support vector machines contain arbitrary kernels instead of scalar products, which eliminates linearity. Replacing the scalar product in (20) with an arbitrary kernel we get:

$$a(x) = \operatorname{sign}\left(\sum_{i=1}^{m} \lambda_i k_i K[x_i, x] - b\right). \tag{21}$$

For a more confident and effective classification in formula (21), matrix (7) was used as the kernel. Problem (21) with kernel (7) takes the form:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{m} \lambda_i + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j k_i k_j d_j^T C d_i \quad \to \quad \min_\lambda \\ \lambda_i > 0, \qquad 1 < i < m \\ \sum_{j=1}^{m} k_i \lambda_j = 0. \end{cases} \tag{22}$$

The objective function of this problem is quadratic, and the constraint is linear functions, so this problem is classified as a quadratic programming problem. The most popular methods for solving problems of this class include gradient methods. Their use in the general case allows one to find the local extremum point. The algorithm for solving the problem using gradient methods is that, starting from a certain point, a sequential transition is carried out to other points in the direction of the antigradient until an admissible solution to the original problem is found. When finding a solution to a problem using gradient methods, the iterative process continues until the gradient of the function at the next point becomes equal to zero or until it exceeds some infinitesimal value (the accuracy of the resulting solution).

The practical implementation of teaching this method of classifying text documents can be described by the following steps:

1. Digitization of a text document: removal of various control characters, tags, stop words and presentation of the test information document in vector form.

2. Compilation of matrix coefficients (7) for the incoming set of text documents.

3. Initial approximation: an arbitrary vector representing a document of one class is selected, and the closest vector of another class is searched for it. For this vector, the closest vector from the first class is searched, etc.

4. Solving problem (22) using the gradient descent method.

The use of support vector machines differs for the better from other methods in that this task can be parallelized. Considering the gigantic power of modern data banks, the size of the training sample should be estimated in the hundreds of thousands. This dimension makes the use of standard numerical methods of quadratic programming impossible. To date, several algorithms have been proposed to optimize such problems. One of these algorithms is the sequential optimization method (SOM). According to SOM, the minimum possible subtask is solved at each iteration. The result of this partitioning is many simple and independent subtasks, which means that their parallel calculation on different machines becomes possible.

4. Classification quality characteristics

Classification quality characteristics are divided into two error levels. A first-level error occurs if a text document is mistakenly not in the required class. Second-level errors include errors when a document is mistakenly found to be in a defined class. Let the number of documents in the test set be N, of which N_p is the number of documents correctly identified to the class, and N_n is the number of documents that are not related to the class. Then, $N = N_p + N_n$. The metric accuracy (A) is used to evaluate the general accuracy of a classifier. It is calculated by dividing the number of correctly classified instances by the total number of instances in the dataset:

$$A = \frac{N_p}{N} \cdot 100\% = \frac{N_p}{N_p + N_n} \cdot 100\%. \tag{23}$$

Let the number of false passes F_n, and false detections F_p, then the number of correct passes $T_n = N_n - F_p$ and correct detections $T_p = N_p - F_n$. The degree of precision (P) and recall (R), which are often used in information retrieval tasks, are calculated based on the T_p and F_p characteristics:

$$P = \frac{T_p}{T_p + F_p} \cdot 100\%, \qquad R = \frac{T_p}{T_p + F_n} \cdot 100\%. \tag{24}$$

Completeness measures the proportion of correct classification across all documents in a given class. Precision measures the proportion of correct detections of all identified resources. Completeness and accuracy are quantities dependent on each other. When developing the architecture of a text document classifier, you usually have to choose one of two characteristics as the dominant one. If the choice falls on accuracy, this leads to a decrease in completeness due to an increase in the number of false positive responses. An increase in completeness causes a simultaneous decrease in accuracy. Therefore, it is convenient to use one value to characterize the classifier, the so-called F1-score or Van Riesbergen measure:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}. \tag{25}$$

F1-score is one of the most common characteristics for this type of system. There are two main approaches to calculating F1-score for classification problems: total F1-score (results for all classes are summarized in one table, from which the F1-score e is then calculated) and average F1-score (for each class, its own F1-score is formed, then the arithmetic mean is calculated for all classes).

5. Research materials and results

As working material for the experiments, a test sample of text documents in two scientific disciplines was taken: information retrieval and continuum mechanics. Those. the test collection needed to be divided into two classes. Each class had approximately the same number of text documents - 200 and 220, which ensured uniformity of results - no one class stood out solely because of the number of documents in it. It should be noted that the accuracy of document definition for a particular class can greatly depend on the quality of the resources of that particular class. The total number of text documents in the sample was 420 documents. They were divided randomly into 2 equal parts of 210 documents each, maintaining approximately equal numbers of resources by class.

Training was carried out on one of these two sets, and testing was carried out on the other. Next, all documents from the training set were divided into 5 parts. Removing the first part of documents from the training set, the classifier was trained on the remaining 80 % of documents. Using the test sample, indicators of the quality of the classifier's work were determined. Then removing the second part from the training set; other values of performance indicators were calculated, etc. As a result, 5 values of classifier performance indicators were obtained. After this, the sets were swapped and the runs were repeated. Their arithmetic mean was taken as the final results. This averaging made it possible to smooth out the results, thereby making them more objective.

Software implementation of these classification methods was carried out in the Python environment. As a result of the work done, the following results were obtained (Figure 2, Table 1).

Table 1
Classifier performance report

Classification methods	Precision	Recall	F1-score
SVM	72.90 %	74.83 %	73.85 %
SVM on the kernel of the term co-occurrence matrix	85.65 %	75.06 %	79.51 %

Since the experiments were carried out on a test collection with high-quality text documents, the classification efficiency indicators are somewhat overestimated. But in any case, a classifier is considered good if its F1-score exceeds 60 %.

6. Conclusions

Testing of the support vector method and its proposed modification was carried out on the same collection, the documents underwent the same pre-processing and digitization, which gives the right to compare the performance indicators of the classification methods.

As seen from the calculation results, the classification quality of the Support Vector Machine (SVM) method using the kernel of the term co-occurrence matrix in a collection of text documents surpasses the quality achieved by the standard SVM method. Furthermore, the first algorithm demonstrates a 13% higher precision and a 6% higher F1-score in classification.

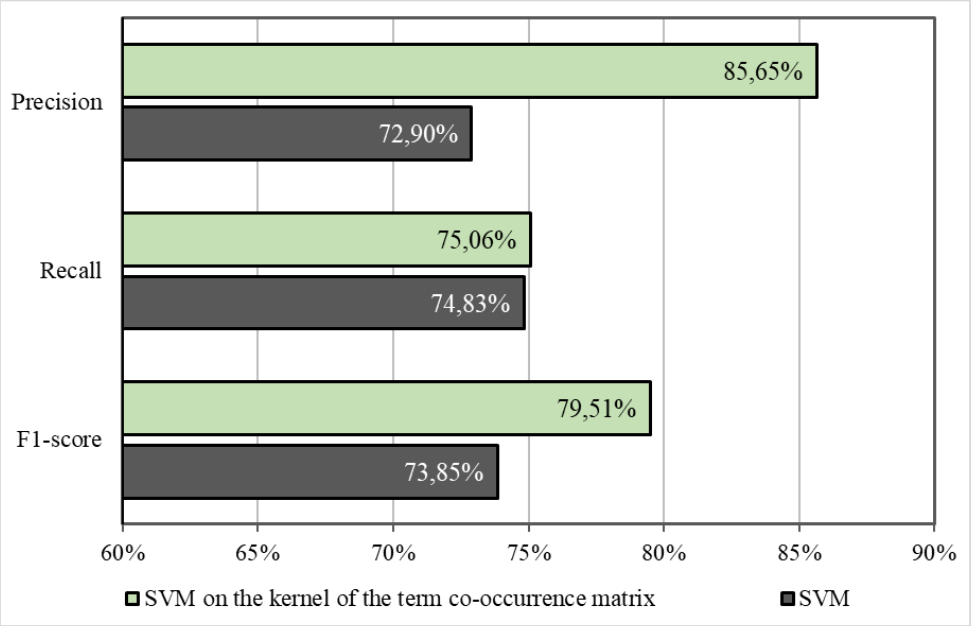

Figure 2: Calculated precision, recall and F1-score for SVM and SVM on the kernel of the term co-occurrence matrix

The assessment of classification quality is based on the detection of regularities for each class, whose attribute values are the same for most objects of the analyzed class and differ from the attribute values of other classes. The absence of such regularities indicates that this class is not a homogeneous set of profiles of text documents. The quality of the classification is considered higher, the closer the profiles of text documents are located within the class. To analyze the dispersion of classified text documents, such a qualitative gradation as condensation is introduced, which allows you to determine how close the profiles of text documents are located within the class in comparison with the location of objects within the entire original population. In order to recognize the completion of the classification procedure, it is necessary to achieve the fulfillment of the condition of compliance of the obtained division into classes with the meaningful concept of condensation. Classes will correspond to the meaningful concept of condensation in the case when the maximum spread between profiles of text documents of one class is less than the mean square spread of objects within the entire original population. This result is achieved thanks to the term co-occurrence matrix.

7. References

[1] K. K. Dukhnovska, Formation of the research dynamic vector space. Artificial Intelligence 3-4 (2015), pp. 28-36. http://nbuv.gov.ua/UJRN/II_2015_3-4_5.

[2] Z. Wang, F. Zhang, M. Ren, D. Gao, A new multifractal-based deep learning model for text mining. Information Processing & Management (2024), 61(1): 103561. doi:10.1016/j.ipm.2023.103561.

[3] Z. Wang, H. Liu, F. Liu, D. Gao, Why KDAC? A general activation function for knowledge discovery. Neurocomputing (2022), 501, pp. 343-358. doi:10.1016/j.neucom.2022.06.019.

[4] Z. Wang, B. Zhang, D. Gao, Text mining of hazard and operability analysis reports based on active learning. Processes (2021), 9(7): 1178. doi:10.3390/pr9071178.

[5] F. Simone, S. M. Ansaldi, P. Agnello, R. Patriarca, Industrial safety management in the digital era: Constructing a knowledge graph from near misses. Computers in Industry (2023), 146: 103849. doi:10.1016/j.compind.2022.103849.

[6] J. Parmar, S. S. Chouhan, V. Raychoudhury, A machine learning based framework to identify unseen classes in open-world text classification. Information Processing & Management (2023), 60(2): 103214. doi:10.1016/j.ipm.2022.103214.

[7] X. Feng, Y. Dai, X. Ji, L. Zhou, Y. Dang, Application of natural language processing in HAZOP reports. Process Safety and Environmental Protection 155 (2021), pp. 41-48. doi:10.1016/j.psep.2021.09.001.

[8] A. Deloose, G. Gysels, B. De Baets, J. Verwaeren, Combining natural language processing and multidimensional classifiers to predict and correct CMMS metadata. Computers in Industry 145 (2023), 103830. doi:10.1016/j.compind.2022.103830.

[9] R. Wang, G. Chen, X. Sui, Multi label text classification method based on co-occurrence latent semantic vector space. Procedia Computer Science 131 (2018), pp. 756-764. doi:10.1016/j.procs.2018.04.321.

[10] D. Wu, R. Yang, C. Shen, Sentiment word co-occurrence and knowledge pair feature extraction based LDA short text clustering algorithm. Journal of Intelligent Information Systems 56 (2021), pp. 1-23. doi:10.1007/s10844-020-00597-7.

[11] S. U. Khan, N. Islam, Z. Jan, K. Haseeb, S. I. A. Shah, M. Hanif, A machine learning-based approach for the segmentation and classification of malignant cells in breast cytology images using Gray Level Co-occurrence Matrix (GLCM) and Support Vector Machine (SVM). Neural Computing and Applications (2022), 34:8365–8372. doi:10.1007/s00521-021-05697-1.

[12] Y. T. Arifin, Komparasi fitur seleksi pada algoritma support vector machine untuk analisis sentimen review. Jurnal Informatika (2016), 3(2), pp. 191-199. https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/868.

[13] S. M. Avram, M. Oltean, A comparison of several AI techniques for authorship attribution on Romanian texts. Mathematics (2022), 10(23): 4589. doi:10.3390/math10234589.

[14] C. Lou, X. Xie, Multi-view intuitionistic fuzzy support vector machines with insensitive pinball loss for classification of noisy data. Neurocomputing (2023), 549(7): 126458. doi:10.1016/j.neucom.2023.126458

[15] S.-H. Na, J.-H. Lee, Memory-restricted latent semantic analysis to accumulate term-document co-occurrence events. Pattern Recognition Letters (2012), 33(12), pp. 1623-1631. https://doi.org/10.1016/j.patrec.2012.05.002.

[16] T. Ding, C. Linga, L. Mingqi, S. Hongyu, C. Gencai, Hierarchical online NMF for detecting and tracking topic hierarchies in a text stream. Pattern Recognition 76 (2018), pp. 203-214. doi:10.1016/j.patcog.2017.11.002.

[17] K. Stuart, A. Botella, Corpus linguistics, network analysis and co-occurrence matrices. International Journal of English Studies (2009), 9(3), pp. 1-20. https://revistas.um.es/ijes/article/view/99481.

[18] K. Dukhnovska, Search for regularities for classes of text documents. Scientific and practical conference “Current problems of the theory of control systems in computer sciences” (2021), Ukraine, Slovyansk, December 21-24, pp. 32-34.

[19] N. Kiktev, H. Rozorinov and M. Masoud, "Information model of traction ability analysis of underground conveyors drives," *2017 XIIIth International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH)*, Lviv, Ukraine, 2017, pp. 143-145, doi: 10.1109/MEMSTECH.2017.7937552.