Designing an information system for the electronic document management of a university: automatic classification of documents

A L Tkachenko¹ and L A Denisova²

¹Siberian State Automobile and Highway University (SibADI), 5, Pr. Mira, Omsk, 644080, Russia
²Omsk State Technical University, 11, Mira ave., Omsk, 644050, Russia
e-mail: tanaleo@mail.ru, denisova@asoiu.com

Abstract. To ensure the effective functioning of the university educational environment, document flow processes automation, which includes the task of documents automatic classification, is of great importance. The article considers the task of classifying university documents by machine learning methods in order to improve the quality of classification. Documents preprocessing was carried out, which made it possible to distinguish significant words in documents, due to which the accuracy of documents classification increased. Described are methods of extracting features from text TF and TF-IDF, which determine keywords by words frequency included in document. A modification of the TF-IDF method is proposed, which consists in calculating the words importance depending on their part of speech. This made it possible to improve the classification quality by highlighting only important and significant words in documents. Suggested is a classification algorithm using a method of support vectors to reduce the documents number involved in classification and a method of k-nearest neighbor for classification. The advantage of this algorithm over the described analogues is shown, which is expressed in the number of mistakenly classified documents decrease.

1. Introduction
The effective functioning of the educational system is largely determined by the methods and means used to automate document flow. The existing electronic document management systems mainly solve a narrow range of tasks and do not contain integration tools with other systems. The university uses several systems for document management. The university uses several systems of document management, and this makes the information searching process time-consuming for an employee. Therefore, the problem of cost reduction, connected with information storage and retrieval, aimed at university education organizing, is relevant.

The task of creating a unified information system for data storage is of great importance, as well as the task of automatically transferring documents from existing systems to a new information system. Automatic document transfer is closely related to the text classification task, for which machine learning methods are used [1].

One of the main machine learning tasks is to improve the document classification accuracy. There are methods [1-4] that have shown their effectiveness in solving the keyword classification problem. In this case, the classification takes place according to the constructed document vectors containing numerical estimates of the included words in the document. To improve the classification accuracy,
some researchers [3, 4] use methods to reduce the document vectors dimension, while others combine classification methods [5, 6].

It is noted, that the combination of the \( k \)-nearest neighbor method and the support vector machine method is effective [2, 7]. The essence of the \( k \)-nearest neighbor method is that the document is assigned to the class, to which most of the nearest documents belong. The main goal of the support vector machine is to construct a hyperplane that divides documents into classes. In the considered works, the classification is reduced to finding the number optimal value of document nearest neighbors, which is equal to the number of support vectors. This solution works well for a collection of documents in which the number of documents in the classes is the same. However, for collections with different numbers of documents in classes, the classification accuracy will be lower.

To improve the classification accuracy, we propose to introduce the word weight dependence on the part of speech when constructing document vectors. This solution will reduce the vector size, as well as assign more weight to the most significant words in the document. This solution will reduce the vector size, as well as assign more weight to the most significant words in the document. We propose to implement the university documents classification procedure in the form of an algorithm using the support vector machine method and \( k \)-nearest neighbor method. In this case, first the support vector machine reduces the documents set involved in the classification, then the \( k \)-nearest neighbor method classifies the documents. This solution will make it possible to improve the classification accuracy by reducing the number of documents involved in the classification process.

2. Theory. Solving the classifying university documents task

2.1. Document preprocessing

Document preprocessing is an important step in the document classification process because it clears document texts from words that do not have a semantic load.

Consider the university documents classifying task using the SibADI document collection, consisting of 1778 documents. Each document of the collection belongs to one of four classes: organizational documents (146 documents), long-term administrative documents (671 documents), information and reference documents (316 documents), short-term administrative documents (645 documents).

We carried out preprocessing of documents, including the following stages: all symbols of document texts are translated into lower case, texts are divided into words, noise words (words that do not have a semantic load) are removed from the texts, the remaining words of the texts are brought to normal form.

Before we applied the preprocessing tools, the university documents collection contained 37451 unique words. After preprocessing, the number of unique words in the collection was reduced to 13460 (table 1).

| Document preprocessing method | Number of unique words in SibADI document collection |
|-------------------------------|-----------------------------------------------------|
| Before preprocessing           | 37451                                               |
| After preprocessing            | 13460                                               |

As a result of preprocessing, all significant words found in the document are distinguished as document characteristics. The number of unique words has been reduced by 3 times, which will reduce the number of classification errors.

2.2. Creating a Numeric Document Model

Creating a numerical model is a necessary step that allows us to represent the document text in the form of a word vector. The vector describes the features and patterns of the text that are important for the university documents classification.

The results of research showed [9] it is better to use the BoW model (bag of words) to classify university documents. This model allows you to represent a document as a vector of words included.
Let's build a BoW model in the form of a W matrix. Matrix rows represent documents in the vectors form of words included in a document. Matrix columns describe the words included in documents:

\[
W = [w_{ik}] = \begin{bmatrix}
    w_{11} & w_{12} & \ldots & w_{1s} \\
    w_{21} & w_{22} & \ldots & w_{2s} \\
    \vdots & \vdots & \ddots & \vdots \\
    w_{ni} & w_{n2} & \ldots & w_{ns}
\end{bmatrix},
\]

where \( w_{ki} \) is weight of the \( k \)-th word in the \( i \)-th document, \( i = 1, n \), \( n \) is number of documents, \( k = 1, s \), \( s \) is number of unique words in the entire document collection.

Let's consider methods for extracting characteristics from text that are used to calculate documents word weights.

TF (term frequency) method evaluates the word importance within a single document [10]. According to the TF method, the word weights are the word frequency within the document, \( TF = l_{ki} / l_i \), where \( l_{ki} \) is number of the \( k \)-th word occurrences in the \( i \)-th document, \( l_i \) is total number of words in the \( i \)-th document.

Another method, TF-IDF (term frequency – inverse document frequency), uses the document reverse frequency to reduce the weights of commonly used words [10]. According to the TF-IDF method, words most common in a particular document and less common in the rest receive more weight \( TF-IDF = TF \cdot IDF \), where \( IDF = \log \left( \frac{D}{D_k} \right) \) expresses reverse frequency of the document, \( |D| \) is number of documents in the dataset; \( D_k \) is number of documents in which the \( k \)-th word occurs.

The methods discussed are good at distinguishing keywords in documents. However, these methods make the most common and rarest words in documents more significant, which may not reflect the essence of the document. Because of this, the classification quality is reduced. To solve this problem, we propose to additionally check part of speech of a word when calculating weight.

We upgrade the TF-IDF method by introducing a dependence of the word frequency on the part of speech. In the proposed method TF-IDF-PS (part of speech), the weight of a word is calculated as follows:

\[
W_{ki} = \begin{cases}
    \frac{l_{ki}}{l_i^n + l_i^v + l_i^a} \log \left( \frac{|D|}{D_k} \right), & \text{if } t_k \in P; \\
    0 & \text{in other cases},
\end{cases}
\]  

(1)

where \( l_i^n \) is number of all nouns in the \( i \)-th document; \( l_i^v \) is number of all verbs in the \( i \)-th document; \( l_i^a \) is number of all adjectives in the \( i \)-th document; \( P = \{P^n, P^v, P^a\} \) is set of words, \( P^n \) is set of nouns, \( P^v \) is set of verbs, \( P^a \) is set of adjectives.

We remove from the matrix \( W \) words with a zero weight value in all documents to reduce the dimension of document vectors.

Table 2 shows the number of words in the BoW model after reducing the dimension of the document vectors. Document vectors contained a weight of 13460 words before dimensioning. The dimension of the vectors did not decrease after the use of the TF and TF-IDF methods. Using the TF-IDF-PS method, it was possible to reduce document vectors to 9116 word weights.

Using the proposed TF-IDF-PS method, noise words will receive the least weight in the document and will not affect the classification result. Thus, the number of noise words missed in the document preprocessing step is also reduced in the document numerical model construction step. This will improve the classification quality by ensuring that the classifier will focus only on important and significant words in documents.
Table 2. Number of words in the BoW model after the document vectors are dimensioned.

| Method      | Number of words |
|-------------|-----------------|
| TF          | 13460           |
| TF-IDF      | 13460           |
| TF-IDF-PS   | 9116            |

2.3. Building a classifier.

Building a classifier is an important part of the document classification process. Because the result of the classification depends on the selected machine learning method.

We will build a classifier based on the SVM (Support Vector Machine) method and the kNN (k-nearest neighbor) method.

Let's consider the SVM method. The essence of the method is to build a hyperplane that separates documents into classes. Let's look at the two-dimensional case of classification when we consider two classes of documents (Figure 1) to illustrate how method does work.

Figure 1. The result of dividing university documents into classes by the support vector method: $w_1$ is weight of the word «provision»; $w_2$ is weight of the word «order».

We took five long-term administrative documents and five organizational documents, which contain the words «provision» and «order» to build a hyperplane. The frequency with which words are found in documents is taken as a weight of word. The figure shows the separating plane, indicated by a solid line, it is drawn so that all documents of one class lie on one side of it, and of the other class lie on the other. Then, to classify unknown documents, it is enough to see on which side of the line they turn out to be.

In general, we can draw an infinite set of planes that separate documents into classes. However, it is better to select the plane as far away from the documents as possible. In the SVM method the distance between the plane and the document nearest to it is maximized. When this condition is met, the plane is called the optimal plane, and the nearest points are called support vectors. In the figure, dashed lines pass through them.

This example explains the case where documents are linearly separated. However, in practice, the data structure is often unknown, and it is impossible to determine whether documents are linearly separable or not. In this case, it is necessary to move from the original document characteristic space to the new one, in which the documents are linearly separable. To do this, we use kernel functions. Let's apply the radial basis function to the example \[ K(d_q, d_z) = \exp(-\gamma \parallel d_q - d_z \parallel^2) \], where $d_q$ and $d_z$ are
document vectors, \( d_z \in D^{\text{train}} \), \( D^{\text{train}} \) is training dataset, \( d_q \in D^{w} \), \( D^{w} \) is set of support vectors; \( \gamma \) is kernel parameter. The kernel calculates the distance between document \( d_z \) and support document \( d_q \). Kernel value is greater when documents are closer. The result of separating documents into classes is shown in Figure 2.

**Figure 2.** The result of dividing documents into classes by the support vector method using the kernel function; \( w_1 \) is weight of the word «provision»; \( w_2 \) is weight of the word «order».

The SVM method using the radial basis function divided the space into areas. The area in the form of a figure corresponds to long is term administrative documents, the rest to organizational documents. In this case, the document class depends on the area in which the document came.

The use of SVM gives high classification rates on data with a large feature space. In addition, SVM accurately classifies documents when classes are split. However, SVM classification accuracy is reduced when classes overlap. In this regard, we propose to supplement the classification with the \( k \)NN method that allows you to accurately classify documents when document classes overlap.

Let's consider the \( k \)NN method. The essence of the method is that the document is assigned to the class to which most of the nearest documents belong. The nearest documents are defined using the distance metric (Figure 3).

**Figure 3.** Document classification by the \( k \)-nearest neighbor method (with \( k = 3 \))
We took five long-term administrative documents and five organizational documents, which contain the words «provision» and «order» to show how the method \( kNN \) classifies documents. The frequency of the words found in documents is taken as a weight of word. The distance to the nearest documents is shown by arrows. The classifier calculates the distance from the document to be classified to the training dataset documents to find the document corresponding class. Next, \( k \) nearest document neighbors are selected from the training dataset. The number of classifying documents is determined in advance and set by the parameter \( k \). The figure shows the case of classification at \( k = 3 \). The \( kNN \) method gives high classification rates for both separable and overlapping classes. However, the method is unstable to data outlier and classifies non-preprocessed documents worse.

We upgrade the \( SVM \) and \( kNN \) methods by combining them into a \( SVM-kNN \) classification algorithm (Figure 4). Let’s consider the stages of document classification with the proposed algorithm.

![Figure 4. SVM-kNN algorithm for document classification](image)

The document classification is divided into two steps. At the first step, the \( SVM \) method narrows the documents space involved in classifying new objects to the reference vectors, which are closer to the separation plane. To do this, we reduce the task of document classifying to a multiple problem of separating into 2 classes using the «one against all» strategy. The «one against all» strategy consists of constructing one \( SVM \) per class, which is trained to distinguish the samples of one class from the samples of all remaining classes. In this case, prediction of the document class occurs according to a classifier with a maximum value. We use a radial basis function to compute support vectors. In the second step, we classify documents by support vectors using the \( kNN \) method. We use the Euclidean distance metric to determine the nearest neighbors of the document \([2]\)  

\[ \rho(d_q, d_i) = \sqrt{\sum_{i=1}^{q} (d_i - d_i^*)^2} , \]

where \( q \) is number of support vectors.

The synthesis of \( SVM \) and \( kNN \) methods allows us to reduce the number of documents participating in the classification at the training stage of the classifier. Data outlier are removed and the classification occurs according to the most informative documents. In addition, \( SVM-kNN \) algorithm allows us to determine the document class in the case of separable and overlapping classes.
2.4. Classifier quality assessment

The final stage of the documents classification is the classifier quality assessment. To qualify objectively, it is necessary to correctly select the volume ratio of the training and test data samples. Using the training dataset, a classifier is created – an algorithm that correlates documents with classes. The test dataset is used to check the work of the constructed classifier. Studies showed [1, 8] that the classifier objective assessment is given by the k-fold cross-validation method. The essence of the method is as follows. The training dataset is split into k smaller sets. Then the model is tested k times, each time k-1 parts are involved in the training, and the part that did not participate in the training is used for classifier work testing. As a result of k-fold cross-validation of the classifier operation, k estimates are obtained, from which one estimate is derived by averaging. To calculate the classification scores, the inconsistency matrix is used (Table 3).

The main classifier quality measures are metrics of precision \( J_p = \frac{G_p^+}{(G_p^+ + G_p^-)} \) and recall \( J_r = \frac{G_p^+}{(G_p^+ + G_n^-)} \), where \( G_p^+ \) is a sign that the classifier correctly defined the document class; \( G_p^- \) is a sign that the classifier incorrectly defined the document class; \( G_n^- \) is a sign that the classifier misidentified that the document is not in a class \( C_j \) [7].

It is noted [7] that in practice the simultaneous achievement of maximum \( J_p \) and \( J_r \) values is impossible. Therefore, the harmonic mean between \( J_p \) and \( J_r \) is calculated \( J_F = \frac{2J_p J_r}{(J_p + J_r)} \).

Based on the obtained values of the metrics, by which the work of the classifier is assessed, a conclusion about the acceptability of using the classification method is made.

3. Results of experimental studies

Two series of experiments were carried out using SibADI documents to assess the quality of the classification using the proposed method of extracting features from the text TF-IDF-PS and the classification algorithm SVM-kNN. The first series shows the evaluation of the classification quality using the TF-IDF-PS method. For this, the collection of documents was divided into years. One training and four test collections are formed. The 2016 document collection (271 documents) was adopted as a training dataset. Testing was conducted at the collections of 2017 (274 documents), 2018 (268 documents), 2019 (272 documents) and 2020 (224 documents).

The classification results were compared using the proposed TF-IDF-PS method with the classification results using the existing TF and TF-IDF methods. The result is compared by the metric \( J_F \) (F-measure, %), the algorithm SVM-kNN is taken as a classifier. The results are shown in Figure 5.

![Figure 5. The result of comparing methods on different years document collections](image_url)
As you can see, the value obtained using the TF and TF-IDF methods deteriorates significantly over time. This result is explained by the fact that the methods are turned to noise data missed in the preprocessing step. Methods also assign weights to non-informative words, such as pronouns and adverbs. The proposed TF-IDF-PS method showed the best result. Due to the introduced dependence of the word weight on the part of speech, non-informative words missed in the pre-processing step are not taken into account in calculating weights. This allows us to improve the quality of classification due to the fact that the classifier focuses only on important and significant words in documents.

It is shown that over four years the quality of document classification according to the metric gradually decreases at the collections of 2017, 2018, 2019 and 2020. The worst classification result was obtained at the 2020 collection using the TF and TF-IDF methods. The \( J_F \) value compared to the result obtained at the 2017 collection decreased by 22%. The decrease in the quality of the classification is due to the fact that the style of writing the text depends on the person who wrote it. Different people may write the same documentation in different years. Accordingly, the set of keywords that the classifier focuses on also changes.

The best classification result on the document datasets of each year is obtained using the TF-IDF-PS method. The \( J_F \) value on the 2020 dataset compared to the result obtained on the 2017 dataset decreased by 10%. TF-IDF-PS method identifies only informative words such as nouns, verbs, and adjectives as classifying features.

The second series of experiments shows the comparison of the classification results with the proposed SVM-kNN algorithm with the classification results using the kNN and SVM methods. The method for extracting characteristics from text TF-IDF-PS is used. The number of nearest neighbors in the methods kNN and SVM-kNN is taken to be three. As a kernel function in the SVM method, the radial basis function is used. We classified the entire dataset (1778 documents). \( J_F \) (F-measure, \%) is used as metric for evaluation. The result of the document classification for each class is shown in Figure 6.

![Figure 6. Result of documents classification by each class](image)

It is shown that the largest difference in \( J_F \) value is received by organizational documents (class \( C_1 \)). The worst result is obtained by the \( kNN \) method, \( J_F \) by class is 86%, which is 2% less than the result obtained by the SVM method, and 9% less than the result obtained by the SVM-kNN algorithm. This is due to the fact that organizational documents are drawn up on different topics, and it is more difficult to find keywords in them.

The best result for all document classes is obtained using the SVM-kNN algorithm. According to long-term administrative documents (class \( C_3 \)), the increase in comparison with the results of classification by the SVM method was 3%, by the \( kNN \) method 5%. The difference between the results obtained by the \( kNN \) method and the SVM-kNN algorithm in \( C_3 \) class was 6%. The difference between the SVM classification results and the SVM-kNN algorithm is 4%. By \( C_4 \) class (short-term administrative
documents), the difference between the results obtained by the methods \( kNN \), \( SVM \) and the algorithm \( SVM-kNN \) is 5% and 3%. Table 3 shows the overall result of document classification.

**Table 3. Result of documents classification by \( kNN \), \( SVM \) and \( SVM-kNN \).**

| Document classification method | \( J_p \), % | \( J_r \), % | \( J_F \), % | Number of incorrectly classified documents |
|-------------------------------|------------|------------|-------------|------------------------------------------|
| \( kNN \)                      | 93.55     | 92.95     | 92.94       | 34                                       |
| \( SVM \)                      | 95.41     | 94.92     | 94.95       | 25                                       |
| \( SVM-kNN \)                  | **98.40** | **98.31** | **98.31**   | **9**                                    |

As we can see, the results of the method \( SVM-kNN \) higher than the results of other methods. Compared to the results obtained by the \( kNN \) and \( SVM \) methods, the number of documents incorrectly classified by the \( SVM-kNN \) method decreased by more than three times.

4. The discussion of the results

Automatic document classification is one of the main tasks of the university electronic document management development. The use of automatic classification increases the convenience of working with the electronic document management system, as it automates the classifying documents routine process so that university employees can focus on basic tasks and decision making.

Based on the study, it was concluded that the proposed method of extracting features from the text of \( TF-IDF-PS \) and the classification algorithm \( SVM-kNN \) are suitable for solving the problem of automatic classification of university documents. Numerical experiments have shown that due to the introduced dependence of word weight on part of speech, \( TF-IDF-PS \) allows assigning more weight to the most significant words in the document, thereby increasing the quality of classification.

Usage of the \( SVM-kNN \) algorithm significantly improves the classification result. By reducing the number of documents involved in the classification process, the accuracy of the classification is improved and the number of erroneously classified documents is reduced.

It seems promising to analyze the suitability of the \( TF-IDF-PS \) method for real-time classifier training.

5. Conclusion

As a result of the issue, document preprocessing was carried out, which made it possible to distinguish all significant words found in documents. This led to a decrease in classification errors, which is important for an employee of the university, since he will spend less time on manual document classification and will be able to devote working time to solving more creative tasks.

Classification was carried out by hybrid method, which includes reduction of multiple documents involved in classification by method of support vectors and document classification by method of \( k \)-nearest neighbors. This solution allows us to distinguish among documents the most informative of them and accurately classify objects in the case of overlapping classes.

It is proposed to calculate features importance depending on part of speech. This reduces the number of noise words missed in the preprocessing step and improves the classification quality due to the fact that the classifier focuses only on important and significant words in documents.

References

[1] Thangaraj M and Sivakami M 2018 Text Classification Techniques: A Literature Review *Interdisciplinary Journal of Information, Knowledge, and Management Volume* vol 13 pp 117–135

[2] Wan Ch H, Lee L H, Rajprasad R and Isa D 2012 A Hybrid text classification approach with low dependency on parameter by integrating K-nearest neighbour and support vector machine *Expert Systems with Applications, elsevier journal* vol 39(15) pp 11880–11888
[3] Su Y, Huang Y and Kuo J C-C. 2018 Efficient Text Classification Using Tree-structured Multilinear Principal Component Analysis *International Conference on Pattern Recognition* pp 585–590

[4] Chen Q, Yao L and Yang J 2016 Short text classification based on LDA topic model *International Conference on Audio, Language and Image Processing* pp 749–75

[5] Goyal Sh 2017 Review Paper on Sentiment Analysis of Twitter Data Using Text Mining and Hybrid Classification Approach *International Journal of Engineering Development and Research* vol 5 issue 2 pp 197–199

[6] Prema A and Gopalakrishnan S 2021 A Novel Hybrid Algorithm For Text Classification To Implement Sentiment Analysis *Design Engineering* pp 8739–8745

[7] Sivakumar M, Karthika C and Renuga P 2014 A Hybrid Text Classification Approach Using KNN And SVM *International Journal of Innovative Research in Science, Engineering and Technology* vol 3 issue 3 pp 1987–1991

[8] Batura T 2017 Automatic text classification methods *Programnye produkty i sistemy [Software & Systems]* vol 30 no 1 pp 85–99

[9] Tkachenko A 2021 Solving the problem of university documents classification based on intellectual analysis methods *Proceedings in Cybernetics* vol 1 pp 12–19

[10] Manal M and Nazlia O 2020 Question classification based on Bloom’s taxonomy cognitive domain using modified TF-IDF and word2vec *PloS One* vol 15(3) pp 1–21

[11] Le Thi Minh Nguten 2019 Text classification based on support vector machine *Dalat university journal of science* vol 9 issue 2 pp 3–19