Multi-Class Text Classification of Uzbek News Articles using Machine Learning

Multi-Class Text Classification of Uzbek News Articles using Machine Learning

I M Rabbimov, S S Kobilov
Samarkand State University, Samarkand, Uzbekistan
E-mail: ilyos.rabbimov91@gamil.com, kobsam@yandex.ru

Abstract. A large amount of online news on various topics is being posted on the Internet. One of the tasks of processing this data is to provide the user with appropriate methods and tools for quick and easy search for important and interesting news. An approach to solve this problem is the reasonable distribution of news into respective classes. This increases the importance of automated classification of an electronic document section. In this paper, we consider the task of multi-class text classification for the texts written in Uzbek. The articles on ten categories were selected from the Uzbek "Daryo" online news edition and a dataset was developed for them. When performing multi-class text classification for this dataset, the following 6 different machine learning algorithms were used: Support Vector Machines (SVM), Decision Tree Classifier (DTC), Random Forest (RF), Logistic Regression (LR) and Multinomial Naïve Bayes (MNB). A detailed technological description of the stages of the proposed functional scheme of text classification and developed software is given. The TF-IDF algorithm and word-level and character-level n-gram models were used as the feature extraction methods. When defining hyperparameters for text classification, 5-fold cross-validation was used. Experiments were conducted and the highest accuracy was 86.88%. The models and methods that are proposed in this paper can be successfully used in the classification of texts written in the Uzbek language and further research in this area.

1. Introduction

Text classification methods are mainly used to determine the belonging of news texts, Internet articles and stories to one or another of the previously determined categories. Therefore, the purpose of classification is to clarify whether a new text belongs to this or that category. Continuous development of information technology directly affects an increase in the amount of information (data). One of the effective approaches to extract important and interesting data from this amount of data is their division into classes (categories). The rapid growth in data volume increases the complexity of classification; it also requires a lot of time and labor for manual classification; these and other factors show the need and importance of automated classification of electronic documents. Therefore, in this paper, we consider the tasks of the automated classification of online Uzbek news articles.

In the last decade, for the languages of great linguistic resources, the number of scientific studies of text classification problems have greatly increased. Our analysis shows that in the field of Uzbek text classification we are far behind. However, the number of Internet texts published in Uzbek is constantly increasing. For example, the list of news and public information departments of the national search system of the Republic of Uzbekistan www.uz includes about 200 sites,

each of them daily publishes online an average of 20 articles. People of various specialties are interested in this news. Finding important and interesting news takes a lot of time and this fact further strengthens the need to solve the problems of news articles classification (categorization).

News is usually classified according to the topic categories or belonging to a particular geographical area, the articles are classified according to the typical areas which they are associated. Automated typing of texts by groups (Art, Economics, Politics, Sports, etc.) is important for both business classes and individuals [1].

A large number of scientific studies on English text classification have been performed. There are studies in this area for other languages using the n-gram model and TF-IDF algorithms. For example, in [2, 3, 4] the problems of text classification in Turkic language are considered. The same problems are solved in studies and publications [5, 6, 7] for Arabic language, the Chinese news text classification in [8, 9], text classification for Indonesian news article in [10, 11], sentiment classification of the Slovenian news texts in [12], text classification of Uyghur and Kazakh languages in [13], category classification and topic discovery tasks for Japanese language in [14], automatic classification of Russian scientific texts in [15], similar problems are studied for Sindhi [16], Vietnamese [17]. We have already noted that, unlike other languages, the tasks of text classification in Uzbek have not been sufficiently studied. One of the first studies in the Uzbek language in the field of sentiment analysis was conducted by E. Kuriyozov, S. Matlatipov [18]. A scarce number of works solved subtasks on text classification, and the lack of resources for automated processing of Uzbek language texts strengthen the novelty and relevance of our work. As for the family and group of Uzbek language, it is a part of the Turkic group of the Altai language family. It is the official language of the Republic of Uzbekistan, about 34 million of the republic's population speaks this language, and in several countries, it is understood and used for communication. In this paper, for the automated classification of Uzbek news articles, we use the word-level n-gram and character level n-gram models with the TF-IDF algorithm. When performing the multi-class text classification, a group of 6 machine learning algorithms is used: Support Vector Machines, Decision Tree Classifier, Random Forest, Logistic Regression and Multinomial Naïve Bayes. The results are compared between various n-gram models and machine learning algorithms.

Description of the text classification task mathematically. Let a set of documents $D = \{d_1, d_2, \ldots, d_n\}$ are given, which consists of n - documents and a predetermined set $C = \{c_1, c_2, \ldots, c_m\}$, consisting of m - categories. The mapping $F : D \times C \to true, false$ is called a classifier. Here $d_i \in D$, $c_j \in C$, if $F(d_i, c_j) = true$, then the document d_i belongs to the category c_j, otherwise the document d_i does not belong to the category c_j. Classification tasks are also divided into different directions, such as binary, multi-class, multi-label problems [17]. In our work, we discuss the task of multi-class text classification for the texts written in Uzbek.

2. Related Work

In this section, the work associated with the task of online news articles classification is analyzed. Therefore, the studies corresponding to our task and relating to different natural languages are analyzed.

In Turkey, when classifying news and informational documents [3], an efficient and fast TF-IDF algorithm and SVM classifier algorithm with linear kernel function was used to create a weight matrix. An experiment was conducted in two Turkish text datasets to determine the categories of news, and detection of the columnists as a result, 99%, and 98% accuracy were obtained respectively. In [2], the problems of categorizing texts in Turkish were solved based on the choice of an n-gram model, punctuation characters and the use of stemming. Here, the basic classification algorithms were Naive Bayes, Support Vector Machines and Random Forest. As a result, the problem of forecasting the writing style of articles, the article's author and the gender of an employee were studied from Turkish newspaper articles.

The text classification problem for two morphologically rich and mutually close agglutinative languages (Kazakh, Uyghur) is considered in detail in [13]. In this work, text classification tasks have been completed for Uyghur texts in nine categories: finance, law, culture, tourism, sports, education, science, health and entertainment, and for Kazakh texts in eight categories: finance, sports, law, tourism, culture, science, education, and entertainment. Based on the results of experiments, it was stated that the morpheme-based approach showed better results than the word-based approach. The machine learning and TF-IDF algorithms and the word vector technology were used to conduct these studies.

The task of automatic tweet classification based on news category in Indonesian language was studied in [10]. 11 categories were identified from various sources. For their classification, the algorithms ZeroR, Naïve Bayes Multinomial (NBM), Support Vector Machine, Random Forest and Sequential Minimal Optimization were used. The best result of 77.47% accuracy was recorded by the NBM classifier. A web-based application was developed based on the results of the study, which makes it possible to determine which category given tweeting belongs to. In [11] the Support Vector Machine and TF-IDF algorithms are used to classification digital news in Indonesian. Based on the results of the experiments, the allowed values of the gamma and C parameters of the Support Vector Machine algorithm were determined and 85% classification accuracy was obtained.

The multi-label text classification problem for Arabic news articles was discussed in [5]. To accomplish this task, the following classifiers were used: Decision Tree, Random Forest and K-Nearest Neighbors with k = 5. In another study [6], to solve the problem of multi-label text classification on Arabic texts, the supervised approach was used, an experiment was performed on the selected dataset with the result of 71% accuracy. To simplify the task of multi-label text classification, several methods were proposed. One of these methods was based on the transformation of the multi-labeled data method to the corresponding single label data method. This is the method used in the above works.

3. Methodology

In this section, we describe the steps involved in executing the multi-class text classification for online news articles. In particular, such stages as dataset collection, text preprocessing, feature selection, implementation of text classification algorithms and selection of hyperparameters for text classifiers are discussed.

3.1. Data Collection

For the data collection of online news articles, the "Daryo" online edition was selected. Based on the statistics of the national search system, in the number of visitors and views, the "Daryo" (`https://daryo.uz`) takes a leading role in Uzbekistan. Another feature of the "Daryo" online news site is that news articles are categorized by site editors. This is one of the best features to accomplish the text classification task on website articles.

We designed a special program to collect data from online news articles During the implementation of this program, Requests Python library and Beautiful Soup Python were used.

The "Daryo" online news website has over 50 categories. We have selected 10 of them: world, science, culture, local, show–business, sport, technologies, auto, photo, and movie. When choosing the categories, the largest number of articles and the number of characters in them were taken into account. To maintain the balance in articles number in each category, a maximum of 2000 articles were selected from each category. These news articles were online published in the time interval from 01/01/2016 to 12/31/2019. Dataset included data of the form: news article, link, text, title, date of publication, category, text length for each article. The texts of

the articles were cleared of HTML tags, URLs, addresses and emojis. Figure 1 shows a dataset formation diagram based on the number of news articles in each category.

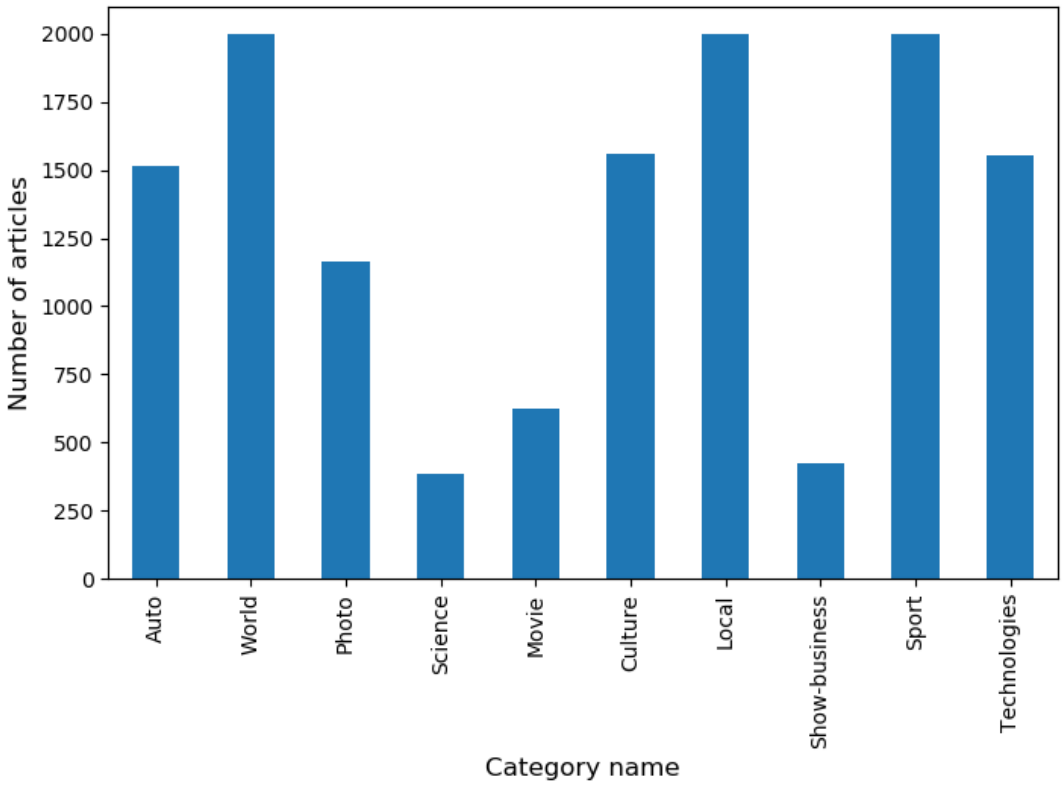

Figure 1. Number of articles in each category.

Table 1 displays dataset statistics for each class of news articles. Table 1 shows the average number of words - 232, and the average number of characters - 1956 in each article. The dataset is prepared by the developers can be issued to those interested in the permission of the special department of the daryo.uz website.

3.2. Text Pre-processing

For the implementation of the text classification task, a process of data pre-processing is performed for the collected dataset. Such a process would help us, firstly, to reduce the volume of material (important) data by removal insignificant (unimportant) data. Secondly, it would increase the accuracy of text classifiers. The pre-processing of news article texts has three stages. At the first stage, all punctuation symbols and numbers are removed from the news articles' texts. Here the use of letters and symbols o‘, g‘, ’ in Uzbek based on Latin graphics are also considered. The second stage is connected to the conversion of all letters to lowercase ones. The third stage removes the stop words from the texts.

The stop words in the tasks of Natural Language Processing (NLP) and Information Retrieval display important and useful features of the text. The main task of stop words is to correctly

Table 1. Some statistic information about the collected dataset.

Class name (Uzbek)	Class name (English)	Number of articles	Average number of words	Average number of characters
Dunyo	World	2000	279	2330
Ilm-fan	Science	384	241	2009
Madaniyat	Culture	1559	154	1270
Mahalliy	Local	2000	670	5739
Show-business	Show-business	424	138	1123
Sport	Sport	2000	276	2348
Texnologiyalar	Technologies	1551	155	1318
Avto	Auto	1516	128	1096
Foto	Photo	1165	152	1227
Kino	Movie	625	132	1096
Barchasi	Total	13224	232	1956

formulate the sentences and link phrases. There is no universal stop words list that NLP tools use. Therefore, in various tools, depending on the purpose, the appropriate stop words lists are used. There is no list of stop words for the Uzbek language. We have compiled such a list. Two methods were used to complete this work. In the first approach, a frequency dictionary [19] and a TF-IDF weight dictionary were developed on the basis of all collected texts and based on most frequency words, the stop words were highlighted. The second approach was based on the translation of existing English stop words. The results of these two approaches were summarized in a list of stop words. Thus, the list of stop words for the Uzbek language was first reported. The list has 373 words. The stop words in Uzbek can be downloaded at `https://github.com/ilyosrabbimov/uzbek-stop-words` in JSON and TXT formats. The list of Uzbek stop words can be expanded, if necessary, in accordance with the tasks to be solved. Table 2 shows the samples from the Uzbek stop words list.

Table 2. Samples of the stop words in the texts written in Uzbek.

Stop words					
men	sen	u	biz	siz	ular
bilan	uchun	ham	bu	ushbu	bo‘lib
...	...	...	...	...	...
hamda	va	lekin	ammo	biroq	yoki
ekan	lozim	nima	qanday	necha	haqida

3.3. Feature Selection

At this stage, each text of news article is converted into a feature vector and using the prepared dataset, a vector of new features is obtained. For sampling the appropriate features from the dataset, we used the TF-IDF vectorization algorithm along with the n-gram model. When obtaining feature vectors, the unigram, bigram, trigram, and four-gram were used for the word

level and character level of the n-gram model with the TF-IDF vectorization algorithm. By using separately the feature vectors, we have performed the task of text classification.

N-gram modeling is considered as a widely used method for processing and modeling natural languages, for defining and analyzing their main features. n-gram is a mutually adjacent sequence of elements of n length. These elements can be a sequence of words, bytes or characters of the n-gram model often used in text classification - these are the word-based or character-based models. As noted before, using these two models and the TF-IDF vectorization algorithm, we have obtained feature vectors, and extracted the appropriate vectors for the problem under consideration.

The weighted TF-IDF value of a word in a set of words or in a document text is a statistical size aimed to display the degree of word importance in a document. TF-IDF weight size is widely used in Text Mining and NLP. Its value is usually calculated as follows. Suppose we have a set of N documents, f_{ij} is the frequency of word i in document j. If n is the number of all words in the document j, then the word frequency is calculated by formula $TF_{ij} = f_{ij}/n$. If the word i in the N set of documents appears in n_i document, then for the word i the IDF (inverse document frequency) is determined by formula $IDF_i = log_{10}(N/n_i)$. The formula $TF-IDF_{ij} = TF_{ij} \times IDF_i$ is also used for the word i in document j. In each row of the dataset, the corresponding feature vector is formulated for six cases and experiments are conducted for each case. These cases and experimental results are detailed in the results section.

3.4. Model

When performing multi-class text classification of online news articles, we applied 6 different machine learning algorithms: Support Vector Machines, Decision Tree Classifier, Random Forest, Logistic Regression, Multinomial Naïve Bayes. For SVM, multiple kernel functions were used, in particular, linear Support Vector Classification (linear SVC) and Support Vector Machine with the Radial Basis Function (RBF SVM). To implement these classifiers, the Scikit-learn machine learning library [20] was used, executed in the Python language environment and programming system. The model we have developed to solving this problem is presented in the following functional diagram (Figure 2).

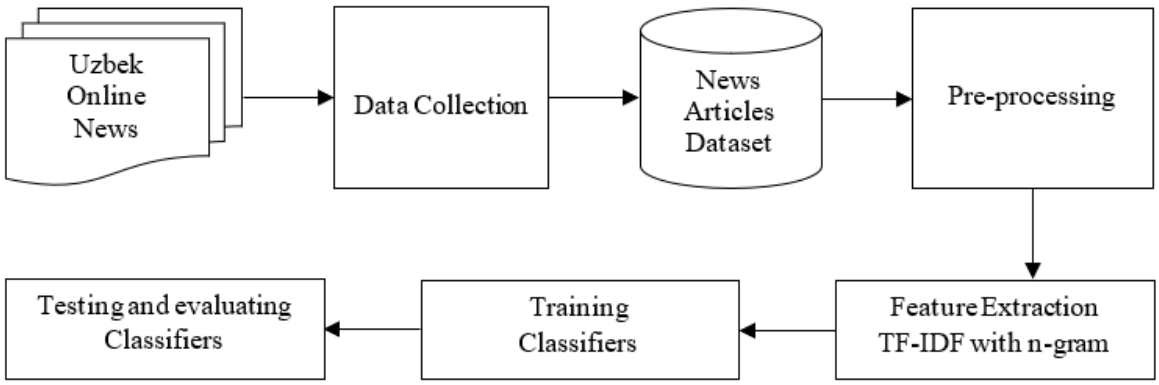

Figure 2. General structure of the proposed model for multi-class text classification of Uzbek news articles.

A detailed algorithmic description of the proposed model, a functional diagram which is given in Figure 1.

Data collection:
Collecting news articles and information about it and clean up the text of articles (HTML tags, URL address)
Save dataset
Load data:
$D \leftarrow \{'news\ article\ 1', \ldots, 'news\ article\ N'\}$
$L \leftarrow \{'label\ 1', \ldots, 'label\ N'\},\ here\ |D| = |L| = N$
$S \leftarrow \{'stop\ word\ 1', \ldots, 'stop\ word\ M'\},\ here\ |S| = M$
Text pre-processing:
for each $d_i \in D$ *do*
  Remove all punctuation symbols ($'.'$, $'!'$, $';'$, $\ldots$, $'?'$) *and numbers* ($[0 \ldots 9]$) *from* d_i
  Conversion of all letters to lowercase from d_i
  for each $s_j \in S$ *do*
    Remove stop word s_j *from* d_i
  end for
end for
Splitting data to train and test:
$TrainD \leftarrow Split[D, size = 0.8],\ TrainL \leftarrow [L, size = 0.8]$
$TestD \leftarrow Split[D, size = 0.2],\ TestL \leftarrow [L, size = 0.2]$
TF-IDF feature extraction:
for each ngram $\in \{unigram,\ bigram,\ trigram\}$ *do*
  $TrainF \leftarrow\ TfidfVectorizer(analyzer =' word',\ ngram,\ TrainD)$
  $TestF \leftarrow\ TfidfVectorizer(analyzer =' word',\ ngram,\ TestD)$
  Training and evaluate classifiers:
  for $c \in \{Linear\ SVC,\ RBF\ SVM,\ DTC,\ RF,\ LR,\ MNB\}$ *do*
    Initialize hyperparameters of c
    Training c with TrainF and TrainL
    Test c with TestF and TestL
    Calculate scores
    Compute confusion matrix
  end for
end for
for each ngram $\in \{bigram,\ trigram,\ fourgram\}$ *do*
  $TrainF \leftarrow\ TfidfVectorizer(analyzer =' char',\ ngram,\ TrainD)$
  $TestF \leftarrow\ TfidfVectorizer(analyzer =' char',\ ngram,\ TestD)$
  Training and evaluate classifiers:
  for $c \in \{Linear\ SVC,\ RBF\ SVM,\ DTC,\ RF,\ LR,\ MNB\}$ *do*
    Initialize hyperparameters of c
    Training c with TrainF and TrainL
    Test c with TestF and TestL
    Calculate scores
    Compute confusion matrix
  end for
end for

The Scikit-Learn specialized machine learning software library for Python includes many classification methods that provide convenient tools for use. We describe the main stages and mechanisms for using the models of this library. Such a description reveals in more detail the steps we performed and the steps involved in using modules, classes, and related parameters.

To implement Linear SVC, sklearn.svm.SVC was used, where SVC is the class, svm is the sklearn library model (Scikit-Learn). For this classifier, the kernel and c parameters were selected as 'linear' and 1, respectively. For the RBF SVM classifier was also executed by sklearn.svm.SVC, where the parameters are kernel = 'RBF', c = 1 and gamma = 1. When implementing the Decision Tree Classifier, the sklearn.tree.DecisionTreeClassifier class was used. The parameters for this criterion and max_depth class are 'gini' and 20. To perform Logistic Regression, the sklearn.linear_model.LogisticRegression class was used. As parameters, multi_class and solver were initialized respectively 'multinominal' and 'newton-cg'. Random Forest Classifier was implemented based on the sklearn.ensemble.RandomForestClassifier class. In this case, the parameter n_estimators = 1000 and max_features = 'auto'. For the Multinomial Naïve Bayes classifier, sklearn.naive_bayes.MultinomialNB was used, and as the parameter alpha = 0.01.

The hyperparameters of these classifiers were determined based on the use of the Grid search algorithm. It is on the basis of these hyperparameters that the classifiers gave the highest accuracy. Hyperparameters were determined using 5-fold cross-validation. The name and values of the hyperparameters of the of each classifier that correspond to the upper accuracy value are given in Table 3.

Table 3. Hyperparameters tuning.

Classifier	*Feature selection method*	*Parameters for tuning*	*Best value*
Linear SVC	Character based fourgram with TF-IDF	C=[0.01, 0.1, 1, 10, 100, 1000]	C=1
RBF SVM	Character based fourgram with TF-IDF	gamma=[10, 1, 0.1, 1e-2, 1e-3], C=[0.01, 0.1, 1, 10, 100, 1000]	gamma=1,C=1
DTC	Character based fourgram with TF-IDF	criterion=[gini, entropy], max_depth=[3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 20, 30]	criterion=gini, max_depth=20
RF	Character based fourgram with TF-IDF	n_estimators=[10, 100, 500, 1000], max_features=[auto, sqrt, log2]	n_estimators=1000 max_features=auto
MNB	Word based unigram with TF-IDF	alpha=[0.0001, 0.001, 0.01, 1]	alpha=0.01

4. Results

It is known that Accuracy, F1-measure, Precision and Recall metrics are used to evaluate the performance of a text classifier. In our work, the Accuracy metric was used. This metric is defined as follows:

$$f(z,u) = \begin{cases} 1, & z = u \\ 0, & otherwise \end{cases}$$

$$Accuracy(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N} f(y_i,\hat{y}_i)$$

Here, N is number of all test samples, y_i is actual class name of $i-$th test sample, $\hat{y}_i$ is predicted class name of $i-$th test sample, y_i, $\hat{y}_i \in \{'world','science','culture','local','show-business','sport','technologies','auto','photo','movie'\}$, $i = \overline{1,N}$, $y = \{y_1, ..., y_N\}$, $\hat{y} = \{\hat{y}_1, ..., \hat{y}_N\}$.

The prepared dataset rows are randomly split to 80% for training and 20% for testing and experimented with them. Table 4 shows the results of a multi-class text classification of news articles data written in Uzbek. The table contains the results of six machine learning classifiers and the accuracy obtained after applying six feature selection methods.

Table 4. Results of multi-class text classification on Uzbek news articles dataset.

Classifier	TF-IDF with word level n-gram			TF-IDF with character level n-gram		
	unigram	bigram	trigram	bigram	trigram	fourgram
Linear SVC	86,39	80,79	71,64	81,06	86,12	86,54
RBF SVM	85,63	80,26	71,64	82,16	86,16	**86,88**
DTC	68,51	55,88	44,50	61,02	68,28	71,00
LR	85,44	80,04	72,59	79,28	85,56	86,20
MNB	83,18	78,37	71,42	69,83	81,44	82,57

As seen from the table 4, the highest accuracy of 86.88% was obtained when using the character level fourgram and TF-IDF of the feature vector, as well as when performing support vector machine classifier using the radial basis function kernel. If we consider the results on the word level n-gram and character level n-gram models separately, then the accuracy obtained by TF-IDF and word level n-gram feature selection based on unigram models is higher than the accuracy obtained for all classifiers using bigram and trigram models. However, in TF-IDF and character level n-gram feature selection method, on the contrary, the accuracy obtained using the fourgram model is higher than the accuracy obtained using the bigram and trigram models. The reading time of machine learning text classifiers has changed depending on the use of unigram, bigram, trigram, fourgram types of the n-gram model. This is considered important in real life application.

To summarize the results of the classification algorithm, the confusion matrix is usually used. Therefore, to derive this matrix based on experiments, we determined the classifier and feature selection method, which gave the highest accuracy. Based on the application of this classifier and method, a confusion matrix was obtained. The elements of this matrix are presented in Table 5.

5. Conclusion and Future work

A dataset was collected for the multi-class text classification task based on online news articles of the "Daryo" online edition which is one of the most popular and mass sites in the Republic of Uzbekistan. Ten news categories were selected for data collection and a maximum of 2000 articles were collected from each category. In the collected dataset, in accordance with each article, the average number of words was 232, and the average number of characters was 1956. As a result, the dataset includes 13224 article texts in corresponding categories. The authors have this dataset. It can be submitted upon request with the permission of the special department of the daryo.uz edition.

The stop words list consisting of 373 words was developed for Uzbek language. The list was used for preliminary processing of the article texts. This list is uploaded on GitHub and can be freely downloaded and used.

Table 5. Confusion matrix for RBF SVM classifier using character level fourgrams TF-IDF feature vector

		Predicted class										
		Auto	World	Photo	Science	Movie	Culture	Local	Show-business	Sport	Technologies	Total
Actual class	Auto	250	3	3	0	1	3	1	0	0	8	269
	World	7	363	4	9	1	5	15	0	0	12	416
	Photo	1	16	191	0	4	7	0	0	3	1	223
	Science	0	6	2	51	0	2	6	0	0	5	72
	Movie	0	2	1	0	94	23	0	0	0	2	122
	Culture	0	5	8	0	36	243	8	6	0	2	308
	Local	4	3	0	0	0	6	394	0	4	4	415
	Show-business	0	1	4	0	1	69	1	8	0	0	84
	Sport	0	1	0	0	0	2	0	0	414	0	417
	Technologies	3	13	3	4	0	3	3	0	0	290	319
	Total	265	413	216	64	137	363	428	14	421	324	2645

The character level and word level n-gram models and TF-IDF algorithms were used to implement feature extraction of Uzbek online news articles. Six machine learning technologies were used to complete the multi-class text classification task. To determine the classifier hyperparametrs, 5-fold cross-validation and Grid search algorithms were used. When executing features extraction, we applied several types of n-gram model: unigram, bigram, trigram and fourgram. For each case, machine learning classifiers were used and the results were obtained. The results showed that in many cases the character level n-gram gives better results than the word level n-grams. The best accuracy was 86.88% when using RBF SVM classifier and TF-IDF with character level fourgram feature selection method.

Designed and implemented algorithms and software will be included in special environment that we are working at now - the Work Station for Linguists [21].

We also plan to expand the dataset to multi-label classification tasks, to increase the number of articles, to apply Deep Learning and Word Embeddings technologies in order to solve the text classification problems. Gradually, the stop words list for Uzbek language will be extended.

References

[1] Hmeidi I, Al-Ayyoub M, Abdulla N A, Almodawar A A, Abooraig R and Mahyoub N A 2015 *Journal of Information Science* **41(1)** 114–24

[2] Deniz A and Kiziloz H E 2017 *International Conference on Computer Science and Engineering (UBMK)* (Antalya: IEEE) pp 655–60

[3] Omurca S İ, Baş S and Ekinci E 2015 *International Journal Of Intelligent Systems And Applications In Engineering* **3(1)** 7-13

[4] Gürcan F 2018 *2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (Ankara: IEEE) pp 1-5

[5] Shehab M Badarneh O Al-Ayyoub M and Jararweh Y 2016 *7th International Conference on Computer Science and Information Technology (CSIT)* (Amman: IEEE) pp 1-6

[6] Ahmed N A Shehab M A Al-Ayyoub M and Hmeidi I 2015 *6th International Conference on Information and Communication Systems (ICICS)* (Amman: IEEE) pp 212-17
[7] Alsaleem S 2011 *International Arab Journal of e-Technology* **2(2)** 124-28
[8] Wu X Fang L Wang P and Yu N 2015 *IEEE 28th Canadian Conference on Electrical and Computer Engineering* (Halifax: IEEE) pp 1260-64
[9] Wei Z Miao D Chauchat J-H Zhao R and Li W 2009 *International Journal of Computational Intelligence Systems* **2(4)** 365-74
[10] Sembodo J E Setiawan E B and Bijaksana M A 2018 *6th International Conference on Information and Communication Technology (ICoICT)* (Bandung: IEEE) pp 389-93
[11] Wongso R Luwinda F A Trisnajaya B C Rusli O and Rudy 2017 *Procedia Computer Science* **116** 137-43
[12] Bučar J Povh J and Žnidaršič M 2016 *9th International Conference on Computer Recognition Systems CORES 2015* (Wroclaw: Springer) pp 777–87
[13] Parhat S Ablimit M Hamdulla A 2019 *Information* **10(12)** 387
[14] Bracewell D B Yan J Ren F and Kuroiwa S 2009 *Electronic Notes in Theoretical Computer Science* **225** 51-65
[15] Romanov A Lomotin K and Kozlova E 2019 *Data Science Journal* **18(1)**
[16] Kandhro I A Jumani S Z Lashari A A Nangraj S S Lakhan Q A Baig M T Guriro S 2019 *Indian Journal of Science and Technology* **12** 33
[17] Duong H-T Hoang V T 2019 11th *International Conference on Knowledge and Smart Technology (KST)* (Phuket: IEEE) pp 23-28
[18] Kuriyozov E Matlatipov S Alonso M A Gómez-Rodríguez C 2019 *Human Language Technologies as a Challenge for Computer Science and Linguistics - 2019* (Roznan) pp 258-262
[19] Kobilov S Rabbimov I 2015 *Science and World* **6 (22)** 21-23 (in Russ.)
[20] Pedregosa F et al 2011 *Journal of Machine Learning Research* **12** 2825-30
[21] Kobilov S Rabbimov I 2017 *Uzbek Journal of the Problems of Informatics and Energetics* **1** 27-33 (in Russ.)