MACHINE LEARNING AND NEURAL NETWORK METHODOLOGIES OF ANALYZING SOCIAL MEDIA

Vladislav Karyukin
Al-Farabi Kazakh National University,
Faculty of Information Technologies,
Almaty, Kazakhstan
al-Farabi av.71
telephone: +7 701 9405992,
vladislav.karyukin@kaznu.kz

Aidana Zhumabekova
Al-Farabi Kazakh National University,
Faculty of Information Technologies,
Almaty, Kazakhstan
al-Farabi av.71
telephone: +7 701 1066721
zhumabekova2702@gmail.com

Sandugash Yessenzhanova
Al-Farabi Kazakh National University,
Faculty of Information Technologies,
Almaty, Kazakhstan
al-Farabi av.71
telephone: +7 707 2728338
esandu92@gmail.com

ABSTRACT

The rapid development of the Internet has led to a significant increase in the number of news sites and social networks that describe various events in the world and society. People actively share their opinions about various events in the world. Manually tracking and analyzing such a volume of information is not possible. So, in this way, the use of algorithms for automatic analysis of texts and user comments is an important feature. Published articles and user comments in most cases are of a certain emotional aspect. This article analyzes texts and user comments of Kazakhstan media space. Sentiment classification is done using machine learning algorithms and convolutional and recurrent neural networks (CNN and RNN). A comparative review of the obtained results was performed after the classification.

CCS Concepts

• **Computing methodologies → Artificial Intelligence → Natural language processing • Computing methodologies → Machine learning → Machine learning algorithms • Computing methodologies → Machine learning → Machine learning approaches → Neural networks.**

Keywords

Social media; sentiment analysis; data processing; stemming; machine learning algorithms; fastText; CNN; LSTM.

1. INTRODUCTION

Main events and people's attitudes towards them are a very significant aspect of society. It is necessary to understand how the perception of various events and the interaction of users with each other occur, which will help to highlight the attitude of people to certain events happening in society and to determine the most important problems that government agencies need to pay more attention to.

Publications of texts and user comments on news portals and in popular social networks in Kazakhstan: VK, Facebook, Twitter, and Instagram, allow you to get a large amount of textual data for analysis. At the same time, texts in the Russian and Kazakh languages are important features of sentiment analysis. To work with the texts in these languages, it is necessary to place a big emphasis on the data preprocessing stage realized with the use of appropriate stemmers and lemmatizers. Word embedding models trained for these languages must be used for vectorizing data. In this work, text preprocessing has been successfully performed, and the corresponding word vectorization models have been implemented. In addition, data have been successfully classified with the use of machine learning algorithms and neural networks. The results obtained are visualized and presented in the form of appropriate tables.

2. RELATED WORKS

The problem of sentiment analysis [1] was started to be dealt quite long ago. A large number of relevant works were devoted to it. Due to the growing interest in this topic, companies were very involved in conducting events to attract more and more customers [2]. They were interested in knowing what people thought and said about them. At the same time, sentiment analysis was important both in the social and political environment. Officials and politicians wanted to know their reputation and how people thought about them in media resources. Also, many texts in medicine, sports, and society were also widely discussed and provoked a huge public reaction [3]. Thus, sentiment analysis is still of great importance in various fields today.

At SemEval international seminars, a large number of unlabeled and labeled data were presented. Labeled data were divided into the following categories: binary – positive and negative sentiment, ternary – positive, neutral and negative sentiment, and multiple – strongly positive, positive, neutral, negative, and strongly negative. At SemEval 2014, sentiment dictionaries and machine learning algorithms were proposed. In the dictionary approach, the sentiment of a sentence or a text is determined by counting a number of positive, neutral, and negative words and dividing them by the total number of words. At the SemEval 2016 seminar, linear regression, Gaussian regression, and random forest machine learning algorithms were used to classify labeled texts. Cleaning and classification of labeled texts with the use of such widespread machine learning algorithms as maximum entropy, Naive Bayes classifier, and support vector machine were presented in the work [4].

In addition to machine learning algorithms, approaches with the use of neural networks started to gain wide popularity. For a large number of textual data, they showed good results. Sentiment analysis of texts with the use of a deep convolutional neural network (CNN) in mobile applications was realized in the article

[5]. Sentiment analysis with CNN was also done in the work [6] where vectorization of texts using the word2vec model and a seven-layer neural network were applied. The binary classification of Indonesian posts using a recurrent neural network (RNN) and the word2vec model was demonstrated in the article [7].

3. METHODOLOGY

For the automatic classification of labeled texts, it is necessary to perform the main stages of obtaining and preprocessing data, as well as classifying it with the use of machine learning algorithms and neural networks.

The first stage includes obtaining textual data from Kazakhstan news portals. The second stage contains data preprocessing that consists of clearing data from unnecessary characters and punctuation marks, reducing all words to lower case, removing stop words that do not carry any semantic load, and removing affixes and endings (stemming) or transforming words into an indefinite form (lemmatization). The third stage is the conversion of words in sentences and texts into a vector form. The fourth stage includes classification using the most popular machine learning algorithms. The fifth stage involves classification with the use of CNN and RNN. At the final stage, the obtained results of classification are carried out in the form of graphs and tables.

3.1 Datasets

Data for the classifier were gathered with the use of web-crawler in OMSystem, an analytical platform of monitoring Kazakhstan media space. This application has been developed at al-Farabi Kazakh National University. Texts and comments have been collected from such popular news portals as tengrinews.kz, zakon.kz, nur.kz, informburo.kz, times.kz, today.kz and bnews.kz. Then they were labeled by experts as positive and negative ones. To avoid overfitting due to the predominance of one class over the other one, the balance of positive and negative classes in equal proportions was kept. The number of texts and comments is 1040 and 7514 for Russian, and 434 and 346 for Kazakh, respectively.

3.2 Data preprocessing

The data preprocessing phase includes the removal of extra characters and punctuation marks, as well as stop words. Removing stop words is necessary, since they are very often found in texts and do not carry any functional load. The next important step is stemming that is realized by removing suffixes and endings from words so that only the foundation remains, which is the same for all its forms. For Russian words, SnowballStemmer from the NLTK library in the Python programming language has been used. For Kazakh words, new Kazakh_Stemmer has been developed. The last stemmer removes unnecessary characters and stop words using the regular expression *"^ a-zA-Za-яА-ЯӘІҢҒҮҰҚӨәіңғүұқөh".* The database of endings and suffixes of the Kazakh language is implemented to accomplish stemming. Endings and suffixes are compared with parts of a word and cut in case of coincidence. If a word has a length that is less than or equal to two, it remains unchanged. This limitation arises from the fact that some two-letter words are similar to endings and suffixes.

3.3 Vectorization of texts

Such widely used classical approaches as "Bag of words" where each word is represented as a vector containing only one "1" number and many zeros, and TF-IDF metric are implemented for vectorization.

TF-IDF metric is represented as follows

$$IDF_{d,w} = \log \frac{count(D)}{count(w,d)}, \quad (1)$$

where $count(w,d)$ is a number of times a word w is found in a document d, and $count(N,d)$ is the total number of words N in the document d.

$$TF_{d,w} = \frac{count(w,d)}{count(N,d)}, \quad (2)$$

where $count(D)$ is the total number of documents, $count(w,d)$ is a number of times the word w is found in the document d.

Classification with the use of deep neural networks requires a large amount of train data that is not always freely available. The solution to this problem can be a huge pretrained corpus of data. The most common models are word2vec [8-9], Glove [10], and fastText [11]. Typically, word embedding patterns ignore the word morphology and study a separate vector for each word. In contrast, fastText creates a vector for each n-gram character. Then the word vector is calculated as the average of its symbolic vectors of the n-gram. The effectiveness and accessibility of trained fastText models for the Russian and Kazakh languages was the main reason for their use in this work.

3.4 Classification with the use of machine learning algorithms

After vectorization of texts, classification is completed using the most widespread machine learning algorithms: Logistic regression, Naive Bayes classifier [12], Support vector machine [13], K-nearest neighbors [14], Decision tree [15] and Random forest [16-17].

Naive Bayes classifier is a very easy and effective vector analysis method that shows results no worse than other more complex classifiers. This classifier is based on the conditional probability, according to which the document d belongs to the class c, and uses the Bayes formula. For a text analysis model, the following formula is used

$$P(c \mid d) = \frac{P(c) \cdot P(d \mid c)}{P(d)}, \quad (3)$$

where vector $d = \{x_1, x_2, ..., x_n\}$, x_i is a weight of the i^{th} word in the document d, and c is the class of the document.

In Support vector machine, a hyperplane that divides the dataset into two classes: positive and negative, is drawn. This hyperplane is given by the following formula:

$$(\vec{x}, \vec{y}) + b = \sum_i y_i \cdot a_i \cdot (\vec{x}, \vec{y}) + b = 0, \quad (4)$$

where $\vec{x} = (x_{i1}, x_{i2}, ..., x_{in})$ is an input vector, y_i is a corresponding output value, $\vec{w} = (w_1, w_2, ..., w_n)$ is a weight vector that defines the hyperplane and a_i are Lagrange multipliers. When the hyperplane is constructed, the class to which the input vector belongs is determined by

$$\vec{w} \cdot \vec{x} + b \geq 0. \quad (5)$$

If the value is more than or equal to 0, it refers to a positive class. Otherwise, it belongs to a negative class.

The K-nearest neighbors method allows you to classify documents using the Euclidean space, which determines the distance between two vectors

$$d(x,y)=\sqrt{\sum_{i=1}^{N}(a_{ix}-a_{iy})^2}\ , \tag{6}$$

where $d(x,y)$ is the distance between two documents, N is the number of unique words in the list of documents, a_{ix} is a weight of the i^{th} term in the document x, and a_{iy} is a weight of the i^{th} term in the document y.

Decision trees are a fast way to classify data. It is a structure with k-nodes in which a test is performed on the features of data. When constructing a decision tree, the following rules are applied. A word is selected, and the documents containing it are put on one side, and the documents not containing it are put on the other side. Thus, the documents belong to two disjoint sets. For each set, a new word is selected, and the step described above is again performed. The procedure is repeated until a uniform set is obtained, in which all documents belong to the same class. A random forest algorithm uses many decision trees. They are built independently from each other. The document is classified by all trees, and its class is determined by the highest number of trees voted for it.

3.5 Classification with the use of neural networks

Convolutional neural networks [18-19] gained great popularity due to their use with images, in which a given filter moves throughout the image. Since images are two-dimensional objects, and texts are one-dimensional ones, the appropriate transformations have to be done to apply the filter. Sentences in the texts have different lengths. Therefore, it is necessary to cut off sentences that are too long or to supplement ones that are too short. In order to do it, we define S as a maximum length of the sentence. When converting, we get a vector of a word of dimension E. Vectorization is performed using the fastText model. In the $E \times S$ matrix, semantically close words are next to each other, and words with different meanings are far away. Next, the first filter $C_1 \times E$ is applied to the matrix. The scalar product of the matrix and filter elements is performed. The result value is transferred to the next layer. As a result of transformations, vectors of smaller sizes are obtained. At the end of the transformations, the text is classified as positive or negative (Figure 1).

One of the most popular RNN models is LSTM [20] (Figure 2). The network structure includes 4 parts: the output gate O_t, the forget gate F_t, the memory cell C_t and the input gate I_t. All three gates take the input of the current time interval and the output of the past time interval. New data come in through F_t and I_t. The O_t output for the current interval is generated by the system. The architecture of LSTM is described by (7)-(10):

$$f_t=\varphi(W_t\cdot[h_{t-1},x_t]+b_f) \tag{7}$$

$$i_t=\varphi(W_t\cdot[h_{t-1},x_t]+b_i) \tag{8}$$

$$O_t=\varphi(W_0\cdot[h_{t-1},x_t]+b_0) \tag{9}$$

$$C_t^{'}=\tanh(W_c\cdot[h_{t-1},x_t]+b_c) \tag{10}$$

4. EXPERIMENTS AND RESULTS

After cleaning and vectorizing labeled text data with the use of TF-IDF metric, classification with machine learning algorithms is done. The text data is divided into train 80% and test 20%. Then each of the machine learning algorithms is applied to the vectorized data. To evaluate the results, the test on the remaining 20% of the data is implemented. Machine learning metrics: accuracy, precision, recall, and F-measure, as well as graphs of ROC curves estimate the quality of the trained model [21].

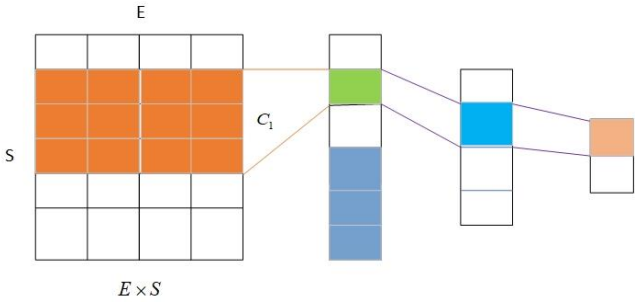

Figure 1. CNN classification

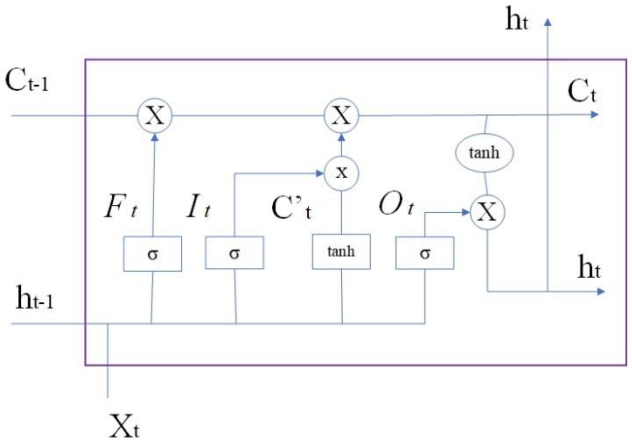

Figure 2. LSTM architecture

The values of metrics for Russian texts and comments are presented in Tables 1, 2. Graphs of ROC curves are demonstrated in Figures 3, 4. The highest accuracy, precision, recall, and F-measure are obtained by the use of logistic regression, Naïve Bayes classifier, support vector machine, and random forest algorithms. K-nearest neighbors and decision tree showed worse results both for texts and comments. Graphs of ROC curves confirm that texts are classified much more precisely than comments.

Table 1. Metrics of classification algorithms for Russian texts

Algorithm	Accuracy	Precision	Recall	F-meas.
Logistic regression	0.82	0.83	0.84	0.83
Naïve Bayes	0.81	0.82	0.82	0.82
K-nearest	0.7	0.82	0.55	0.66

neighbors				
Support vector machine	0.82	0.85	0.79	0.82
Decision tree	0.66	0.68	0.68	0.68
Random forest	0.82	0.81	0.86	0.83

Table 2. Metrics of classification algorithms for Russian comments

Algorithm	Accuracy	Precision	Recall	F-meas.
Logistic regression	0.68	0.68	0.66	0.67
Naïve Bayes	0.68	0.67	0.68	0.68
K-nearest neighbors	0.58	0.62	0.4	0.48
Support vector machine	0.68	0.69	0.65	0.67
Decision tree	0.6	0.58	0.62	0.6
Random forest	0.65	0.63	0.64	0.64

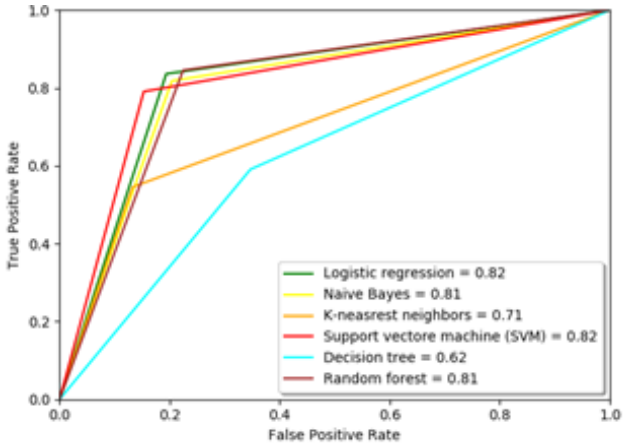

Figure 3. ROC curves for Russian texts

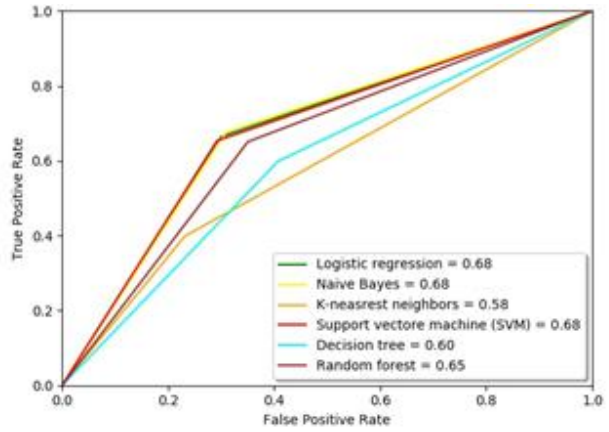

Figure 4. ROC curves for Russian comments

Values of metrics for Kazakh texts and comments are presented in Tables 3, 4. Graphs of ROC curves are demonstrated in Figures 5, 6. All six algorithms showed good results of classification for texts, but logistic regression, Naïve Bayes classifier, support vector machine, and random forest were slightly better for comments.

Table 3. Metrics of classification algorithms for Kazakh texts

Algorithm	Accuracy	Precision	Recall	F-meas.
Logistic regression	0.86	0.87	0.85	0.86
Naïve Bayes	0.86	0.84	0.89	0.87
K-nearest neighbors	0.83	0.9	0.74	0.81
Support vector machine	0.85	0.85	0.85	0.85
Decision tree	0.82	0.83	0.8	0.81
Random forest	0.81	0.9	0.7	0.79

Table 4. Metrics of classification algorithms for Kazakh comments

Algorithm	Accuracy	Precision	Recall	F-meas.
Logistic regression	0.66	0.75	0.64	0.69
Naïve Bayes	0.69	0.76	0.69	0.73
K-nearest neighbors	0.57	0.77	0.4	0.53
Support vector machine	0.64	0.74	0.62	0.68
Decision tree	0.57	0.69	0.52	0.59
Random forest	0.61	0.74	0.55	0.63

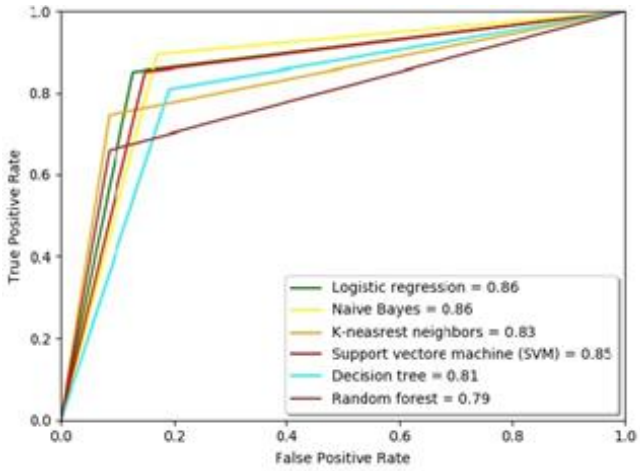

Figure 5. ROC curves for Kazakh texts

To classify texts and comments using neural networks, data is vectorized using the fastText model. On the next step, data are divided into train 60%, validation 20%, and test 20% parts. In the CNN, three convolutional layers, one pooling layer, and one dropout layer are used as it was stated in [6]. Since there is not much data for classification, 3-5 epochs are enough to avoid overfitting.

The results of the classification of Russian and Kazakh texts and comments are presented in Table 5 and Figure 7. CNN demonstrated good results for classifying texts and comments. The results are mostly equal to ones obtained above by the use of machine learning algorithms.

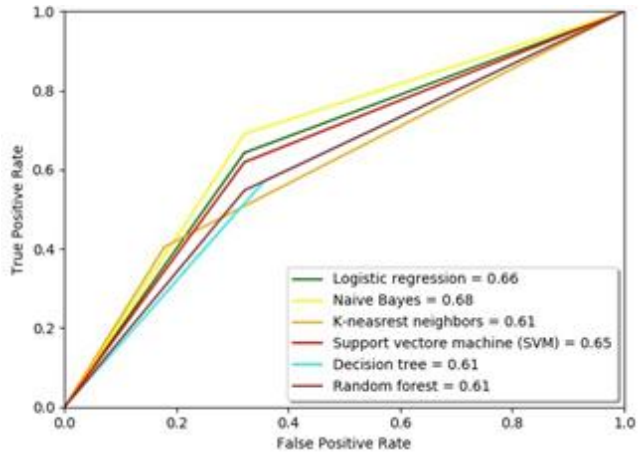

Figure 6. ROC curves for Kazakh comments

Table 5. Metrics of texts and comments classification with CNN

Metrics	Russian texts	Russian comments	Kazakh texts	Kazakh comments
Accuracy	0.79	0.68	0.86	0.7
Precision	0.77	0.66	0.93	0.79
Recall	0.83	0.68	0.7	0.65
F-measure	0.8	0.67	0.8	0.71

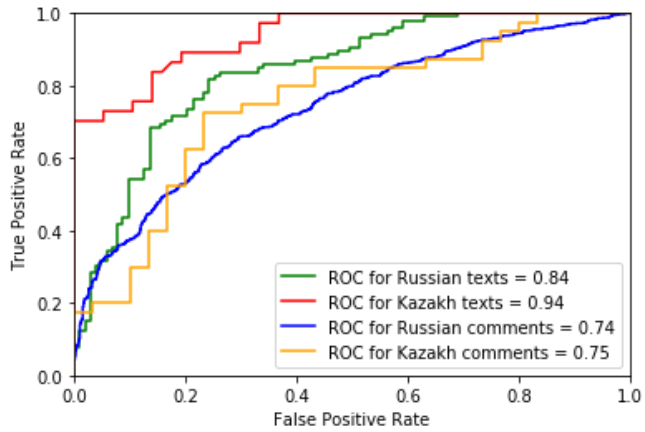

Figure 7. ROC curves for classification with CNN

Then, classification with LSTM was realized. Data were divided into train 60%, validation 20%, and test 20% parts. Two hidden layers, two fully connected layers, and one dropout layer were used. As LSTM neural network works longer, 10-20 epochs were put in the training process. The results of the classification of Russian and Kazakh texts are presented in Table 6 and Figure 8. LSTM neural network demonstrated significantly worse results for texts and comments. Thus, CNN is preferred in this experiment.

Table 6. Metrics of texts and comments classification with LSTM

Metrics	Russian texts	Russian comments	Kazakh texts	Kazakh comments
Accuracy	0.52	0.48	0.8	0.42
Precision	0.8	0.48	1	0.01
Recall	0.08	1	0.49	0.01
F-measure	0.14	0.65	0.65	0.01

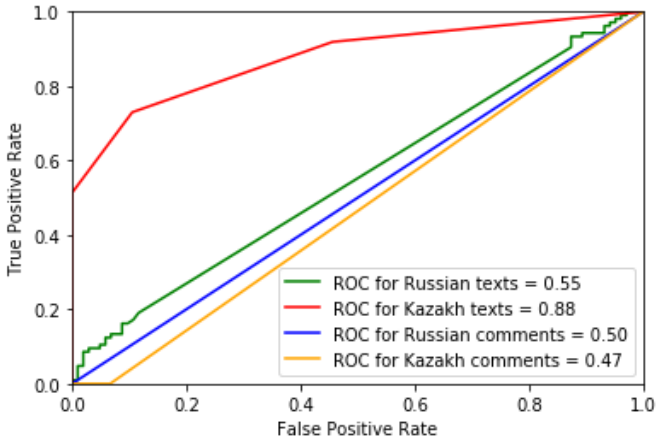

Figure 8. ROC curves for classification with LSTM

5. CONCLUSION

Sentiment analysis of texts and user comments showed that the values of metrics had demonstrated better results for full texts of articles than comments. This is due to the fact that the texts of articles are much longer, and they are well structured and contain a small number of errors. At the same time, comments are often very short and include few words and characters, many of which are written with errors. For a small number of texts and comments, classification metrics of machine learning algorithms and neural networks are comparable. This is because neural networks require large datasets. Nevertheless, RNN demonstrated much worse results than CNN, so it is supposed that this neural network requires a significant amount of data. Among machine learning algorithms, Naive Bayes classifier, logistic regression, and support vector machine showed the best results. Generally, machine learning algorithms and neural networks worked well even taking into account comparatively small datasets.

This research showed which of the used algorithms better trained datasets of texts and comments in the Russian and Kazakh languages. It would be useful to design more advanced algorithms that could give improved results in the evaluation of user data. Such algorithms can be implemented for monitoring and evaluating social media space in the analytical platforms.

6. REFERENCES

[1] Keyvanpour, M., Karimi Zandian, Z., Heidarypanah, M. OMLML: a helpful opinion mining method based on lexicon and machine learning in social networks. *Soc. Netw. Anal. Min.* 10**,** 10 (2020). DOI=https://doi.org/10.1007/s13278-019-0622-6.

[2] Ananiadou, S., Thompson, P., Nawaz, R. (2013) Enhancing Search: Events and Their Discourse Context. In: Gelbukh A. (eds) *Computational Linguistics and Intelligent Text*

Processing. CICLing 2013. Lecture Notes in Computer Science, vol 7817. Springer, Berlin, Heidelberg. DOI= https://doi.org/10.1007/978-3-642-37256-8_27.

[3] Araque, O., Zhu, G., Iglesias, C.A. A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, vol. 165, pages 346–359 (2019). DOI= https://doi.org/10.1016/j.knosys.2018.12.005.

[4] Alam, S., Yao, N. The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Comput Math Organ Theory* 25**,** 319–335 (2019). DOI= https://doi.org/10.1007/s10588-018-9266-8.

[5] Ouyang, X., Zhou, P., Li, C.H. and Liu L. "Sentiment Analysis Using Convolutional Neural Network," *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, 2015, pp. 2359-2364, DOI**=** 10.1109/CIT/IUCC/DASC/PICOM.2015.349.

[6] Smetanin, S., Komarov M., "Sentiment Analysis of Product Reviews in Russian using Convolutional Neural Networks," *2019 IEEE 21st Conference on Business Informatics (CBI)*, Moscow, Russia, 2019, pp. 482-486, DOI= 10.1109/CBI.2019.00062.

[7] Kurniasari, L., Setyanto, A. Sentiment Analysis using Recurrent Neural Network. *Journal of Physics: Conference Series*, vol. 1471, 1st Bukittinggi International Conference on Education, West Sumatera, Indonesia (2019). DOI= https://doi.org/10.1088/1742-6596/1471/1/012018.

[8] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* 2013, arXiv:1301.3781 (2013).

[9] Yilmaz, S., Toklu, S. A deep learning analysis on question classification task using Word2vec representations. *Neural Comput & Applic* **32,** 2909–2928 (2020). DOI= https://doi.org/10.1007/s00521-020-04725-w.

[10] Sakketou, F., Ampazis, N. A constrained optimization algorithm for learning GloVe embeddings with semantic lexicons, *Knowledge-Based Systems*, 195, 105628 (2020). DOI= https://doi.org/10.1016/j.knosys.2020.105628.

[11] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguist*, 5, pages 135–146 (2017).

[12] Samiul Salehin, S.M., Miah R., Saiful Islam Md. A comparative sentiment analysis on Bengali Facebook posts. *ICCA 2020: Proceedings of the International Conference on Computing Advancements*, 21, Pages 1–8. DOI= https://doi.org/10.1145/3377049.3377078.

[13] Syahputra, H., Basyar, L.K., Tamba, A.A.S. Setiment Analysis of Public Opinion on The Go-Jek Indonesia Through Twitter Using Algorithm Support Vector Machine. *Journal of Physics: Conference Series*, 1462. (2020). DOI= https://doi.org/10.1088/1742-6596/1462/1/012063.

[14] Bruno T., Sasa M., Dzenana D. KNN with TF-IDF based Framework for Text Categorization. *Procedia Engineering*, vol. 69, pages 1356-1364 (2014). DOI= https://doi.org/10.1016/j.proeng.2014.03.129.

[15] Rajesh Kumar, E., Rama Rao, K.V.S.N., Nayak S.R., Chandra R. Suicidal ideation prediction in twitter data using machine learning techniques. *Journal of Interdisciplinary Mathematics*, 23, 1, pages 117-125 (2020). DOI= https://doi.org/10.1080/09720502.2020.1721674.

[16] Alam, S., Yao, N. The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis. *Comput Math Organ Theory*, 25**,** 319–335 (2019). DOI= https://doi.org/10.1007/s10588-018-9266-8.

[17] Singh, N.K., Tomar, D.S. & Sangaiah, A.K. Sentiment analysis: a review and comparative analysis over social media. *J Ambient Intell Human Comput*, 11**,** 97–117 (2020). DOI= https://doi.org/10.1007/s12652-018-0862-8.

[18] Ghorbani, M., Bahaghighat, M., Xin, Q. *et al.* ConvLSTMConv network: a deep learning approach for sentiment analysis in cloud computing. *J Cloud Comp* 9**,** 16 (2020). DOI= https://doi.org/10.1186/s13677-020-00162-1.

[19] Sankar H., Subramaniyaswamy V., Vijayakumar V., Arun Kumar S., Logesh R., Umamakeswari A. Intelligent sentiment analysis approach using edge computing-based deep learning technique. *Software: Practice and Experience*, vol. 50, issue 5, pages 645-657 (2020). DOI= https://doi.org/10.1002/spe.2687.

[20] Zhang, Qiang & Gao, Tianze & Liu, Xueyan & Zheng, Yun. Public Environment Emotion Prediction Model Using LSTM Network. Sustainability. 12. 1665 (2020). DOI= https://doi.org/10.3390/su12041665.

[21] Bianchi, B., Bengolea Monzón, G., Ferrer, L. *et al.* Human and computer estimations of Predictability of words in written language. *Sci Rep* 10**,** 4396 (2020). DOI= https://doi.org/10.1038/s41598-020-61353-z.

Authors' background

Your Name	Title*	Research Field	Personal website
Vladislav Karyukin	PhD student	Data science, Natural language processing, Machine learning, Neural networks	https://pps.kaznu.kz/ru/Main/Personal/90/446/8799/%D0%9A%D0%B0%D1%80%D1%8E%D0%BA%D0%B8%D0%BD%20%D0%92%D0%BB%D0%B0%D0%B4%D0%B8%D1%81%D0%BB%D0%B0%D0%B2%20%D0%98%D0%B3%D0%BE%D1%80%D0%B5%D0%B2%D0%B8%D1%87#pills-profile
Aidana Zhumabekova	PhD student	Data science, information security, data forensics, blockchain	https://pps.kaznu.kz/ru/Main/Personal/90/348/13345/%D0%96%D2%B1%D0%BC%D0%B0%D0%B1%D0%B5%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%90%D0%B9%D0%B4%D0%B0%D0%BD%D0%B0%20%D0%A2%D3%A9%D0%BB%D0%B5%D1%83%D2%9B%D1%8B%D0%B7%D1%8B#pills-contact
Sandugash Yessenzhanova	Master student	Data science, machine learning, neural networks	

***This form helps us to understand your paper better, the form itself will not be published.**

***Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor**