ISSN 2007-9737

Topic-Aware Sentiment Analysis of News Articles

Iskander Akhmetov[1,2], Alexander Gelbukh[3], Rustam Mussabayev[1]

[1] Institute of Information and Computational Technologies (IICT), Kazakhstan

[2] Kazakh-British Technical University (KBTU), FIT, Kazakhstan

[3] Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación, Mexico

i.akhmetov@ipic.kz, gelbukh@gelbukh.com

Abstract. We consider the problem of sentiment analysis in news media articles cast as a three-way classification task: negative, positive, or neutral. We show that subdividing the training corpus by topic (local news, sports, hi-tech, and others) and training separate sentiment classifiers for each sub-corpus improves classification F1 scores. We use topics since some words carry different sentiments in different domains: e.g., the word "force" is typically positive in the sports domain but negative in the political domain. Our experiments on the Kaggle dataset with sentiment-labeled Kazakhstani news articles in Russian language using the Convolutional Neural Network (CNN) model partially proved our hypothesis, showing that for the most prominent "kz" (local news) topic, we achieve an F1 score of 0.70, which is greater than the baseline model trained without the topic-awareness showing just 0.67. Topic-aware improves F1 scores in some topics, but due to the topic/class imbalance further research is needed. However, the performance in terms of F1 over all the corpus does not improve or the improvements are very small. Moreover, our approach shows better results on topics with many text samples than those with relatively small amounts of articles.

Keywords. Mass media, natural language processing, news articles, sentiment analysis.

1 Introduction

Sentiment analysis is a task of classifying a given text as expressing *negative*, *neutral* or *positive* feeling which is thought to have emotional impact on the reader. Thus, it is becoming essential for governments and corporations to monitor the sentiments expressed on the sensitive matters connected to public administration and customer feed-backs.

It has recently found numerous applications, which range from opinion mining, product review analysis, fake news detection [13], and government mass media monitoring to forensics and public relations tools.

Furthermore, automatic text sentiment analysis is also an essential feature in the rising number of developments in the area of IoT where it has numerous applications [12].

The motivation of our work is to achieve better F1 scores in sentiment analysis of news articles elaborating on the hypothesis that prior topic classification of articles and selecting domain-specific sentiment model will improve the results over the traditional methods used. Therefore, the objectives were to build the topic classification models and respective domain-specific sentiment classifiers.

The novelty of our work is that this is the first topic-aware sentiment analysis research performed on the mass media news articles showing that domain-specific models outperform mixed domain models. The usual approach to the sentiment analysis task is to train a classifier on all available texts of a given corpus, irrespective of their topics; see Fig. 1 (a). However, the exact words or expressions can carry different sentiments in texts on different topics: for example, in sports news, the word *"aggressive"* could mean something positive in terms of athletic achievements, while in political news, the same word means something negative, referring to civil riots, war accidents [4, 6], and similar events.

In this work, we elaborate on the question if we can achieve a greater F1 scores by training separate models for different topics present in the corpus, aiming to disambiguate word meanings from topic to topic.

Namely, we first classify the input texts by their topics; then, depending on the topic of a specific document identified by the topic classifier, we choose one of the sentiment classification models: the one trained with the documents on this topic, see Fig. 1 (b).

Thus, the topic classifier serves as a meta-classifier that chooses a specific model to use (hence the index finger in the figure). We call this strategy a topic-aware sentiment classifier.

We consider a case study of Kazakhstani news articles in the Russian language. In our corpus, each news article related to one of several topics, such as sport or politics, indicated by the original publisher of the news. With this information, we:

1. Separate the corpus into sub-corpora corresponding to the topics and train a separate sentiment classifier for each sub-corpus;
2. Train a classifier that assigns a given document to one of the topics to choose the sentiment model trained on the corresponding sub-corpus.

The main contributions of this article are:

— A topic-aware approach for sentiment analysis,

— Augmenting three existing sentiment-labeled text corpora of Kazakhstani news articles in the Russian language with topic labels and lemmatization.

The code of our topic-aware classifier and the augmented corpora are available to researchers [2] [1].

The rest of this paper is organized as follows. In Section 2 we give a review of the related works. In Section 3, we describe the data we used. In Section 4, we present our classification method. In Section 5, we describe our experiments. In Section 6, we discuss our experimental results. Finally, Section 7 concludes the paper and outlines future work directions.

2 Related Work

Addressing the importance and necessity of classification of texts by topics with high accuracy for better information extraction, [19] classified Twitter trending topics into 18 common categories such as sports, technology, politics, and others.

Furthermore, they experimented with two methods for topic classification: (i) Bag-of-Words (BOW) and (ii) network-based classification. Experiments on a dataset of randomly selected 768 articles with trending topics (within 18 classes) show achieving topic classification accuracy of up to 65% and 70% using BOW and network-based classification models, respectively [19]. Mining the domain shared information for sentiment analysis modeling [34] sets the perspective for targeting at building different sentiment classification models for each topic, however, they worked with Amazon product reviews and used topic modeling.

Further, [27] worked on the implementation of domain-specific sentiment analysis of news streams in Russia and showed sentiment and opinions on news related to Ukraine and the Paris agreement, which were the hot topics in 2018. [31] also suggested and implemented a method of topic-wise selection of negative

[1] Code: https://github.com/iskander-akhmetov/Topic-Aware-Sentiment-Analysis-of-News-Articles/; corpora: https://data.mendeley.com/datasets/m4ndy7tcss/2

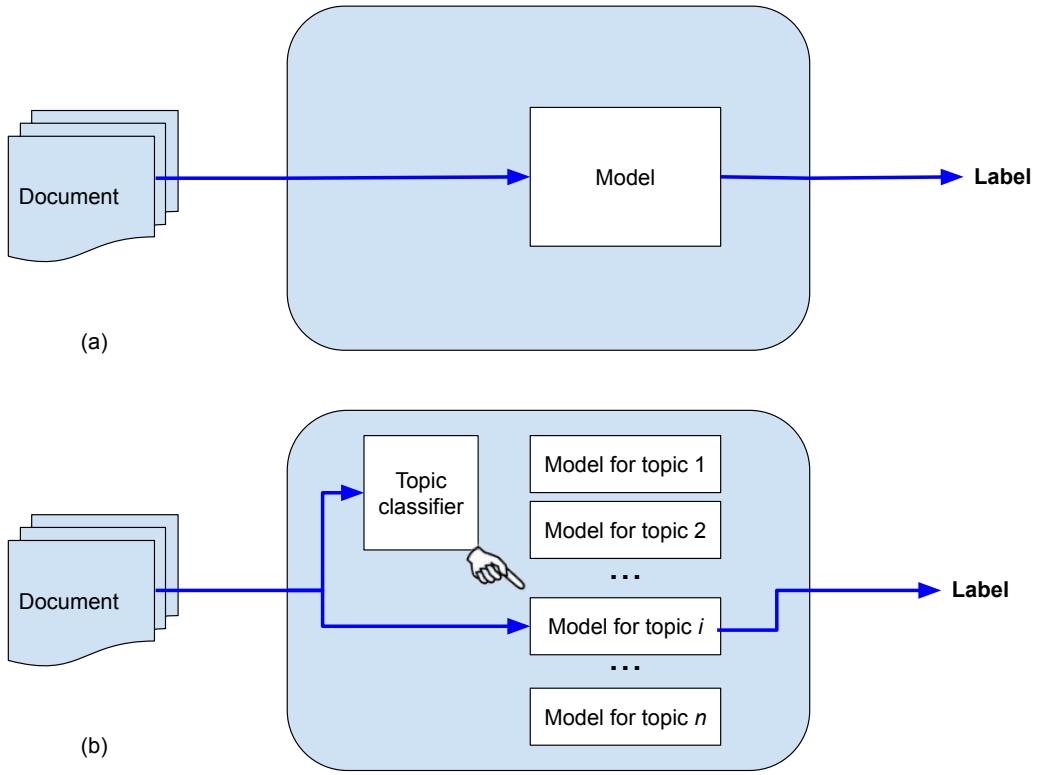

Fig. 1. (a) Conventional single-model classifier, (b) Instead, our classifier consists of a topic meta-classifier, which, for a given input document, selects one of several sentiment classifier models to be applied to derive the sentiment label for the document

sentiment consumer reviews and use it in business as inputs for technical support and quality control.

Series of works by different authors were devoted to the use of Neural Network architectures for Sentiment analysis: [22] describe modern methods for the sentiment analysis task of the news texts in Kazakh and Russian languages with the use of deep RNN.

Specifically, they used LSTM layers to consider long-term dependencies in the entire text, showing that good results can be achieved even without learning the peculiar linguistic features of a particular language.

Furthermore, [22] used Word2vec and GloVe vector embeddings as the main features in machine learning algorithms.

The word embeddings concept is the vector representation of words so that semantic interconnections between words preserved and basic linear algebra operations could be implemented [22].

The Aspect-Aware LSTM addresses the retention of aspect irrelevant text features [33]. After introducing the aspect context into the modeling process, the Aspect-aware LSTM can select helpful information about the target and discard the useless information through information flow control.

Recent development of the LSTM approach shown by [8] fights the feature space dimensionality expansion and introduces attention mechanism allowing to put more or less emphasis on certain words.

Another approach to deal with the aspect-level sentiment classification is applying syntactic edge-enhanced graph convolutional networks [32]. The fuzzy ontology text mining context and the use of word embedding for ontology-based topic modeling also suggest the use of the LSTM approach [7, 6].

Pazdnikova [25] implemented CNN in her work on aspect-oriented sentiment analysis for review summarization to show the soundness of her approach, which is to extract aspects of a text in the hierarchical form and use it as a feature.

Although the focus of current work is not on the aspect-level sentiment analysis, we cited the papers on the subject to give a broader view of current work problems.

Li et al. [20] proposed a framework for a fast, compact and optimized parameter conversational sentiment analysis using modern language models and embedding techniques: Bidirectional Emotional Recurrent Unit (BiERU). For this purpose in BiERU, a generalizing neural tensor block is followed by a double classifier to perform context and sentiment classification. This approach is also an example of applying a prior pseudo-topic classification before performing the sentiment analysis. [5] suggests a similar approach to monitor social network posts for healthcare purposes.

The summary of related work cited in this paper is shown in the Table 1.

3 Data

3.1 Topic Classification Corpus

For the topic classification model training, we used the corpus of 201,230 articles from Tengrinews.kz[2] collected by Institute of Information and Computational Technologies (IICT) laboratory personnel for the NLP project related to analyzing mass-media news articles' impact on society [2].

On average, texts contained 1,503.65 characters with a standard deviation of 1,001.96 characters, maximum of 39,973, and a minimum of 3 characters. Thus, 75% of the texts contained less than 1,745 characters.

Before topic classification, we had to clean the data from duplicate texts, zero-length texts, and texts in a language other than Russian. We also removed the stop-words and lemmatized the whole corpus texts. As a result, we detected 215 zero-length texts, which we excluded from the corpus.

In total, 6,122 texts in the corpus were in the Kazakh language partially, which was detected by the use of the Kazakh alphabet characters in the text, so they were also excluded from the corpus because we aimed to analyze the Russian language texts. We reserved mixed-language texts for future work.

3.2 Sentiment Analysis Data

3.2.1 Kaggle Corpus

We obtained sentiment-labeled dataset from Kaggle Data Science Competition website[3]. The training data consists of 8,263 records with three fields: identification number (id), sentiment (positive, negative, neutral), and the text itself. The sentiment label distribution is shown in Table 2. The texts had no topic labels; therefore, we applied the topic classification model to them.

3.2.2 KZ News

KZ News corpus consists of 5,211 news articles from various sources disclosed in Table 3.

The expert annotators, hired in the course of the project run by IICT on mass media news analysis of the impact on society, labeled the sentiment of the articles in the KZ news corpus by counting the number of words belonging to positive, negative and neutral words dictionaries and estimating the sentiment by the maximum number of words from the corresponding dictionary. The sentiment label distribution is shown in Table 2.

[2] Kazakhstani news aggregator founded in 2010. Part of Alash Media Group media-holding.

[3] https://www.kaggle.com/c/sentiment-analysis-in-russian/data

Table 1. Summary of related work

Paper	Main ideas
[19]	Importance and necessity of classification of texts by topics with high accuracy for better information extraction.
[34]	Mining the domain shared information for sentiment analysis.
[27]	Implementation domain-specific sentiment analysis of news streams in Russia
[31]	Method of topic-wise selection of negative sentiment consumer reviews and use it in business as inputs for technical support and quality control.
[22]	Modern methods for the sentiment analysis task of the news texts in Kazakh and Russian languages with the use of deep RNN.
[32]	Dealing with the aspect-level sentiment classification is applying syntactic edge-enhanced graph convolutional networks.
[8]	Fighting the feature space dimensionality expansion and introduces bi-directional layers allowing to put more or less emphasis on certain words
[6, 7]	LSTM approach in the fuzzy ontology text mining context, and in the use of word embedding for ontology-based topic modelling.
[25]	Implementation of CNN in her work on aspect-oriented sentiment analysis for review summarization.
[20]	A framework for a fast, compact and optimized parameter conversational sentiment analysis using modern language models and embedding techniques
[5]	Monitoring of social network posts for healthcare purposes.

Table 2. Kaggle and KZ News corpora sentiment label distribution

Sentiment	Kaggle	KZ News corpus
Negative	1,434	696
Neutral	4,034	4,120
Positive	2,795	395
Total	8,263	5,211

Table 3. KZ News corpus: composition by article source

Source	Number of articles
tengrinews.kz	1,889
camonitor.kz	619
zakon.kz	1,999
azatlyk.kz	381
kazakhstan.kz	323
Total	5,211

4 Methods

In this section, we describe our approach and the methods we used in our research: the Data preprocessing stage (lemmatization, stop-words removal, vectorization, data scaling), Classification (Convolutional Neural Networks, Decision Trees, Random Forest classifier, Extra-trees classifier, Multi-layer Perceptron classifier) and Evaluation (F1 score).

4.1 Topic-Aware Sentiment Analysis Approach Outline

Using the end-to-end text classification pipeline we have devised a step-by-step approach as follows:

1. Building topic classification model on the Tengrinews.kz corpus; see Section 3.1.
2. Classify the articles in Kaggle (Section 3.2.1) and KZ News (Section 3.2.2) corpora by topics.
3. Build sentiment classification models:
   (a) Topic-unaware model,
   (b) Topic-aware model.
4. Compare results if the topic-aware approach brings any improvements.

Table 4. Examples of the stop-words for the Russian language derived from the NLTK Python module. Underlined are the words for which the lemmatized form we used differs from the original NLTK stop-words

NLTK stop-words	Lemmatized	Meaning
бы	бы	would
всегда	всегда	always
него	он	him
ну	ну	well, hmm, ah
она	она	she
свою	свой	her, his
такой	такой	such
того	тот	that one
ты	ты	you
чтобы	чтобы	in order to

For building classification models, we use Convolutional Neural Networks and classic classification algorithms.

4.2 Data Preprocessing and Representation

4.2.1 Lemmatization

For lemmatization purposes, we considered the approach of Highly language-independent word lemmatization using Machine learning [3]. However, we have found that the approach shows descent results for the Russian language due to the morphological complexity of the language. Therefore we used the PyMorphy2 Python module [17] which employs the dictionary approach coupled with the morphological inference, and performs on the error level of 1% and even better than MyStem 3.0 by Yandex [29].

Lemmatization allowed us to reduce the text feature space by normalizing different words to their lemmas, preserving the word semantics in most cases.

After lemmatization, 232,181 unique words were left instead of the initial vocabulary of 1,425,789 words. This huge vocabulary reduction happened because of the morphological complexity of the Russian language (when lots of word forms derive from a single lemma), which is one of the properties of synthetic languages (Russian, Kazakh, Turkish, German), as opposed to the analytical languages (ex. English, Indonesian Bahasa).

4.2.2 Stop-Words Removal

Next, we removed the stop-words from the texts using a stop-words list. We derived our list from the Python NLTK module [30], which for the Russian language contains 151 stop-words.[4] However, since the input text was lemmatized, for our list, we also used lemmatized versions of the words from the NLTK list; see Table 4 for some examples. For this, we used the same PyMorphy2 module that we used for lemmatizing the input text.

In fact, the stop-word list from the Python NLTK module is quite incomplete; however, we used it for the simplicity and reproducibility of our results. Note that our list does not include negation words, so removing the stop-words does not alter the sentiment polarity. For other languages, care should be taken not to remove negation words.

[4]The complete list is: а ‘and’, без ‘without’, более ‘more’, больше ‘more’, будет ‘will be’, будто ‘as if’, была ‘was’, были ‘were’, было ‘It was’, был ‘was’, быть ‘be’, бы ‘would’, вам ‘you’, вас ‘you’, вдруг ‘suddenly’, ведь ‘after all’, вот ‘here it is’, во ‘in’, впрочем ‘however’, всегда ‘always’, всего ‘totally’, всех ‘all of them’, все ‘all’, всю ‘the whole’, вы ‘you’, в ‘in’, где ‘where’, даже ‘even’, да ‘yes’, два ‘two’, для ‘for’, до ‘before’, другой ‘other’, его ‘him’, ее ‘her’, ей ‘to her’, ему ‘him’, если ‘if’, есть ‘there is’, еще ‘more’, же ‘the same’, ж ‘indeed’, зачем ‘what for’, за ‘behind’, здесь ‘here’, из ‘out of’, или ‘or’, им ‘them’, иногда ‘sometimes’, их ‘them’, и ‘and’, какая ‘which’, какой ‘which’, как ‘as’, когда ‘when’, конечно ‘sure’, кто ‘who’, куда ‘where to’, к ‘to’, ли ‘whether’, лучше ‘better’, между ‘between’, меня ‘me’, мне ‘to me’, много ‘a lot of’, может ‘can’, можно ‘might’, мой ‘my’, моя ‘my’, мы ‘we’, на ‘on’, надо ‘need’, над ‘over’, наконец ‘finally’, нас ‘us’, него ‘him’, нее ‘her’, ней ‘her’, нельзя ‘can’t’, нет ‘no’, не ‘not’, нибудь ‘some’, никогда ‘never’, ним ‘him’, них ‘them’, ничего ‘nothing’, ни ‘nor’, но ‘but’, ну ‘well’, об ‘about’, один ‘one’, она ‘she’, они ‘they’, он ‘he’, опять ‘again’, от ‘from’, о ‘about’, перед ‘front’, под ‘under’, после ‘after’, потому ‘because’, потом ‘after then’, почти ‘nearly’, по ‘by’, при ‘at’, про ‘about’, разве ‘wheather’, раз ‘time’, сам ‘myself’, свою ‘one’s own’, себе ‘myself’, себя ‘myself’, сейчас ‘now’, совсем ‘quite’, со ‘with’, с ‘with’, такой ‘such’, так ‘so’, там ‘there’, тебя ‘you’, тем ‘themes’, теперь ‘now’, тогда ‘then’, того ‘that one’, тоже ‘also’, только ‘only’, том ‘about that’, тот ‘that one’, то ‘then’, три ‘three’, тут ‘here’, ты ‘you’, уже ‘already’, уж ‘already’, у ‘at’, хорошо ‘good’, хоть ‘though’, чего ‘what’, чем ‘than’, через ‘through’, чтобы ‘so that’, чтоб ‘so that’, что ‘what’, чуть ‘slightly’, эти ‘these’, этого ‘of this’, этой ‘this’, этом ‘this’, этот ‘this’, эту ‘this’, я ‘I’.

4.2.3 Vectorization

The Bag of Words representation The Bag of Words (BoW) models [15] allow representing a text as a feature vector of unique token statistics (counts, term frequencies, and many others) irrespective of the word order in the text. We vectorized the texts using the `CountVectorizer` from the `sklearn` Python module, which converts a collection of texts to a token count matrix.

The Python module `sklearn` provides tools to extract numerical feature vectors from a bunch of texts: (1) Strings tokenizing and assigning an integer identification number for each unique token used, (2) Counting the tokens occurrences in each text, (3) Weighting and normalizing with diminishing importance of tokens that are present in the majority of texts.

Individual tokens occurrence frequency is considered a feature vector representing all the tokens for a given text.

A text dataset can be converted to a matrix with one row for each text and one feature column for each of the tokens present in the corpus's vocabulary.

The vectorization method was used for classical machine learning model training.

Sequence Vectorization and Word Embeddings The sequence vectorization was performed, representing each text with a sequence of integers, where each integer is the index of a unique token in a dictionary. The obtained integer sequences formed vectors of length 1,000, which is the sequence of the first 1,000 words' identification numbers in a text.

Subsequently, we substitute the token indexes by the word embeddings of length 300, which store the word semantics derived from the word co-occurrence information.

The vectorization method was used for Convolutional Neural Network (CNN) model training.

4.2.4 Scaling

The BoW vectors were scaled to be comparable by subtracting the mean and scaling to unit standard deviation.

The standard score of the sample x is calculated as follows:

$$Z = \frac{x - \mu}{s}, \tag{1}$$

where μ is the training samples mean, and s is the training samples variance (standard deviation).

Scaling a dataset is generally recommended for ML algorithms because they could bring in the bias from absolute value domination of some features over others.

4.3 Classification

4.3.1 Convolutional Neural Networks (CNN)

CNN are neural networks with non-cyclic layer links using multilayer perceptrons to reduce data preprocessing workloads. This model was inspired by the research of human visual cortex function research, finding their application mainly in the computer vision tasks [18]. However, many recent works show that CNN also shows promising results in the area of NLP.

Each convolution result will appear when the algorithm finds a sought pattern. By changing the shape of the kernels and putting their results together, the network can detect patterns of different sizes. For example, text patterns could consist of expressions (word n-grams), and therefore CNNs [9] can identify them in the texts regardless of their position in text [21].

4.3.2 Classic Classification Models

We chose to work with Multi-Layer Perceptron, Decision Tree algorithm and its variations for our classification task because they perform faster than K-neighbors or Support Vector Machine algorithms and provide better interpretability intuition than Naive Bayes classifiers.

Decision Trees *Decision Trees (DTs)* [26] are a non-parametric supervised learning algorithm applicable to both classification and regression tasks. The method itself is as ancient as our civilization or even older [23]. The algorithm aims to build a model that forecasts the target value by learning simple decision tree rules derived from the independent variables. The deeper the tree, the more complex the decision tree rules, and it may overfit the data, which is the biggest problem with this algorithm. The Decision Tree algorithm also shows some potential to work with highly imbalanced data, which is the case for sentiment labeled data where the people attitude has no even distribution [24].

Random forest classifier In Random Forests (RF) [16], each of the ensemble trees is built on a sample randomly drawn with replacement from the training dataset. As a result of randomness, forest bias usually increases. However, due to averaging, its variance decreases, generally more than compensate for the increase in bias, thus generating a better model generally [26].

Extra-trees classifier The method implements a meta-estimator that trains several randomized decision trees on various dataset samples and then averages the results. Similar to the Random Forest algorithm, it uses a random feature subset. However, instead of looking for the most discriminative thresholds, the algorithm selects them randomly, scores them, and uses the best instances as the splitting rule. As a result, it generally reduces the model variance slightly more than the Random Forest algorithm while increasing the bias [26].

Multi-layer Perceptron classifier (MLP) MLP optimizes the log-loss function using stochastic gradient descent. The Python class `MLPClassifier` uses iterative training because, on each step, only the partial derivatives of the loss function can be computed for parameters update. To deal with the overfitting problem, one could add regularization parameters [26]; we did not do it for simplicity. Instead, we used the default configuration of each classifier.

4.4 Evaluation

F1 score For imbalanced datasets the accuracy metric is not a proper measure to use as it can be maximized simply classifying all the items for the majority class. Therefore other quality metrics should be used accounting for the issue, such as F1 score. It can be defined as the harmonic mean of Precision and Recall [11, 28]:

$$\text{F1 score} = \frac{2PR}{P+R}, \tag{2}$$

where P is the Precision and R is the Recall, see Equation 3 and Equation 4 respectively.

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives +false positives}}, \tag{3}$$

where $false\ positives$ is the number of Type I errors when not a target class is classified as positive.

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives +false negatives}}, \tag{4}$$

where $false\ \ negatives$ is the number of Type II errors when a target class is misclassified as negative.

4.5 Baseline Model

As a baseline model we used a majority class prediction proxy, meaning classifying all the articles as if they were from the dominant sentiment class. Hence, calculating respective F1 values to compare with the models we experiment with.

5 Experiments

Our methodology consists of the following parts:

1. Developing a topic classification model
2. Developing and testing a topic-wise sentiment analysis model

[5] WW2 chronicle
[6] WW2 stories
[7] WW2 facts
[8] Elections
[9] Exhibitions
[10] Explosions

Table 5. Generalizing on parser generated categories

Parser generated category stack	Generalized category
story, mixnews, others, other, strange_news, autos, style, kids, life, social, multispace, discounts, tourism, travel, around-the-world, travel-notes, vov_chronicle[5], vov_stories[6], vov_facts[7]	mix
euro-2016, local_sport, sports, boxing, hockey, fifa2018, world_sport, sport, cycle_racing, promises_sport, tennis, football, allsports	sport
my-country, kazakhstan_news, astana_almaty_2011, bi-group, politic, vibori[8], news	kz
near_east, world_news, osce, asia, europe, usa, latin	world
tech, gadgets, internet, progress, science, medicine	hitech
russia, sng	sng
ipo, private_finance, money, markets, buildings, economic	fin
oscar2019, events, theaters, cinema, clubs, show, vistavki[9], expo, music, picture_art, books, festival	leis
accidents, crime, incidents, vvz[10]	accid

Table 6. Initial Topic Distribution and Distribution after Applying SMOTE

Topics	Initial topic distribution	Post SMOTE
kz	73,514	49,439
fin	11,900	49,439
accid	17,763	49,439
hitech	10,149	49,439
leis	26,081	49,439
world	12,948	49,439
sport	18,555	49,439
sng	9,382	49,439
mix	14,601	49,439
Total	194,893	444,951

5.1 Topic Classification

5.1.1 Existing Topic Category Labels

We labeled the texts obtained from Tengrinews.kz by category, obtained by parsing the article URL. So we had our labels to train the topic classifier.

Due to the excessive number of topics obtained from URL parsing, we decided to group them and leave just nine topics (Table 5).

Balancing Topic Distribution As Table 6 shows, the topic distribution is highly imbalanced in favor of KZ news (73,514), followed by a considerable lag by leisure articles (26,081), sports news (18,555), and accidents (17,763).

In order to balance the topic distribution, we used the Synthetic Minority Over-sampling Technique (SMOTE), which oversampled the minority class by introducing synthetic examples of k-nearest neighbors for each data point in the minority class sample [10]. Then, depending on the amount of oversampling needed, it chooses K-nearest neighbor elements.

Shown in Table 6 the SMOTE under-sampled the primary class from 73,514 to 49,439 items and over-sampled the other classes to the same number of items. The method is said to bring optimal results in the original paper [10].

5.1.2 Classic Topic Classification Models for Topic Classification

We ran the cross-validation (CV) on three folds to select the topic classification model. CV results in the Table 7 suggest that the Extra Tree classifier model is the best algorithm.

After training the ExtraTreeClassifier model, which showed the best CV result of 0.92 on accuracy score, it performed poorly with a 0.62 accuracy score. Other models behaved similarly on the test data (Table 7).

Table 7. Cross-validation (CV) model selection for the topic classification task (average accuracy score) and subsequent test results

	CV	Test
DecisionTreeClassifier	0.81	0.56
ExtraTreesClassifier	**0.92**	**0.62**
RandomForestClassifier	0.13	0.08
MLPClassifier	0.85	0.60

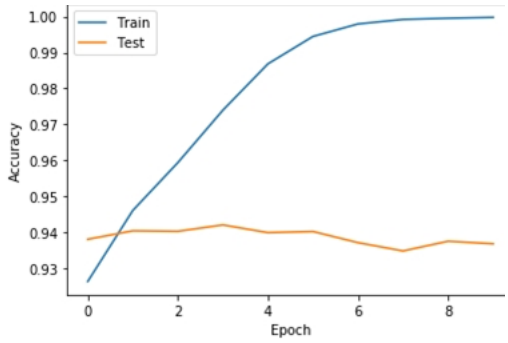

Fig. 2. Training history of CNN topic classification model

5.1.3 Convolutional Neural Network (CNN) for Topic Classification

The CNN built according to the design by [14] resulted in a 0.93 accuracy score (Fig. 2) on the test data with a 95% confidence interval of 0.9345–0.9393, and we decided to use this model in further experiments. The model summary is presented in Table 8.

5.1.4 Classify Experimental Sentiment-Labeled Datasets by Topics:

The sentiment-labeled datasets were classified by topics and associated topic weights are shown in the Table 9.

Kaggle corpus topics We obtained a labeled sentiment analysis dataset consisting of news articles in the Russian language from the Kaggle website. The training data consists of 8,263 records with three fields: identification number, sentiment (positive, negative, neutral), and the text itself. The texts in the Kaggle dataset had no topic labels, so we applied the topic classification model to them (Table 10).

KZ News Corpus Topics KZ News corpus consists of 5,211 news articles from various sources: Tengrinews.kz (1,889), camonitor.kz (619), zakon.kz (1,999), azatlyk.kz (381), Kazakhstan.kz (323). Experts hired in the course of the project run by IICT on mass media news analysis of the impact on the society labeled the articles by sentiment (Table 10).

5.2 Sentiment Analysis

As mentioned before, this work's final objective was to determine if the domain-specific sentiment analysis can bring any better F1 scores than trying to model on a document set of diverse topics.

5.2.1 Topic-Wise Sentiment Analysis

Some words and expressions can have different meanings when used in different domain contexts. For example, the exact words used by sports commentators and stock exchange traders or medical doctors can have drastically different meanings because of the usage context. The same thing is for movie and home appliances reviews made by internet users or news articles about finance or accidents. Therefore we first defined the topic of the text and then applied the sentiment analysis model that we trained for that specific topic.

5.2.2 CNN Sentiment Classification Model

For sentiment analysis, we applied a CNN with the architecture shown in Table 8 to both the Kaggle and KZ News corpus. For the majority class baseline model F1 scores, see Table 11 and for per-class F1 scores, see Table 12.

Kaggle Corpus CNN Results The sentiment analysis results on the Kaggle corpus are in the Table 12. The topic subset of the corpus, classified as "kz" (containing 5,606 articles; see Table 9 for topic weights for Kaggle and KZ News corpus) when classified for sentiment, gave an F1 score of 0.70 on the test set exceeding the baseline of 0.67. Minor topics "sng" and "sports" containing 75 and 158 articles, respectively, also exceeded the Baseline model showing 0.83 and 0.68 F1 score.

Table 8. CNN Topic classification model summary.

Layer	Type	Output Shape	Number of parameters
`embedding_3`	Embedding	1000×300	15,000,300
`conv1d_3`	Conv1D	996×128	192,128
`global_max_pooling1d_3`	GlobalMaxPooling1D	128×1	0
`dense_20`	Dense	128×1	16,512
`dense_21`	Dense	9×1	1,161
Total params:			15,210,101
Trainable params:			15,210,101
Non-trainable params:			0

Table 9. Corpus topic weights

	Kaggle		KZ News	
	Articles	Weight	Articles	Weight
kz	**5,506**	**0.67**	**2,865**	**0.55**
fin	**1,751**	**0.21**	285	0.05
accid	245	0.03	302	0.06
hitech	91	0.01	141	0.03
leis	70	0.01	460	0.09
world	40	0.00	259	0.05
sport	158	0.02	431	0.08
sng	75	0.01	189	0.04
mix	327	0.04	279	0.05
Total	8,263	1.00	5,211	1.00

KZ News Corpus CNN Results The F1 scores in Table 12 tell us a different story. We surpass the topic-unaware sentiment analysis for all the topics but only for the *neutral* class, and for the *negative* and *positive* classes, our model fails, having an F1 score of 0.00. We can attribute this to the fact that KZ News corpus is heavily imbalanced, see Table 2, and biased towards the *neutral* class. Therefore CNN model failed to learn from the data.

5.2.3 Classic Classification Models for Sentiment Analysis

We also experimented with classic models for topic-aware sentiment analysis on the two labeled corpora. For per class F1 score please see Table 13.

Kaggle Corpus Classic Models Results Model selection on Kaggle corpus using cross-validation on three folds advised us to use the Random Forest model (Table 14). The F1 scores for Random Forest model on Kaggle dataset show that topic-aware generally passes the Baseline making it 0.34 over 0.32 on F1 score. The topic-aware model also surpasses the Baseline on such topics as “mix”, “sng”, “sport”, “leis”, “hitech”, “accid”, and “fin”, see Table 13.

KZ News Corpus Classic Models Results We also selected the Random Forest classifier for sentiment analysis of the KZ news dataset. The cross-validation scores are higher here than the same thing for the Kaggle corpus, but still, F1 scores show that the results are biased by the dominance of the *neutral* sentiment class in the corpus. It is worth noting that cross-validating `KNeighborsClassifier`, `RandomForestClassifier`, and `SVC` (SVM Classifier) models gave surprisingly identical accuracy of 0.79. One possible explanation is rounding conversion. Still, very close results obtained by completely different algorithms are subject to research as currently, we do not have a good explanation for this fact.

Training the Random Forest model for topic-aware sentiment analysis gave us results consistent with those achieved on training the CNN. Topic-aware model surpasses the Baseline on “mix”, “sport”, “world”, “leis”, “hitech”, “accid” and “kz” topics on F1 scores, see Table 13.

6 Discussion

The major limitation of the present work is the amount of publicly available sentiment-labeled texts. The Data section suggests that the topic-labeled corpus consisted of 194,893 articles,

Table 10. Kaggle sentiment labeled corpus topics classification results

	Kaggle corpus			KZ News corpus		
Topic	Articles	Majority sentiment class	Majority sentiment class (%)	Articles	Majority sentiment class	Majority sentiment class (%)
world	40	neu	50%	259	neu	79%
leis	70	neu	57%	460	neu	84%
sng	75	neu	80%	189	neu	61%
hitech	91	pos	68%	141	neu	90%
sport	158	pos	47%	431	neu	83%
accid	245	neg	45%	302	neu	85%
mix	327	pos	45%	279	neu	80%
fin	1,751	neu	56%	285	neu	74%
kz	5,506	neu	49%	2,865	neu	82%
Total	8,263	neu	48%	5,211	neu	77%

Table 11. Majority class baseline model F1 score on test dataset (the numbers in bold indicate results exceeding the baseline score). WA stands for weighted average of F1 score

	Kaggle				KZ News			
	neg	neu	pos	WA	neg	neu	pos	WA
Baseline: Fig. 1(a)	0.00	0.65	0.00	0.31	0.00	0.87	0.00	0.66
Topic-aware: Fig. 1(b)	0.02	0.61	0.05	0.31	0.00	0.87	0.00	0.66
world	0.00	0.67	0.00	**0.33**	0.00	0.88	0.00	**0.70**
leis	0.00	0.73	0.00	**0.42**	0.00	0.91	0.00	**0.76**
sng	0.00	0.89	0.00	**0.71**	0.00	0.75	0.00	0.46
hitech	0.00	0.00	0.81	**0.56**	0.00	0.95	0.00	**0.85**
sport	0.00	0.00	0.64	0.30	0.00	0.91	0.00	**0.75**
accid	0.62	0.00	0.00	0.28	0.00	0.92	0.00	**0.78**
mix	0.00	0.00	0.63	0.28	0.00	0.89	0.00	**0.72**
fin	0.00	0.71	0.00	**0.40**	0.00	0.85	0.00	0.63
kz	0.00	0.66	0.00	0.32	0.00	0.90	0.00	**0.74**

Table 12. CNN model per topic F1 score on test dataset (the numbers in bold indicate results exceeding the baseline score). WA stands for weighted average of F1 score

	Kaggle				KZ News			
	neg	neu	pos	WA	neg	neu	pos	WA
Baseline: Fig. 1(a)	0.60	0.71	0.67	0.67	0.10	0.86	0.02	0.68
Topic-aware: Fig. 1(b)	0.64	0.71	0.62	0.67	0.00	0.90	0.00	**0.69**
world	0.67	0.73	0.00	0.53	0.00	0.88	0.00	**0.70**
leis	0.57	0.67	0.00	0.54	0.00	0.90	0.00	**0.76**
sng	0.67	0.92	0.00	**0.83**	0.00	0.75	0.00	0.46
hitech	0.00	0.00	0.81	0.56	0.00	0.95	0.00	**0.85**
sport	0.60	0.67	0.72	**0.68**	0.00	0.91	0.00	**0.75**
accid	0.68	0.52	0.00	0.52	0.00	0.92	0.00	**0.78**
mix	0.22	0.50	0.60	0.52	0.00	0.89	0.00	**0.72**
fin	0.51	0.74	0.46	0.62	0.00	0.85	0.00	0.63
kz	0.69	0.72	0.68	**0.70**	0.00	0.90	0.00	**0.74**

Table 13. Random Forest model per topic F1 score on test dataset (the numbers in bold indicate results exceeding the baseline score). WA stands for weighted average of F1 score

	Kaggle				KZ News			
	neg	neu	pos	WA	neg	neu	pos	WA
Baseline: Fig. 1(a)	0.01	0.66	0.00	0.32	0.00	0.87	0.00	0.67
Topic-aware: Fig. 1(b)	0.07	0.62	0.08	0.34	0.02	0.87	0.00	0.67
mix	0.00	0.32	0.55	**0.39**	0.00	0.86	0.00	**0.69**
sng	0.00	0.92	0.00	**0.74**	0.15	0.79	0.00	0.52
sport	0.00	0.38	0.68	**0.44**	0.00	0.86	0.00	**0.71**
world	0.00	0.50	0.00	0.25	0.32	0.83	0.00	**0.72**
leis	0.25	0.56	0.00	**0.39**	0.00	0.88	0.00	**0.74**
hitech	0.00	0.25	0.76	**0.58**	0.00	0.98	0.00	**0.88**
accid	0.73	0.32	0.00	**0.46**	0.00	0.90	0.00	**0.77**
fin	0.00	0.70	0.04	**0.40**	0.00	0.82	0.00	0.61
kz	0.01	0.63	0.00	0.31	0.00	0.88	0.00	**0.72**

Table 14. Cross-validation for Kaggle and KZ News corpora sentiment analysis (accuracy)

Model	Kaggle	KZ News
LinearSVC	0.34	0.64
SGDClassifier	0.39	0.68
MultinomialNB	0.34	0.62
KNeighborsClassifier	0.44	**0.79**
RandomForestClassifier	**0.49**	**0.79**
DecisionTreeClassifier	0.37	0.63
ExtraTreesClassifier	0.42	0.76
MLPClassifier	0.37	0.65
SVC (SVM Classifier)	0.46	**0.79**

whereas corpora with sentiment-labeled texts contained 8,263 and 5,211 articles, respectively.

After the articles in the Kaggle corpus and the KZ News corpus were classified by topics, it came out that there were 1 or 2 essential topics and the rest were not represented sufficiently, and the corpora were imbalanced by topics. Therefore, we need more sentiment-labeled data for the proper model training process. Hence, the results obtained by CNN for sentiment analysis may differ if we apply them to a larger corpus. Thus, the question is the availability of a sentiment-labeled corpus of news article texts in Russian of a size suitable for proper neural network training.

There is also a question about the labeling quality for the Kaggle corpus. The Kaggle corpus also contains a small number of large texts that are not news article texts but some wordy law projects and may negatively affect the quality. As for the KZ News corpus, we have a concern of a heavy dominance of the *neutral* class in the dataset.

Finally, the two sentiment-labeled datasets are highly imbalanced, and there is a need for a more extensive corpus or application of the SMOTE method, which we used with the data used for the TC model.

As can be seen from the Fig. 3, the two datasets have similar amounts of articles with neutral sentiment but drastically different amounts of positive and negative sentiment article amount, especially considering that the Kaggle dataset is more than 50% larger than the KZ News dataset.

7 Conclusion

Obtained results for topic-aware sentiment analysis on Kazakhstani news corpora partially prove the hypothesis that domain specificity can help achieve greater F1 scores.

As we showed in Table 11 the weighted average F1 scores for Majority class baseline model Topic-unaware and Topic-aware models are 0.31238/0.31183 and 0.66458/0.66458 for the Kaggle and the KZ News corpora respectively. Thus the Topic-aware approach shows slightly worse results for this particular model. But the CNN and RF models outperform the baseline model yielding 0.67497/0.66718 and 0.32/0.34 for Topic-unaware/Topic-aware models on the Kaggle

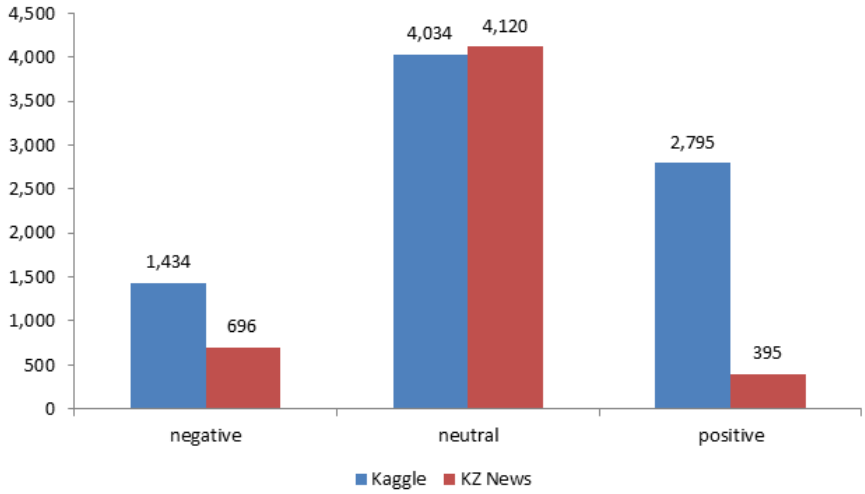

Fig. 3. Relative sentiment labels distribution among datasets

corpus, and 0.68/0.69 with 0.67/0.67 on the KZ News corpus, see Table 12 and Table 13. The gain of Topic-aware over Topic-unaware models is not great because of the topic and sentiment imbalances in the corpora leading to the data sparseness where we actually work with only "neutral" sentiment and "kz" and "fin" topics to train our models.

The reason why the CNN model outperformed Random Forest (RF) classifier is due to the following reasons:

— CNN model does not require complex feature engineering and derives the features from the trained parameters on its own. At the same time, RF depends on the features provided before the training process.

— CNN models can scale effectively with the data size. Although the classical models seem to work better than Neural Network models on small data, we can see that is not always the case. So we suppose that not overly complex CNN architecture fitted just right for our sentiment analysis task with a relatively small amount of data.

The neural network design used in this work and the general approach can find applications in various social and commercial applications:

1. Monitoring sentiments of specific topics expressed by mass media and social network posts.

2. Detecting negative aspects (topics) of business or public policy or administration when discussed in social networks and mass media to tackle the problems promptly.

3. Stratify the society according to the attitude towards specific topics by detecting sentiment analysis in the comments to the articles in mass media and posts in social networks.

4. Profiling the mass media sources and authors according to the topic-sentiment blend.

For future work, we plan to implement the following enhancements:

1. Experiment with the oversampled topic classes for both corpora (Kaggle and KZ news especially) to balance different topics in the mass media articles.

2. Develop an approach for sentiment analysis of mixed language (ex. Russian, English and Kazakh) news articles.

3. Test our approach on a larger corpus labeled for sentiment, which we could obtain by an expert or crowd-sourced labeling, or even using some unsupervised machine learning algorithms.
4. Try the approach for different languages to build a language-independent model that could find numerous applications worldwide.
5. Address the problem of handling articles belonging to more than one topic because some articles can have relations to several topics with a certain probability for each. So a probabilistic topic modeling could be employed here.
6. Replace Topic classifier used in current work with an Unsupervised Topic Model using K-means clustering and Variable Neighborhood optimization heuristics [1].
7. Use state-of-the-art vectorization methods such as BERT, ELMO, Electra, and GPT-2/3. The contextualized embeddings bring better results in text classification tasks.
8. Research the dependency of the corpora size with the quality of the topic-aware sentiment analysis model to define the corpora size we need for proper modeling.
9. Use multilingual/Russian pre-trained language models and fine tune them for Topic-aware sentiment analysis task, which should reduce the quantity of sentiment-labeled data required to train the model.

Acknowledgment

This research is conducted within the framework of the grant num. AP09058174 “Development of language-independent unsupervised methods of semantic analysis of large amounts of text data”.

The work was done with the support from the Mexican Government through the grant A1-S-47854 of the CONACYT, Mexico and grants 20211784, 20211884, and 20211178 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico.

There was no additional external funding received for this study.

References

1. **Akhmetov, I., Mladenovic, N., Mussabayev, R. (2021).** Using k-means and variable neighborhood search for automatic summarization of scientific articles. **Mladenovic, N., Sleptchenko, A., Sifaleras, A., Omar, M.**, editors, Variable Neighborhood Search, Springer International Publishing, Cham, pp. 166–175.
2. **Akhmetov, I., Mussabayev, R., Gelbukh, A. (2021).** Topic-aware sentiment analysis. https://data.mendeley.com/datasets/m4ndy7tcss/2. DOI: 10.17632/M4NDY7TCSS.2.
3. **Akhmetov, I., Pak, A., Ualiyeva, I., Gelbukh, A. (2020).** Highly language-independent word lemmatization using a machine-learning classifier. Computacion y Sistemas, Vol. 24, No. 3. DOI: 10.13053/CYS-24-3-3775.
4. **Ali, F., Ali, A., Imran, M., Naqvi, R. A., Siddiqi, M. H., Kwak, K.-S. (2021).** Traffic accident detection and condition analysis based on social networking data. Accident Analysis and Prevention, Vol. 151, pp. 105973. DOI: https://doi.org/10.1016/j.aap.2021.105973.
5. **Ali, F., El-Sappagh, S., Islam, S. M., Ali, A., Attique, M., Imran, M., Kwak, K. S. (2020).** An intelligent healthcare monitoring framework using wearable sensors and social networking data. Future Generation Computer Systems, Vol. 114. DOI: 10.1016/j.future.2020.07.047.
6. **Ali, F., El-Sappagh, S., Kwak, D. (2019).** Fuzzy Ontology and LSTM-based Text Mining: A Transportation Network Monitoring System for Assisting Travel. Sensors, Vol. 19, No. 2, pp. 234. DOI: 10.3390/s19020234.
7. **Ali, F., Kwak, D., Khan, P., El-Sappagh, S., Ali, A., Ullah, S., Kim, K. H., Kwak, K. S. (2019).** Transportation sentiment analysis using word embedding and ontology-based topic modeling. Knowledge-Based Systems, Vol. 174. DOI: 10.1016/j.knosys.2019.02.033.

8. **Basiri, M. E., Nemati, S., Abdar, M., Cambria, E., Acharya, U. R. (2021).** ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis. Future Generation Computer Systems, Vol. 115, pp. 279–294. DOI: https://doi.org/10.1016/j.future.2020.08.005.
9. **Britz, D. (2015).** Understanding Convolutional Neural Networks for NLP – WildML.
10. **Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W. (2002).** SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR), Vol. 16, pp. 321–357. DOI: 10.1613/jair.953.
11. **Chinchor, N. (1992).** Muc-4 evaluation metrics. Proceedings of the 4th Conference on Message Understanding, MUC4 '92, Association for Computational Linguistics, USA, pp. 22–29. DOI: 10.3115/1072064.1072067.
12. **Desai, A., Jhaveri, R. (2018).** The role of machine learning in internet-of-things (IoT) research: A review. International Journal of Computer Applications, Vol. 179, pp. 36–44.
13. **Hakak, S., Alazab, M., Khan, S., Gadekallu, T., Maddikunta, P., Khan, W. (2021).** An ensemble machine learning approach through effective feature extraction to classify fake news. Future Generation Computer Systems: the international journal of grid computing: theory, methods and applications, Vol. 117, pp. 47–58. DOI: 10.1016/j.future.2020.11.022.
14. **Hamel, L. (2019).** Deep learning finds fake news with 97% accuracy.
15. **Harris, Z. S. (1954).** Distributional structure. WORD, Vol. 10, No. 2-3, pp. 146–162. DOI: 10.1080/00437956.1954.11659520.
16. **Ho, T. K. (1995).** Random decision forests. Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, pp. 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.
17. **Korobov, M. (2015).** Morphological analyzer and generator for Russian and Ukrainian languages. CoRR, Vol. abs/1503.07283.
18. **LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. (1989).** Backpropagation applied to handwritten zip code recognition. Neural Comput., Vol. 1, No. 4, pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.
19. **Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A., Choudhary, A. (2011).** Twitter trending topic classification. 2011 IEEE 11th International Conference on Data Mining Workshops. DOI: 10.1109/icdmw.2011.171.
20. **Li, W., Shao, W., Ji, S., Cambria, E. (2021).** Bieru: Bidirectional emotional recurrent unit for conversational sentiment analysis.
21. **Maheshwari, A. (2018).** Report on text classification using CNN, RNN & HAN. https://medium.com/jatana.
22. **Narynov, S., Zharmagambetov, A. (2016).** On one approach of solving sentiment analysis task for Kazakh and Russian languages using deep learning. Computational Collective Intelligence Lecture Notes in Computer Science, pp. 537–545. DOI: 10.1007/978-3-319-45246-3_51.
23. **Norman, J. (2019).** The Scala Praedicamentalis or Porphyrian Tree, the earliest metaphorical tree of knowledge.
24. **Panigrahi, R., Borah, S., Bhoi, A. K., Ijaz, M. F., Pramanik, M., Kumar, Y., Jhaveri, R. H. (2021).** A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets. Mathematics, Vol. 9, No. 7. DOI: 10.3390/math9070751.
25. **Pazdnikova, M. (2017).** Автоматическе реферирование отзывов на снове аспектно-ориентированного тонального анализа (in Russian) [Automatic review summarization based on the aspect oriented sentiment analysis]. MSc thesis. St Petersburg University.
26. **Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011).** Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, Vol. 12, pp. 2825–2830.
27. **Sadekova, T. (2018).** Выделение мнений в тематичесих моделях новостных потоков (in Russian) [Opinion mining in topic models of news streams]. MSc thesis. Lomonosov Moscow State University. MachineLearning.ru.
28. **Sasaki, Y. (2007).** The truth of the F-measure. Teach Tutor mater.
29. **Segalovich, I. (2003).** A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine. **Arabnia, H. R., Kozerenko, E. B.**, editors, Proceedings of

the International Conference on Machine Learning; Models, Technologies and Applications. MLMTA'03, June 23 - 26, 2003, Las Vegas, Nevada, USA, CSREA Press, pp. 273–280.

30. **Steven Bird, E. L., Ewan Klein (2009).** Natural Language Processing with Python. O'Reilly Media.
31. **Tutubalina, E. (2016).** Методы извлечения и резюмирования критических отзывов пользоателей о продукции (in Russian) [Methods of extraction and summarization of critical consumer reviews]. MSc thesis. Kazan Federal University.
32. **Xiao, Y., Zhou, G. (2020).** Syntactic edge-enhanced graph convolutional networks for aspect-level sentiment classification with interactive attention. IEEE Access, Vol. 8, pp. 157068–157080. DOI: 10.1109/ACCESS.2020.3019277.
33. **Xing, B., Liao, L., Song, D., Wang, J., Zhang, F., Wang, Z., Huang, H. (2019).** Earlier attention? Aspect-aware LSTM for Aspect Sentiment Analysis. CoRR, Vol. abs/1905.07719.
34. **Zhou, G.-Y., Huang, J. X. (2017).** Modeling and mining domain shared knowledge for sentiment analysis. ACM Trans. Inf. Syst., Vol. 36, No. 2. DOI: 10.1145/3091995.

Article received on 29/07/2021; accepted on 30/11/2021.
Corresponding author is Alexander Gelbukh.