A Machine Learning Model for Automatic Emotion Detection from Speech

Nataliia Kholodna, Victoria Vysotska, Solomiia Albota

Lviv Polytechnic National University, S. Bandera street, 12, Lviv, 79013, Ukraine

Abstract

The paper aims to create a machine-learning model for automatic emotion detection from speech. The developed model is to be used in the system of monitoring public emotions. A short analysis of other research papers on the process of designing machine-learning models for automatic emotion detection from speech has been provided in the article. Classical and deep machine learning methods and algorithms and some features of the initial dataset have been considered. The DailyDialog and its applicability for training the classificatory have been regarded. Moreover, developing and selecting the optimal model for automatic emotion detection from speech has been described. The research results on the influence of factors such as the number of records of each category in the training data set, text pre-processing, methods of vectorisation or word embedding, the choice of machine learning method for text classification, parameters and architecture of the model have been given. The examples of using the machine-learning model to analyse the collected real-world data have been demonstrated in the last section. The process of correlation of the change in the number of records that belong to different emotional categories with specific events in the life of society, the population of a geographical region or a community has been shown. Finally, the limitations of the created machine-learning model and some possible steps to refine the system for detecting emotions have been considered.

Keywords [1]

Machine learning, emotion detection, deep learning, machine learning model, text pre-processing, neural network, learning model, deep learning model, word embedding, created machine learning model, logistic regression, classical machine learning method, social network, temporary convolutional neural network, social network Twitter, naive Bayesian classifier, pre-processing, information resource, machine learning method, convolutional neural network, classification method, learning method, vectorisation method, text classification, stop word, artificial neural network, adaptive boosting algorithm

1. Introduction

Emotions play an essential role in our daily lives and affect our social interaction, behaviour, relationships with other people, and even how we make decisions that are important to us.

Text is an essential source for identifying emotions. Such content may contain information about the person's psychological state and reflect the feelings experienced by society at a particular time. Analysis of text that is rich in emotions can be used in many areas, such as:

- Early warning systems for government, health or emergency services;
- To assess the individual psychological state of a person by their activity in the social network;
- A different approach for making critical business decisions by analysing feedback on a product or service;
- Analysis of news and technical literature.

MoMLeT+DS 2021: 3rd International Workshop on Modern Machine Learning Technologies and Data Science, June 5, 2021, Lviv-Shatsk, Ukraine

EMAIL nataliia.kholodna.sa.2018@lpnu.ua (N. Kholodna); Victoria.A.Vysotska@lpnu.ua (V. Vysotska); Solomiia.M.Albota@lpnu.ua (S. Albota)
ORCID: 0000-0002-6908-2900 (N. Kholodna); 0000-0001-6417-3689 (V. Vysotska); 0000-0002-0845-5304 (S. Albota)

The research object is the problem of creating an adequate model of emotion recognition in the text and the study of the influence of various factors and parameters on the quality of classification.

The work aims to create a machine-learning model for automatic emotion detection from text. The developed model can be further integrated into the system of monitoring the emotional state of society, the population of a geographical region or a community.

2. Related works

Maryam Hasan, Elke Rundensteiner, Emmanuel Agu [1] developed a system of automatic emotion detection in the flow of text publications on the social network Twitter – EmotexStream. Experiments show that the model correctly classifies 90% of records. The authors also solve the problem of fuzzy boundaries of emotional categories, using a dimensional model of emotions and fuzzy classification, which indicates the probability that the record belongs to a specific emotional category.

EmotexStream can be considered the primary analogue to the system, which is supposed to use a created machine learning model. However, Emotex uses a different emotion model – dimensional but not discrete. In addition, Emotex developers created and used their dataset of automatically annotated data using hashtags from the social network Twitter. Emotex, unlike the designed system, do not use the concept of identifying trending words or hashtags for a certain period to simplify the correlation of changes in emotions with important events in the society's life. EmotexStream is also proposed to be used for real-time analysis of message flows.

Fabio Calefato, Filippo Lanubile, Nicole Novielli [2] developed EmoTxt – an open-source toolkit for recognising emotions in the text. EmoTxt supports the recognition of emotions in the text and training on other models of classification emotions using manually annotated data.

Rahul Venkatesh Kumar, Shahram Rahmanian, Hessa Albalooshi [3] developed an experimental model for emotion detection in short messages and posts on social networks. Several standard algorithms were used in the study: logistic regression, naive Bayesian classifier, CNN – BiLSTM and CNN – LSTM with fastText vectorisation. The model using logistic regression had the best accuracy – 91.83%, in the second place – the CNN-LSTM model.

Srinivasu Badugu and Matla Suhasini [4], in their work, chose a rule-centred approach. As a result, the accuracy of the model was 85%. However, the disadvantage of the developed model that can be considered is that it does not reveal individual emotions. Still, it only indicates to which group (Happy-Active, Happy-Inactive, Unhappy-Active, Unhappy-Inactive) a specific record belongs.

Nafis Irtiza Tripto, Mohammed Eunus Ali [5] suggested the use of deep learning to determine sentiment and emotion in the comments written in Bengali.

Nafis Irtiza Tripto and Mohammed Eunus Ali point out that LSTM has better accuracy, but CNN is much faster. In the case of word2vec, here Skip Gram has the best accuracy estimates.

Luyao Ma, Long Zhang, Wei Ye, Wenhui Hu [6] proposed deep learning architecture with biLSTM neural network, which uses an emotion-oriented attention network.

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro [7] developed a model in which BiLSTM layers, the self-attention layer and convolutional neural networks (CNN) are combined. Lu Chen, Amit Sheth, Krishnaprasad Thirunarayan, Wenbo Wang [8] created an automatically annotated dataset from posts on the social network Twitter for their research. To classify records according to their emotion, the researchers chose a logistic regression classifier from the open-source library Liblinear and a naive Bayesian classifier implemented in the Weka program.

3. Material and methods

Dataset selection. The manually annotated DailyDialog dataset [9], presented in 2017, was chosen as the initial dataset for the study. It contains seven categories of emotions that correspond to the discrete model of Paul Ekman [10] (anger, disgust, fear, happiness, sadness, surprise and a "no emotion" category), and consists of 13 118 dialogues, each of them contain a different number of lines.

The DailyDialog data does not contain unnecessary characters, emoji's, links, etc. Its only drawback is its imbalance: the number of records that belong to the "anger" category is 1022, "disgust" – 353, "fear" – 174, "happiness" – 12885, "sadness" – 1150, "surprise" – 1823, "no emotion" – 85572. To avoid the possibility of bias of the model related to the largest class, it is necessary to investigate the behaviour of the classifier by removing different numbers of records or adding records that belong to the same categories from other datasets.

Data representation in vector format. Although machine-learning algorithms deal with numbers, the information in the selected dataset is presented as text. Therefore, to classify these records using machine-learning classifiers, they must be written as numerical vectors. This process is called vectorisation or word embedding. The most popular methods of representing data in vector format are TF-IDF, Bag-of-Words, fastText, Glove, Word2Vec (CBOW, Skip-Gram).

Classical machine learning methods. The critical task for the system of automatic emotion detection from text is the classification – the prediction of which of the known classes of emotions the record or publication belongs to. The classical supervised machine learning methods include the following algorithms: logistic regression, decision tree, naive Bayesian classifier, support vector machine, k-nearest neighbours. In addition, the random forest algorithm, which belongs to the family of ensemble algorithms, was used to classify the records in the study of Marco Polignano, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro [7]. The AdaBoost method also belongs to this family. This method was not used in any of the reviewed studies, but in several works [3, 4], the XGBoost method was used, which also boosted (ensemble) algorithms. Thus, for comparison, the classical and ensemble methods of machine learning were chosen, which are most often used in similar studies to create an automatic emotion detection from text model. To select the best classification method for its further use in real-world data analysis, it is necessary to compare the accuracy, precision, recall, and F-measure indicators for all the algorithms mentioned above.

Deep learning methods. Convolutional and recurrent neural networks and multilayer perceptron are used to detect emotions in textual data.

- Convolutional Neural Network, CNN. CNN consists of input and output layers, as well as several hidden layers. Hidden CNN layers typically consist of convolutional layers, aggregation layers, fully bonded layers, and normalisation layers. Convolution layers apply a convolution operation to the input, passing the result to the next layer. Although convolutional networks are mainly used for image analysis, in some studies [3, 5, 7], their use allowed obtaining good accuracy results than other deep learning methods.
- Recurrent Neural Networks, RNN. Recurrent neural networks are networks that contain recurrent connections and can store information. The recurrent connections allow transferring data from one-step of the network learning to another. LSTM network is a type of RNN. LSTM (Long Short-Term Memory) is an RNN capable of learning long-term dependencies. LSTM networks consist of repeating elements. Each element contains four layers, and one key difference of this type of neural networks is that it has a cell of long short-term memory. The ability of LSTM networks to successfully learn data with long-term dependencies makes them a good choice for solving problems in which both input and output information is represented in the form of sequences of some elements (e.g., letters, words, sentences) [11].

Language and libraries. The most popular languages for machine learning are Python, R, Java, C++. The advantage of Python, among other languages, for creating an automatic emotion detection from text machine-learning model is its support for a large number of libraries for:

- Classical machine learning methods: Scikit-Learn;
- Creating artificial neural networks, using deep learning: TensorFlow, Keras, PyTorch;
- Natural language data pre-processing: NLTK, spaCy, WordNet;
- Operations with arrays and matrices: NumPy;
- Tables: pandas;
- Data visualisation: Matplotlib, seaborn, Plotly.

The Scikit-Learn library supports data pre-processing, data dimension reduction, and machine learning models for regression, classification, or cluster analysis. However, Scikit-Learn does not have comprehensive support for creating deep learning models. The Keras library was chosen to create artificial neural networks, which serves as a high-level API for TensorFlow 2. Keras allows building sequential models in the form of a graph, the vertices of which are layers of a specific type with a given number of nodes. In addition, Keras allows to the combination of the results from several separate parts of the neural network for their further processing. Such a structure is not linear.

TensorFlow allows importing a pre-trained machine-learning model for later use in other programs. TensorFlow also supports low-level tensor operations using CPUs, GPUs, and tensor processing units.

The NLTK library in this study is used for text pre-processing: tokenisation, removal of stop words, stemming, and lemmatisation. In addition, with the help of functions from this library, one can find the most popular n-grams and parts of speech of individual tokens, recognise named entities etc.

Additional libraries that simplify work with natural language include Regex and emoji – to use regular expressions and replace emoji's with their meanings, respectively.

4. Experiments

According to the results of the analysed works, each set of text classification data has its combination of such factors as text formatting (removal of punctuation characters, replacement of emoji with appropriate words, etc.), the number of entries of each category in the training dataset, text pre-processing, vectorisation or word embedding, machine learning model for text classification, model's parameters and architecture etc., that will result in the best possible classification accuracy.

Therefore, when developing such systems for detecting the text features, the most critical step is to choose the best combination of the factors mentioned above.

The optimal number of records for each category. To avoid bias of the future machine-learning model to the type with the most significant number of documents, it is essential to make sure that the training dataset is balanced before training the model or studying the influence of other factors. Therefore, the initial dataset, The DailyDialog, contains a different number of records of each category.

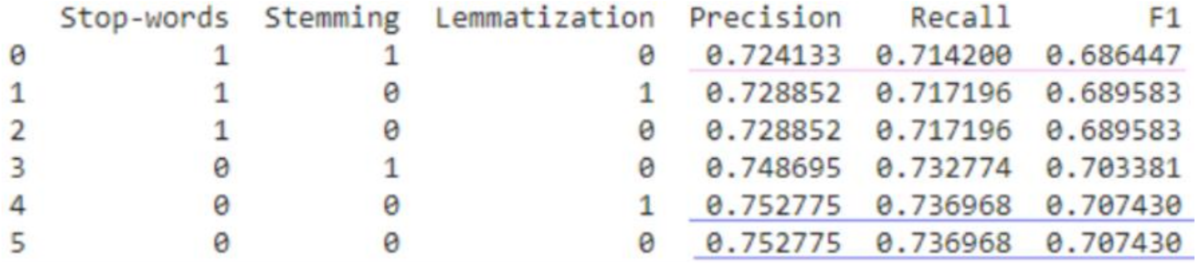
To investigate how many records of each category will be the most optimal choice, three experiments with the following initial conditions: pre-processing of text: removal of stop words and lemmatisation, vectorisation method: Bag-of-words, classifier: logistic regression, binary classification approach used for multi-class classification: One-Vs-Rest were conducted.

In the first experiment, with the original number of records in each category, there are high scores (precision - 86%, recall - 96%, F-measure - 91%) for the category with the largest number of records, low recall and F-measure for other classes (e.g., category 3: precision - 67%, recall - 12%, F-measure - 20%). In addition, the model did not correctly detect any record that belongs to category 2.

In the third experiment (deletion of 70.5 thousand records of category 0 and 10 thousand records of category 4), the scores mentioned above increased for certain classes. Still, the model's accuracy is too low (57%), so the option of deleting 60 thousand records of category 0 as a base distribution of documents for the following experiments (accuracy - 67%) was chosen.

Text pre-processing. First, it should be noted that the records in the dataset The DailyDialog do not require spell check (because the dialogues were collected and annotated manually), lowercasing is the default, and punctuation is removed.

Methods used: logistic regression, OneVSRest approach, TF-IDF is a vectorisation method, precision, recall and F-measure - weighted average.

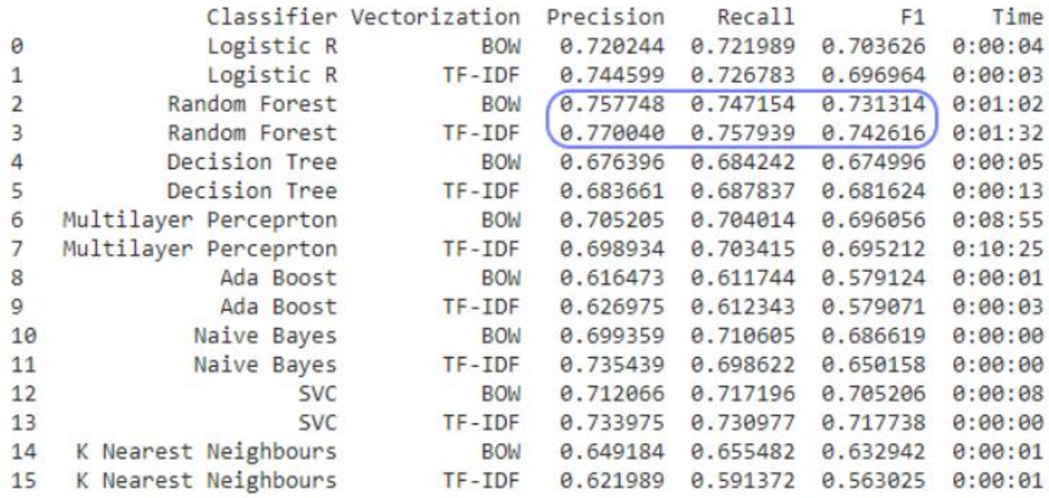


```
   Stop-words  Stemming  Lemmatization  Precision    Recall        F1
0           1         1              0   0.724133  0.714200  0.686447
1           1         0              1   0.728852  0.717196  0.689583
2           1         0              0   0.728852  0.717196  0.689583
3           0         1              0   0.748695  0.732774  0.703381
4           0         0              1   0.752775  0.736968  0.707430
5           0         0              0   0.752775  0.736968  0.707430
```

Figure 1: Table of comparison of text pre-processing methods

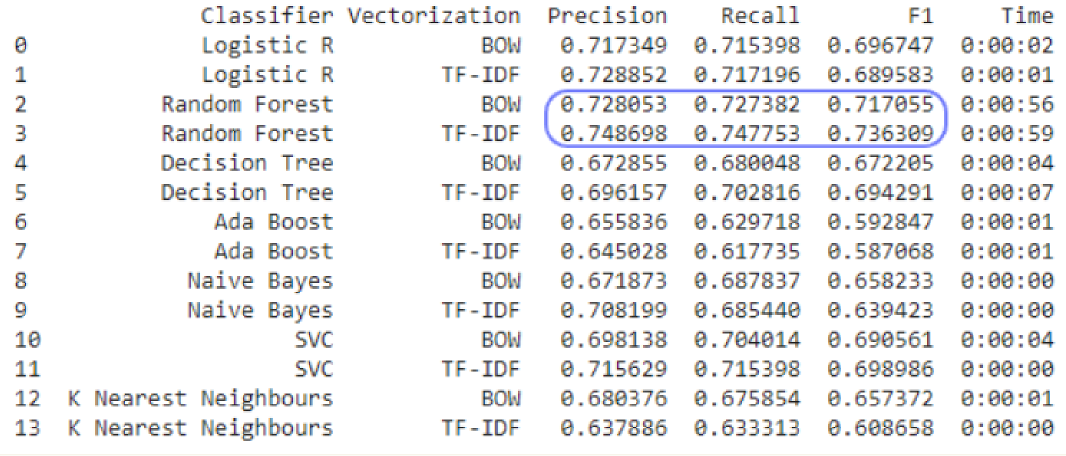
Thus, the removal of stop words reduces the accuracy of the model. Lemmatization does not affect the performance of the model.

Vectorisation and classification. In this step, results were obtained for all possible combinations of vectorisation methods (Bag-of-Words, TF-IDF) and classical machine learning methods for sort (logistic regression, random forest, decision tree, multilayer perceptron, Adaptive Boosting algorithm, naive Bayesian classifier, support vector machine, the technique of k-nearest neighbours).



```
              Classifier Vectorization  Precision    Recall        F1     Time
0             Logistic R           BOW   0.720244  0.721989  0.703626  0:00:04
1             Logistic R        TF-IDF   0.744599  0.726783  0.696964  0:00:03
2          Random Forest           BOW   0.757748  0.747154  0.731314  0:01:02
3          Random Forest        TF-IDF   0.770040  0.757939  0.742616  0:01:32
4          Decision Tree           BOW   0.676396  0.684242  0.674996  0:00:05
5          Decision Tree        TF-IDF   0.683661  0.687837  0.681624  0:00:13
6   Multilayer Perceprton          BOW   0.705205  0.704014  0.696056  0:08:55
7   Multilayer Perceprton       TF-IDF   0.698934  0.703415  0.695212  0:10:25
8              Ada Boost           BOW   0.616473  0.611744  0.579124  0:00:01
9              Ada Boost        TF-IDF   0.626975  0.612343  0.579071  0:00:03
10           Naive Bayes           BOW   0.699359  0.710605  0.686619  0:00:00
11           Naive Bayes        TF-IDF   0.735439  0.698622  0.650158  0:00:00
12                   SVC           BOW   0.712066  0.717196  0.705206  0:00:08
13                   SVC        TF-IDF   0.733975  0.730977  0.717738  0:00:00
14   K Nearest Neighbours          BOW   0.649184  0.655482  0.632942  0:00:01
15   K Nearest Neighbours       TF-IDF   0.621989  0.591372  0.563025  0:00:01
```

Figure 2: Table of comparison text vectorisation and classification methods without deleting stop words



```
             Classifier Vectorization  Precision    Recall        F1     Time
0            Logistic R           BOW   0.717349  0.715398  0.696747  0:00:02
1            Logistic R        TF-IDF   0.728852  0.717196  0.689583  0:00:01
2         Random Forest           BOW   0.728053  0.727382  0.717055  0:00:56
3         Random Forest        TF-IDF   0.748698  0.747753  0.736309  0:00:59
4         Decision Tree           BOW   0.672855  0.680048  0.672205  0:00:04
5         Decision Tree        TF-IDF   0.696157  0.702816  0.694291  0:00:07
6             Ada Boost           BOW   0.655836  0.629718  0.592847  0:00:01
7             Ada Boost        TF-IDF   0.645028  0.617735  0.587068  0:00:01
8           Naive Bayes           BOW   0.671873  0.687837  0.658233  0:00:00
9           Naive Bayes        TF-IDF   0.708199  0.685440  0.639423  0:00:00
10                  SVC           BOW   0.698138  0.704014  0.690561  0:00:04
11                  SVC        TF-IDF   0.715629  0.715398  0.698986  0:00:00
12 K Nearest Neighbours           BOW   0.680376  0.675854  0.657372  0:00:01
13 K Nearest Neighbours        TF-IDF   0.637886  0.633313  0.608658  0:00:00
```

Figure 3: Table of comparison text vectorisation and classification methods with deleting stop words

Therefore, the following conclusions can be drawn from the obtained tables: removal of stop words reduces the precision, recall and F-measure by 2-3% for almost all of the classifiers (except for the Adaptive Boosting algorithm and the method of k-nearest neighbours); TF-IDF increases precision, recall and F-measure by 1-3% compared to the Bag-of-Words vectorisation method for almost all of the classifiers (except the Adaptive Boosting algorithm and the k-nearest neighbour method); classification methods with the best indicators: random forest (77% of weighted precision), logistic regression (74%), naive Bayesian classifier (73.5%).

Parameters of the classifiers. To study the influence of the parameters of classification methods on the quality scores of the model, logistic regression and Bag-of-Words were chosen as a method of vectorisation. The main logistic regression parameters from the sklearn library are the algorithms used for optimisation ("solvers"), and parameter C is the inverse of the regularisation strength. The change in parameter C increased the weighted F-measure by 2% max, but the model's accuracy is still relatively low. In addition, the change of algorithms used for optimisation has hardly changed the accuracy of classification. The effect of the class_weight parameter is on the accuracy of the random forest method was also investigated. This parameter affects the weight of the classes and is used to classify unbalanced datasets. The value of the parameter class_weight = 'balanced_subsample' slightly increases recall for classes with low support (1-5%) and decreases precision (1-7%) for most classes.

Binary classification strategies for multi-label classification. Classifiers such as logistic regression, perceptron, and support vector machine were made for binary classification. However, they do not support classification problems with more than two classes. The strategies used for multi-class classification apply these methods: One-vs-Rest or One-vs-One. To compare binary classification strategies for multi-label classification, logistic regression as a classification method, TF-IDF was chosen as a vectorisation method, and lemmatisation – as a text pre-processing.

Analysis of the obtained results: both approaches have a high recall, low - precision for classes with low support; there are no correctly identified records for the classes with minor support; the One-vs-One approach slightly increases some scores (1-2%).

Conclusions on the first part of choosing the best model (classical methods of machine learning):

- The DailyDialog dataset is not suitable for classical machine learning methods and further research, as its imbalance negatively affects the quality of the classification model;
- To train which The DailyDialog dataset records were used, the final classifier model is biased against classes with the most extensive support. It poorly classifies the minor classes;
- The best results were obtained for the following combination: lemmatisation + TF-IDF + random forest;
- Changing the parameters of the classifiers does not sufficiently change the precision, recall, F-measure of the model;
- The accuracy of the obtained classification model is quite low for its further use.

Comparison of deep learning with classical machine learning methods. Since traditional machine learning methods provide some unsatisfactory results, for further research, a temporary convolutional neural network [12-14] was chosen as a basic initial method, which, according to Diardano Raihan [12], is an alternative to recurrent architecture that can accept long sequences and does not suffer from "forgetting" important information. For word embedding, a pre-trained GloVe model [15], the dictionary containing 400 thousand words, the dimension of the vector for each word is 100. For training and classification, the initial number of entries in the dataset The DailyDialog was kept; for pre-processing, only lemmatisation was used. As a result, the model has the following accuracy indicators: training set - 86.9%, validation set - 89.6%, test set - 84.2%. The resulting model has better accuracy, but it does not remain very objective concerning the classes with the most support.

Creating a new data set. Because popular datasets used to build a machine-learning model for automatic emotion detection from the text use an uncompleted list of basic emotions according to Paul Ekman [10], records from other datasets to preserve the original emotion categories: Emotions dataset for NLP, Kaggle [16] and Semeval-2018 Task 1, E-c [17] were used. In addition, every three coherent entries from the categories "no emotion" and "happiness" were combined. So now, the distribution of records by type looks like this:

```
There are 27690 records in the dataset now.
```

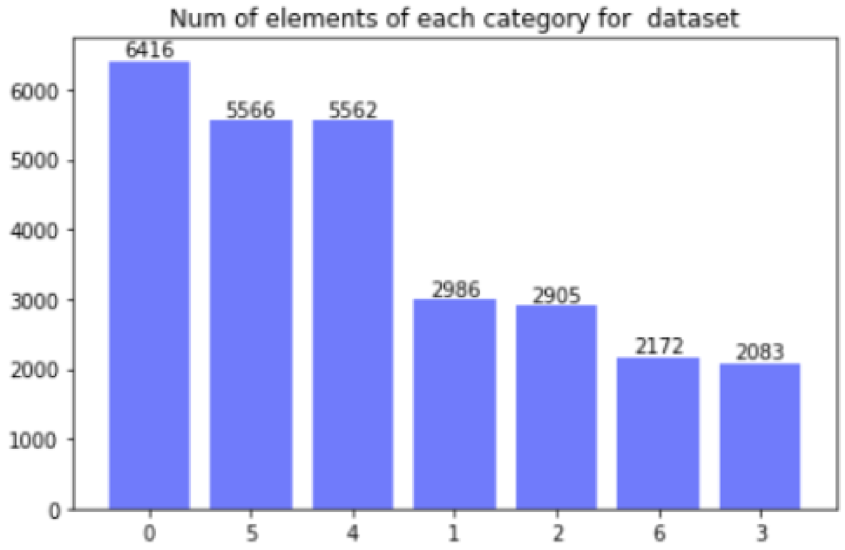
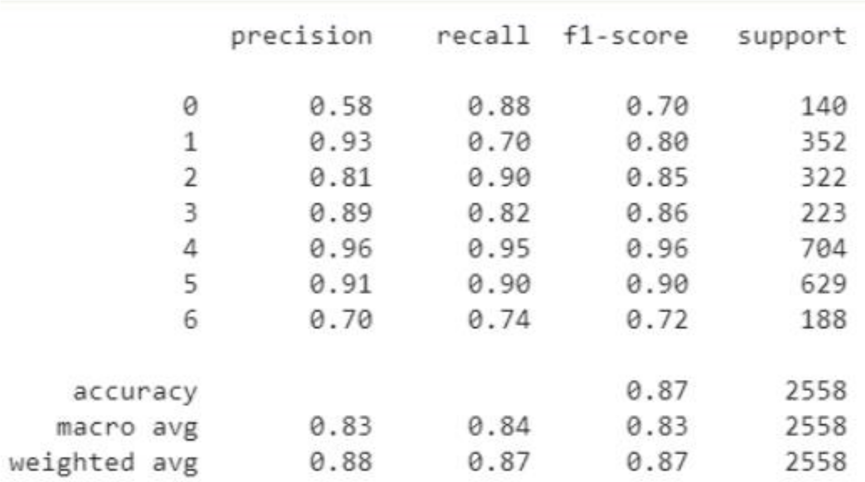


Figure 4: The number of records of each category in the new dataset

Now the temporary convolutional neural network shows satisfactory results for all classes:



```
              precision    recall  f1-score   support

           0       0.58      0.88      0.70       140
           1       0.93      0.70      0.80       352
           2       0.81      0.90      0.85       322
           3       0.89      0.82      0.86       223
           4       0.96      0.95      0.96       704
           5       0.91      0.90      0.90       629
           6       0.70      0.74      0.72       188

    accuracy                           0.87      2558
   macro avg       0.83      0.84      0.83      2558
weighted avg       0.88      0.87      0.87      2558
```

Figure 5: Precision, recall, F-score for new balanced dataset

The deep learning model shows 87% accuracy on the data set for validation, the random forest method - 67% (-20%).

Text pre-processing. The DailyDialog and Emotions dataset for NLP, Kaggle datasets use standard formatting: removing punctuation and lowercasing. Since the dataset Semeval-2018 Task 1, E-c contains posts from the social network Twitter, the records are also further cleared of mentions of other users, links, numbers. Emoji are replaced with the appropriate values. Hashtags are not deleted.

Table 1

Comparison of text pre-processing methods

Stop-words removal	Stemming	Lemmatisation	Precision	Recall	F-score	Accuracy on the validation set	Accuracy on the test set
1	1	0	0.87	0.86	0.86	0.67	0.85
1	0	1	0.88	0.85	0.86	0.85	0.85
1	0	0	0.87	0.85	0.86	0.85	0.85
0	1	0	0.7	0.69	0.69	0.68	0.69
0	0	1	0.87	0.85	0.86	0.85	0.85
0	0	0	0.89	0.87	0.87	0.87	0.86

Therefore, in this case, the primary stages of pre-processing reduce the accuracy of the model.

Word embedding. For deep learning, either pre-trained word embedding models or models trained on the corpus can be used. In this study, the number of words in the combined dataset is not enough to obtain satisfactory accuracy of the deep learning model. Also, reducing the dimension of the vector (from 100 to 50), adding to each word its semantic score from the AFINN dictionary, and indicators such as valence, arousal and dominance from the ANEW dictionary can significantly increase the accuracy of the model.

Table 2

Comparison of vector embedding models trained on a combined data set

Word embedding model	Precision	Recall	F-score	Accuracy on a validation set	Accuracy on a test set
Word2Vec, without text pre-processing	0.55	0.59	0.53	0.61	0.62

Word2Vec, stop-words removal and lemmatization	0.48	0.52	0.48	0.52	0.56
fastText, without text pre-processing	0.49	0.56	0.5	0.56	0.7
Word2Vec + AFINN + ANEW, without text pre-processing	0.69	0.69	0.68	0.69	0.67

Pre-trained models have almost the same good accuracy. The addition of semantic scores does not affect the result.

Table 3

Comparison of pre-trained word embedding models

Word embedding model	Precision	Recall	F-score	Accuracy on a validation set	Accuracy on a test set
Stanford's Glove	0.88	0.87	0.87	0.87	0.86
Google's Word2Vec	0.88	0.86	0.87	0.86	0.87
fastText on Wikipedia	0.88	0.88	0.89	0.88	0.86
fastText + AFINN + ANEW	0.88	0.87	0.87	0.87	0.86
Glove + AFINN + ANEW	0.88	0.86	0.87	0.87	0.85

Dictionary size. Methods used: fastText for word embedding, temporary convolutional neural network as a classifier. According to the results in Table 4, the size of the dictionary does not significantly affect the quality of the model.

Table 4

Comparison of the influence of the number of words in the dictionary on the quality of the model

Number of words in the dictionary	Precision	Recall	F-score	Accuracy on a validation set	Accuracy on a test set
3 000	0.88	0.86	0.87	0.86	0.87
5 000 – baseline	0.89	0.87	0.88	0.87	0.87
10 000	0.89	0.87	0.88	0.88	0.86
15 000	0.88	0.88	0.88	0.88	0.87
20 000	0.88	0.86	0.87	0.87	0.85

Architecture of the model of deep learning. Presented in table 5, models of deep learning are graphs, the vertices of which are layers. All models, except the CNN + BiLSTM architecture, are sequential.

The layers from which the model is built can have different nodes, node types, and activation functions. Parameters such as the number of filters and the filter size are used for convolutional neural network layers. Layers such as pooling operation, the transformation of a matrix into a single array of values (flatten), exclusion of neurons (dropout) are also used.

Table 5

Comparison of deep learning models

NN architecture	Precision	Recall	F-score	Accuracy on a validation set	Accuracy on a test set
ML perceptron (Dense layers)	0.56	0.53	0.54	0.54	0.56
CNN 1	0.71	0.69	0.7	0.7	0.71

CNN 2	0.74	0.73	0.73	0.72	0.72
BiLSTM 2	0.88	0.87	0.87	0.88	0.85
BiLSTM 1	0.89	0.89	0.89	0.88	0.86
CNN + BiLSTM	0.87	0.85	0.85	0.85	0.84
RNN	0.87	0.85	0.85	0.85	0.845
TCN – baseline	0.89	0.87	0.88	0.87	0.87

The best results were obtained for deep learning models with layers of long-short term memory and temporary convolutional neural network. For further research, the artificial neural network “BiLSTM 1” was chosen because it is larger than other models scores of precision, recall, F-measure and accuracy. Its architecture is given in Table 6.

Table 6
Architecture of the model of deep learning "BiLSTM 1"

Layer	Number of cells	Activation function	Dropout
Bidirectional LSTM	128	tanh, sigmoid	0.2
Bidirectional LSTM	256	tanh, sigmoid	0.2
Bidirectional LSTM	128	tanh, sigmoid	
Dense (perceptron)	7	softmax	

Conclusions on the second part of choosing the best model (deep learning):

- Combining records from multiple datasets eliminated the imbalance of the original dataset and the bias of the model relative to the most significant class;
- Best results were obtained for the following combination: pre-trained fastText word embedding model + deep learning model with long-short memory layers, without text pre-processing;
- The obtained model shows good results so that it can be used for further experiments on the analysis of real-world data collected from information resources

5. Results and discussions

The system of monitoring public sentiment, a part of which the created machine-learning model has to be, may use data from several information resources depending on the purpose and scope of its application. For example, as an information source for experiments with the collected real-world data, the social network Twitter was chosen because the entries in this social network are short. In addition, they often contain hashtags that simplify the search by keywords.

In addition, according to Wakamiya, S. et al. [18], social networks contain a large amount of public data, available in real-time and rich in emotional content. Therefore, such data sources are well suited for behavioural research [19-27], both for studies the emotions of a particular person and specific groups. Data can be collected according to the following parameters: start and end dates (required parameters), keywords, geographical location (specified by the author of the publication), and the minimum number of likes. The user must also specify the maximum number of records that will be collected for each day of the interval between the start and end dates. To get more relevant data, one should increase the minimum number of likes. For the first experiment, the following parameters: start date - 1 May 2021; end date - 11 May 2021 (inclusive); the maximum number of records - 500; the minimum number of likes for each record - 200; the keyword is India was specified.

The result of the analysis of records is the following plot:

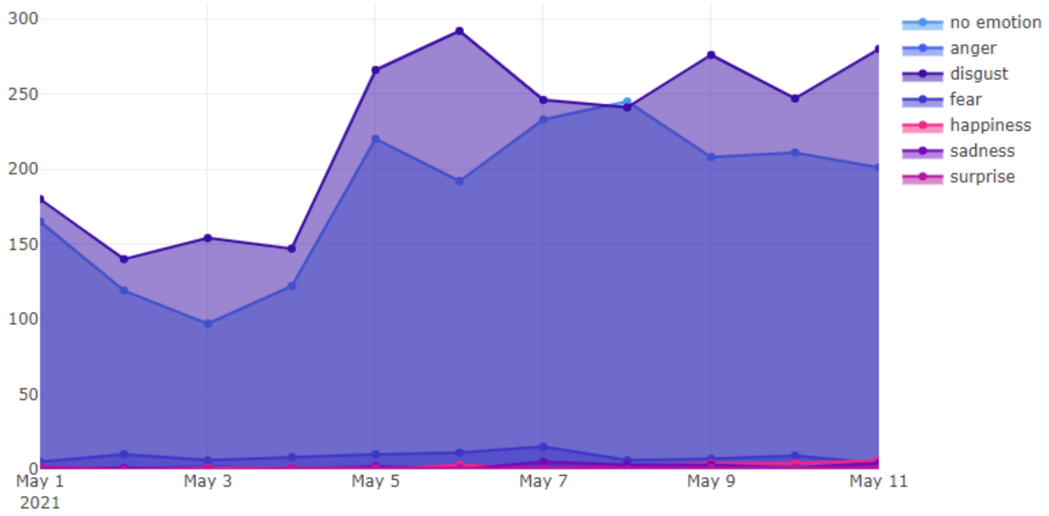
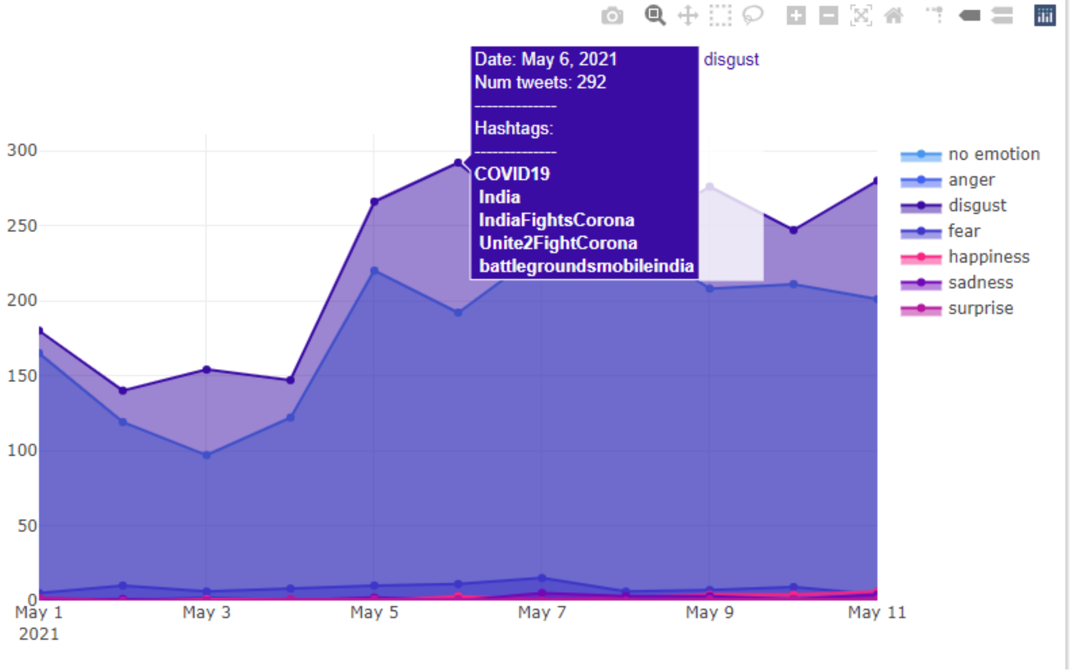


Figure 6: The results of the first analysis of the collected data

A quick analysis of the obtained graph shows that the vast majority of records belong to the categories of "disgust" and "no emotions". The level of other emotions is deficient.

When hovering the plot by the cursor to the point that corresponds to the peak of negative emotions - 6 May, we get a list of five popular hashtags on this day (Fig. 7).


Date: May 6, 2021
Num tweets: 292

Hashtags:

COVID19
India
IndiaFightsCorona
Unite2FightCorona
battlegroundsmobileindia

Figure 7: The results of the first analysis of the collected data

The most popular words and hashtags on 6 May are related to the COVID-19 pandemic. According to google.com, 6 May in India was the peak in the number of new cases of COVID-19 in two weeks, from 29 April to 11 May. It is worth noting that the predominant emotion in such a situation should not be disgust but sadness or anger. This result may be the reason that the category "disgust" was added from the data set Semeval-2018 Task 1, E-c, which is intended for multi-class classification. Thus, the emotion of "disgust" may be a secondary emotion for such records that, in the case of a simple type, would fall into the categories of "sadness" or "anger". The obtained results can be interpreted as strong negative emotions prevailing in the population. If we remove the two "biggest" emotions and increase the scale, we can see 2 small peaks of emotions on May 9 and 11. The recordings of 9 May primarily relate to foreign and domestic political issues.

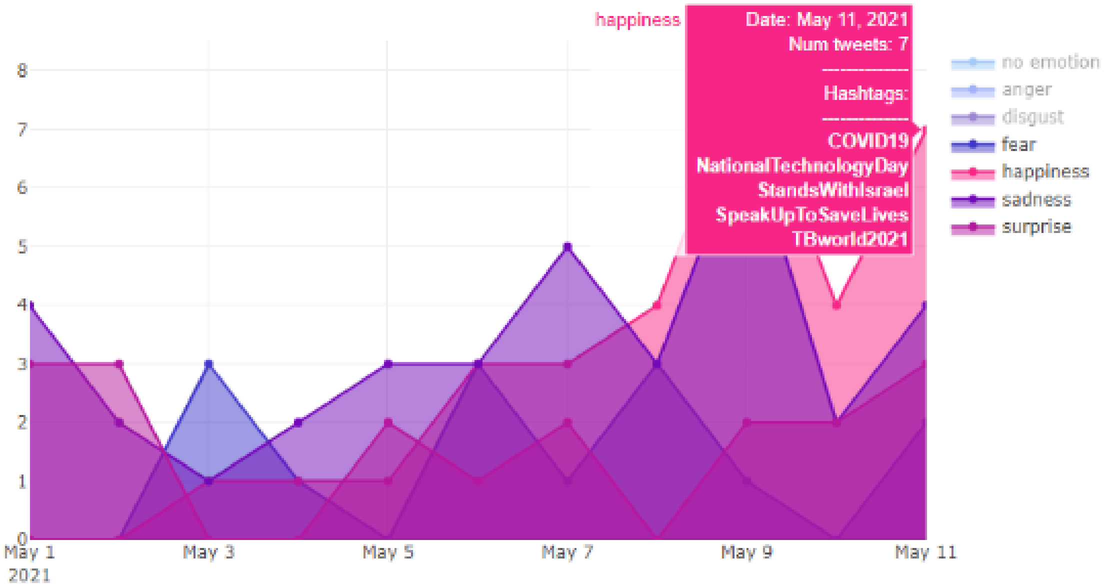


Figure 8: Peak of emotions on May 11

The slight increase in the number of 11 May records in the “happiness” category can be explained by the fact that one date in India is a celebration of National Technology Day (the second most popular hashtag). Thus, as a result of the first analysis of the collected data, it was found that the model incorrectly classifies negative emotions and marks the records as belonging to the category of “disgust”. Therefore, to correct this error, this category (now the data set does not contain records from the Semeval dataset) and re-trained the model and tokenizer were deleted.

The dictionary size to 10,000 words and added semantic estimates from the AFINN and ANEW dictionaries were also increased. The dimension of the embedding vector for each word risen from 300 to 304. According to table 4, the addition of such indicators does not increase the model’s accuracy that processes large-scale vectors. However, such semantic scores can improve the accuracy of real-world classification data collected from the information resource. Now the model has the following scores: precision - 90%, recall - 89%, F-score - 0.89%, accuracy - 89% (on data for validation), accuracy on test data - 88%. The following experiment concerns the reaction of social network users to the blocking of the Suez Canal from 23 March to 29. The data were selected from 20 to 31 March.

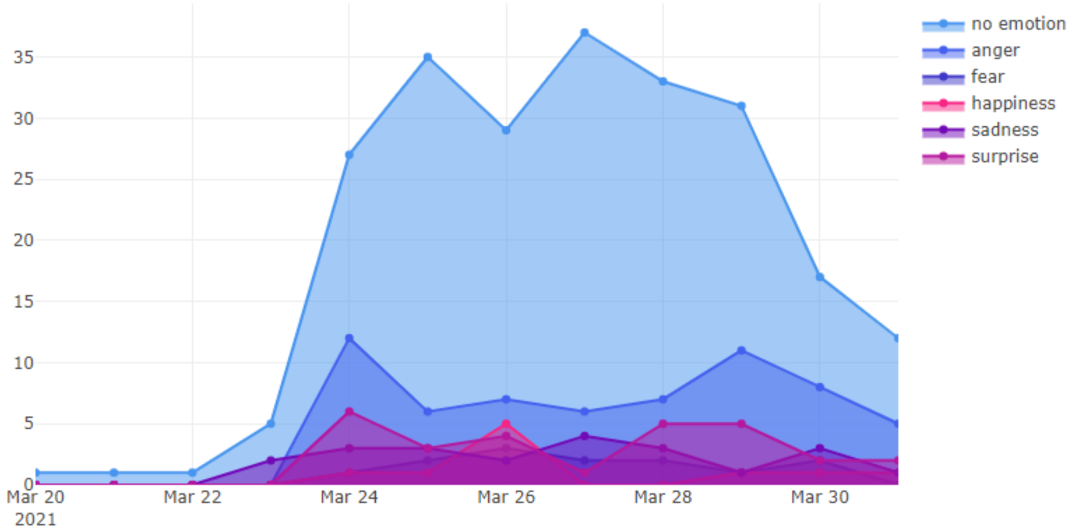


Figure 9: Reaction to Suez Canal blockage

The obtained plot shows an increase in the number of "non-emotional" records after 23 March; March 24 and 29 - bursts of anger and surprise; the days of maximums of the number of "non-emotional" records do not correspond to the days of increase in the number of published "emotional" records. It is worth noting that the number of collected records is not enough to comprehensively analyse the reaction of social network users to this event.

The following analysis concerns the reaction of users to the international song contest "Eurovision", which took place in 2019. The graph does not show the records of the category "without emotions" because, in this analysis, the change in the number of such publications is not informative.

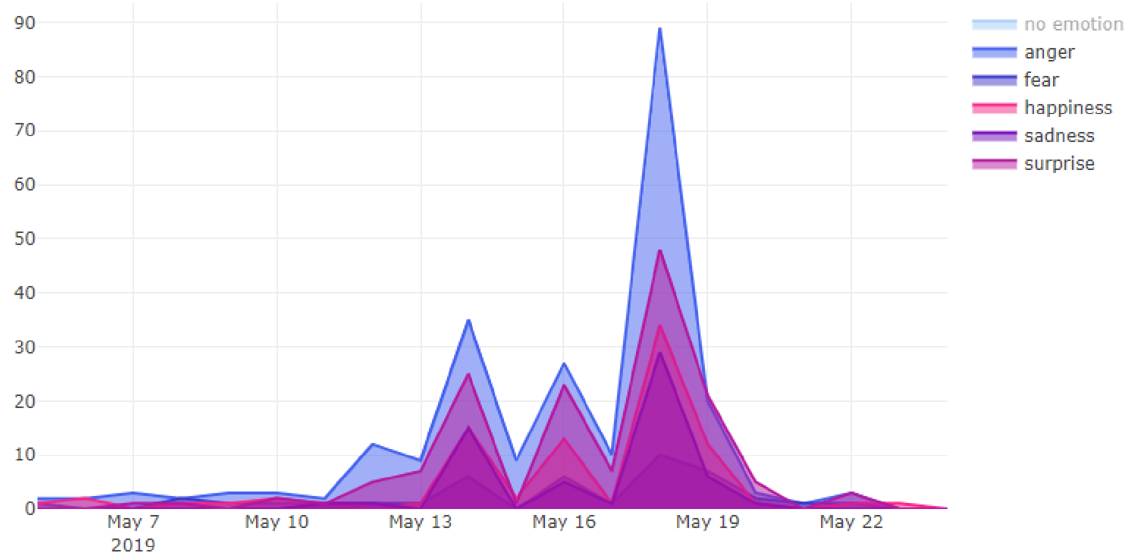


Figure 10: Reaction to the song contest "Eurovision-2019"

In 2019, the first semi-final took place on 14 May, the second semi-final on 16 May, and the final on 18 May. In the same days, there are bursts of emotions of all categories. However, the most prominent peak of emotions falls on the final of the competition. The following reaction concerns the holiday of the New Year 2021. For almost the entire time interval (from 25 December 2020 to 4 January 2021) in the publications collected under the critical phrase "new year", the predominant emotion is happiness. The peak of this emotion falls on 31 December.

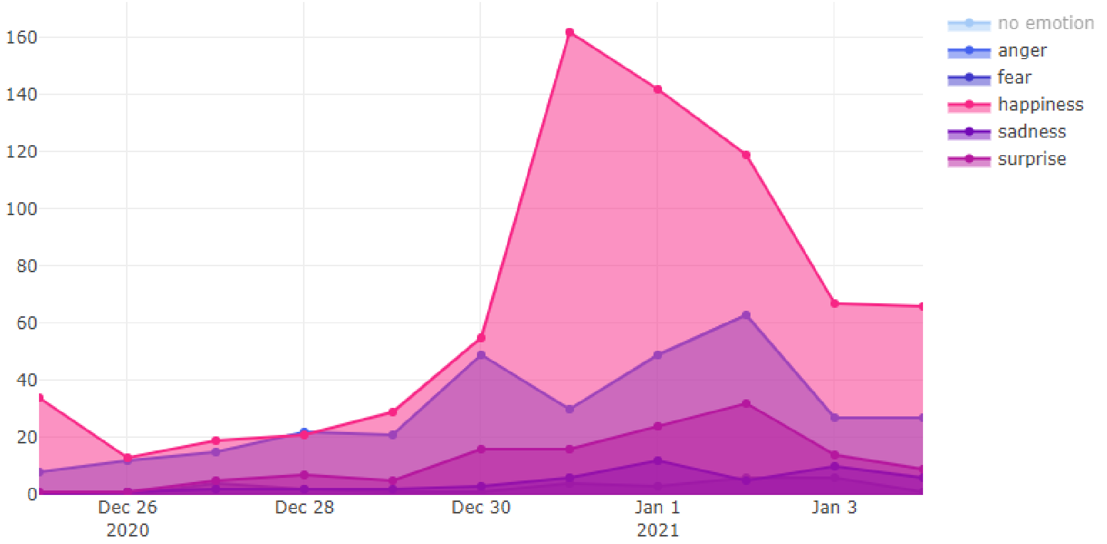


Figure 11: Reaction on a New Year holiday

Fig. 12. shows the reaction of social network users to the 93rd Academy Awards ceremony, which took place on 25 April 2021. In this case, there is a slight peak of all emotions on 26 April, but most of the publications did not contain emotions.

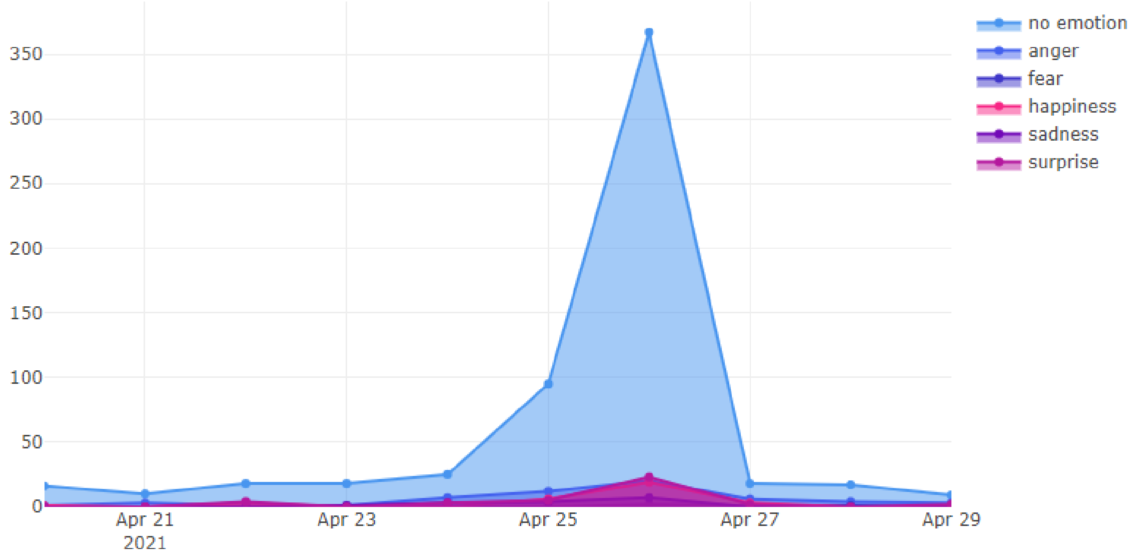


Figure 12: Reaction to the Oscar 2021

By examining the classified records, one can see that the model classifies sentences well with the correct spelling of words, clearly articulated thought, a small number of emoji's and hashtags. On the contrary, the classification of "noisy" records even after formatting (lowercasing, deleting unnecessary characters, replacing emoji with their values) is difficult for the model. It should also be considered that "unfamiliar" words for the model have an index of 0. The vector corresponding to this index also consists of zeros. This word indexing is caused by the dictionary's limitation of the 10,000 most famous words in the data set, so "unfamiliar" words are skipped. As a result, the model takes in a sequence of words that have no logical connection. The model often classifies such publications as "anger" category entries, which explains a large number of tweets in this category on some plots.

In the previous paragraph, some examples of the application of the system of automatic emotion detection from the text were given for such purposes as analysis of the emotional state of the population of a particular geographical region and study on the reaction of users of information resources to socially popular events. Other examples of the use of the system may be its use by moderators of information resources and social networks to maintain the quality of publications and follow trends, business owners to research the market and customer satisfaction with various products or services, government, health and emergency services as early warning systems. The limitations of the created machine-learning model of automatic emotion detection from the text are:

- Data collection from one information source;
- Lack of graphical interface to simplify the interaction of the user and the system;
- Incorrect classification of "noisy" records;
- No connection to a separate server to collect process and store large amounts of data.

Therefore, the model mentioned above of determining emotions in the text is not suitable for a full-fledged analysis of the reaction of the authors of such publications, which are posted on information resources. However, such a model can be improved for further use in detecting emotions that prevail in a certain audience for a certain period to correlate the results with events that occurred during this period. For further refinement of the system, it is proposed to:

- Collect training and test data of the "disgust" category so that the list of possible types corresponds to the list of basic emotions of the discrete model proposed by Paul Ekman;
- Conduct further research on the dataset used for training the model to ensure that the data is of appropriate quality and properly annotated;

- Explore other possible architectures of the deep learning model: using GRU layers, BERT model, a convolutional neural network working with letter-level data, models with different combinations of CNN and (or) LSTM layer sequences;
- Select hyper-parameters using a grid or random search;
- Introduce methods for detecting “noisy” data for its further specified processing or deletion;
- Introduce the replacement of emoji not by their name, but by a popular semantic meaning, e.g., 🎉 - not “tada”, but “excitement”, “happiness”, “holiday”;
- Implement automatic data collection from several sources or one, at the user’s choice;
- Create a graphical interface of the program and organise the system’s infrastructure with sufficiently powerful servers for processing and storing information.

6. Conclusion

The result of the research is a created machine-learning model for automatic emotion detection from text. For experiments with real-world data collected from the information resource, the model of deep learning was chosen, the architecture of which includes layers of bidirectional long-short memory (Bidirectional LSTM) in combination with a pre-trained model of word embedding fastText.

Created and improved due to experiments with real-world data, the machine-learning model showed satisfactory results: precision - 90%, recall - 89%, F-measure - 0.89%, accuracy - 89% (on data for validation), accuracy on test data - 88%. The analysis results obtained using the created model could not be considered reliable because the model had problems with the correct classification of “noisy” data. This problem required the introduction of an algorithm for detecting and further processing such records. Other main proposed stages of system refinement were data collection of the missing category “disgust” and re-training the model; research of different types of architecture of the deep learning model; selection of model’s hyper-parameters; replacing emoji their name but by popular semantic meaning. The developed model can be further improved and used in monitoring the emotional state of society, the population of a geographical region or a specific community.

7. References

[1] M. Hasan, E. Rundensteiner, E. Agu, Automatic emotion detection in text streams by analysing Twitter data, in: Int J Data Sci Anal 7, pp. 35–51 (2019), doi: 10.1007/s41060-018-0096-z.

[2] F. Calefato, F. Lanubile, N. Novielli, EmoTxt: A Toolkit for Emotion Recognition from Text, in: Proceedings of the Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos, ACIIW 2017, doi: 10.1109/ACIIW.2017.8272591.

[3] R. V. Kumar, Sh. Rahmanian, H. Albalooshi, EmotionX-SmartDubai_NLP: Detecting User Emotions in Social Media Text, in: SocialNLP@ACL (2018), doi: 10.18653/v1/W18-3508.

[4] S. Badugu, M. Suhasini, Emotion detection on twitter data using knowledge base approach, in: International Journal of Computer Applications, 2017, doi: 162(10):28-33.

[5] N. Tripto, M. Ali, Detecting multilabel sentiment and emotions from bangla youtube comments, in: Proceedings of the 2nd International Conference on Communication, Computing and Networking, 2018, pp. 1-6, doi: 10.1109/ICBSLP.2018.8554875.

[6] L. Ma, L. Zhang, W. Ye, W. Hu, PKUSE at SemEval-2019 Task 3: emotion detection with emotion-oriented neural attention network, in: Proceedings of the 13th International Workshop on Semantic Evaluation, 2019, pp. 287-291, doi: 10.18653/v1/S19-2049.

[7] M. Polignano, P. Basile, M. Gemmis, G. Semeraro, A comparison of word-embeddings in emotion detection from text using bilstm, cnn and self-attention, in: Proceedings of the Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalisation, 2019, pp. 63-68, doi: 10.1145/3314183.3324983.

[8] L. Chen, A. Sheth, K. Thirunarayan, W. Wang, Harnessing Twitter ‘Big Data’ for Automatic Emotion Identification, in: International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, pp. 587-592, doi: 10.1109/SocialCom-PASSAT.2012.119.

[9] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, Sh. Niu, DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset, in: Proceedings of the Eighth International Joint Conference on Natural Language Processing, 2017, Vol. 1: Long Papers.

[10] P. Ekman, Basic emotions. Handbook Cognit Emot (1999).

[11] O. I. Sheremet, V. S. Zaporozhets, Application of recurrent neural networks to perform machine rewrite, in: Scientific Bulletin of the DSEA, № 1 (25E), 2018, pp. 62 – 68.

[12] D. Raihan, Deep Learning Techniques for Text Classification. Towards Data Science, Medium, URL: https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c.

[13] S. Bai, J. Kolter, V. Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, in: arXiv (2018).

[14] Ph. Rémy, Keras TCN, 2021. URL: https://github.com/philipperemy/keras-tcn.

[15] J. Pennington, R. Socher, Ch. D. Manning, GloVe: Global Vectors for Word Representation, 2004. URL: https://nlp.stanford.edu/projects/glove/

[16] Emotions dataset for NLP. Kaggle, URL: https://www.kaggle.com/praveengovi/emotions-dataset-for-nlp

[17] S. M. Mohammad, F. Bravo-Marquez, M. Salameh, S. Kiritchenko, Semeval-2018 Task 1: Affect in Tweets, in: Proceedings of International Workshop on Semantic Evaluation, (SemEval-2018), pp. 1-17, doi: 10.18653/v1/S18-1001.

[18] S. Wakamiya, L. Belouaer, D. Brosset, R. Lee, Y. Kawai, K. Sumiya, C. Claramunt, Measuring crowd mood in city space through twitter, in: International Symposium on Web and WirelessGeographical Information Systems, 2015, pp 37-49, doi: 10.1007/978-3-319-18251-3_3.

[19] S. Albota, Resolving conflict situations in reddit community driven discussion platform, in: Proceedings of the 4th International conference on computational linguistics and intelligent systems, (COLINS 2020), Vol. 2604, pp. 215–226.

[20] S. Albota, Contradictory statement as a basis for conflict resolution strategies, in: Proceedings of the international workshop on conflict management in global information networks (CMiGIN 2019) co-located with 1st International conference on cyber hygiene and conflict management in global information networks, (CyberConf 2019), Vol. 2588, pp. 336-345.

[21] D. Nazarenko, I. Afanasieva, N. Golian, V. Golian, Investigation of the Deep Learning Approaches to Classify Emotions in Texts, volume Vol-2870 of CEUR Workshop Proceedings, 2021, pp. 206-224.

[22] I. Bekhta, N. Hrytsiv, Computational Linguistics Tools in Mapping Emotional Dislocation of Translated Fiction, volume Vol-2870 of CEUR Workshop Proceedings, 2021, pp. 685-699.

[23] I. Spivak, S. Krepych, O. Fedorov, S. Spivak, Approach to Recognizing of Visualized Human Emotions for Marketing Decision Making Systems, volume Vol-2870 of CEUR Workshop Proceedings, 2021, pp. 1292-1301.

[24] Z. Kochuieva, N. Borysova, K. Melnyk, D. Huliieva, Usage of Sentiment Analysis to Tracking Public Opinion, volume Vol-2870 of CEUR Workshop Proceedings, 2021, pp. 272-285.

[25] Artemenko, O., Pasichnyk, V., Kunanets, N., Shunevych, K.: Using sentiment text analysis of user reviews in social media for e-tourism mobile recommender systems, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 259-271.

[26] Bobicev, V., Kanishcheva, O., Cherednichenko, O.: Sentiment Analysis in the Ukrainian and Russian News, in: First Ukraine Conference on Electrical and Computer Engineering (UKRCON), 2017, pp. 1050-1055.

[27] S. Bhatia, M. Sharma, K. Bhatia, P. Das, Opinion Target Extraction with Sentiment Analysis, volume 17(3) of International Journal of Computing, 2018, pp. 136-142.