UDC 004.8 http://doi.org/10.15407/pp2024.01.23

*V. R. Kobchenko, V.M. Shymkovysh, P.I. Kravets, A.O. Novatskyi, L.L. Shymkovysh, A.Yu. Doroshenko*

# AN INTELLIGENT CHATBOT FOR EVALUATING THE EMOTIONAL COLOURING OF A MESSAGE AND RESPONDING ACCORDINGLY

In this article, a recurrent neural network model, a database designed for neural network training, and a software tool for interacting with a bot have all been created. The architecture of the neural network model underwent optimization to enhance classification outcomes. Furthermore, work was conducted on enhancing the user interface. The developed application was tested, and the results were demonstrated. The resulting model demonstrated 85% accuracy in determining sentiments. The implemented application has got basic design which can be cutomized and some settings for chatbot. Further improvement of the model's classification quality can be achieved by collecting a larger and better organised dataset or by researching other RNN architectures.

Key words: recurrent neural networks, sentiment analysis, Python, tensorflow, keras, chatbot.

## Introduction

Emotions in the broad sense are psychological and physiological reactions that arise in a person due to certain events, stimuli or thoughts. They are reflected through internal experiences and external expressions.

Usually, it is very easy to determine a person's emotions by observing their behavior, looking at their faces, or listening to their voices. However, the text itself does not have an emotional color. Instead, it can evoke emotions in the reader and reflect or express the emotional state of the author. This makes the task of assessing emotionality more difficult. And we usually don't need to know exact emotions, it is enough to know are they positive or negative.

This task is known as sentiment analysis [1,2] and is used in many areas of life. For example, processing reviews for products in a store can greatly simplify the work of marketers.

Besides marketing it is used in psychology, both for usual conversations and professional online help by psychologists.

And while it is not hard task for people, it can be still useful to automatize such analysis [3]. For marketers it will help, because they have to work with big number of reviews. Automatic detection of sentiments can facilitate the work of psychologists too. And this is possible to do with neural networks [4,5].

Artificial neural networks (ANNs) are computer models that imitate the functioning of biological neural networks [6,7]. Structurally they consist of interconnected artificial neurons that exchange signals in the form of numbers. There can be hundreds, thousands or even millions of such neurons. Each neuron receives input data, performs a certain function on this data and transmits the result to the next neurons in the network.

ANNs can be used in a wide range of applications, such as natural language processing, computer vision and speech recognition [7-17].

Recurrent Neural Networks (RNNs) are a special class of artificial neural networks [18,19]. Their feature is that they take into account the contextual dependence between elements of the sequence, that is, information about previous states to be taken into account in the current state.

In recurrent networks a directed sequence of elements is formed, presented in the form of a directed graph, and neurons are able to transmit it together with the processed data. This is achieved by using loops in each neuron.

## 1. Development of RNN

As already mentioned, the main feature of recurrent networks is the possibility of using pre-processed information for a better understanding of the existing information. This is a big step forward and it is actually effective. But does it always work? It all depends on how far down the sequence the context we need now is. If this "distance" is small, then everything really works well.

But it is quite clear that this is usually not the case and this "distance" can often be very large. At the same time, with its increase, the network at some point becomes unable to learn and combine the necessary information.

This problem is known as the gradient vanishing problem and is one of the main problems of recurrent neural networks.

Various modifications have been developed to overcome this problem. LSTM (Long Short-Term Memory) is one of those [20].

LSTM is an improvement of the recurrent neural network architecture that allows to study of long-term dependencies [21-23]. This allows such networks to cope with a large number of problems, thanks to which they have been widely used.

All recurrent neural networks have the form of a chain of repeated neural network modules. In standard RNNs, this chain has a simple structure, usually one layer with a certain activation function: hyperbolic tangent, sigmoid, etc.

LSTM also has a chain-like structure, but the module structure is different. Instead of one layer, there are four layers that interact with each other as shown in the figure 1.

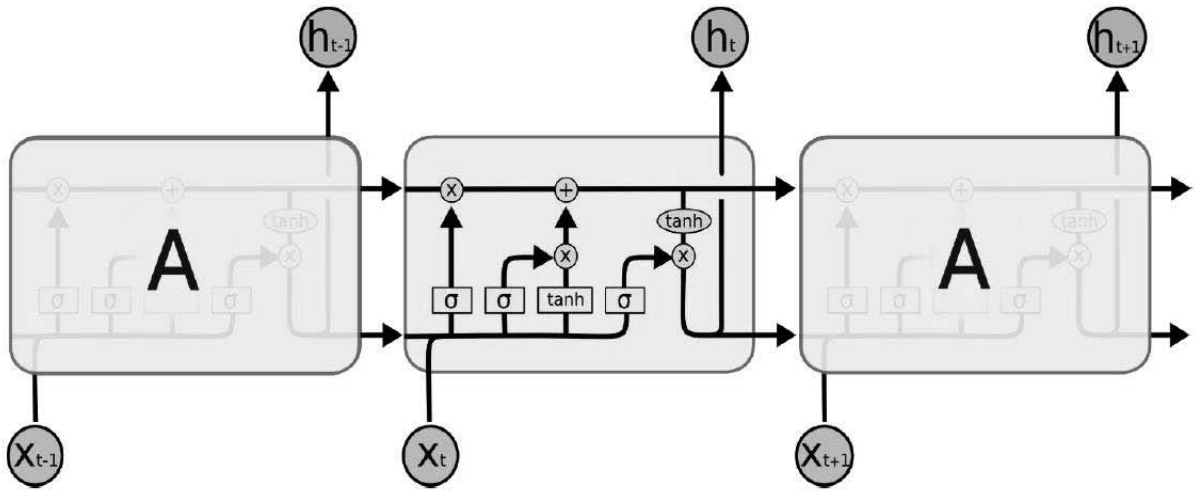


Figure 1. A recurrent LSTM module with four layers [24]

In the figure 1 each line carries a vector from the output of one node to the inputs of the others [24]. Pink circles represent element-by-element operations such as vector addition, and yellow rectangles represent trained neural network layers. Merged lines indicate concatenation, while split lines indicate copying of content, which is then sent to different locations.

The main idea behind LSTM is the cell state, the horizontal line that runs along the top of the chart.

The state of the cell is like a conveyor belt. It is simply laid along the entire chain with minor linear interactions. Information easily passes along the entire tape without changes.

But LSTMs have the ability to remove or add information to the state of a cell, which is controlled by structures called gates.

Filters are a way of controlling the transmission of information. They consist of a sigmoidal neural network layer and an element-by-element multiplication operation between the results obtained from the sigmoidal layer and the current state of the cell state.

The sigmoid layer outputs a number between zero and one, describing how much of each component should be transmitted. A

value of zero means "skip nothing" and a value of one means "skip everything".

LSTMs have three such filters to control the state of the cell.

The cell value is calculated in several steps. First, analyzing the information from the previous cell and deciding what can be removed from it. This decision is made by a sigmoidal layer called the "forgetting filter layer".

The next step is to decide what new information we will store in the cell state, and it consists of two parts. Firstly, the sigmoid layer, which called the "input filter layer", decides which values need to be updated. Secondly, the hyperbolic tangent (tanh) layer creates a vector of new values that can be added to the state.

After that updating the old state of cell $C_{t-1}$ and obtaining a new $C_t$ are performed. To do this, we multiply the old state by the value of $f_t$ (the "forgetting filter layer"), thus forgetting what was decided to be forgotten in the first step. Then we add the product of it and $C_t$. These are the new potential values multiplied by the "importance" factor, which is how much we decide to update each state value.

Finally, we need to decide what we're going to output. This will be based on our cell state, but should also be filtered. First, a sigmoid layer is applied, which decides which parts of the cell's state we pass next. Then we pass the cell state through the tanh layer so that the values are in the range of -1 to 1 and multiply by the output values of the other sigmoid layer to output only what is needed.

This is the basic idea of LSTM. These have the ability to remember long-term dependencies, delete and add information to the cell's state, and control what will go to the output.

## 2. Development of a dataset for the training of model

This section describes the data collection process for training the LTSM model.

Data collection and preparation is an important stage in the process of developing and training a neural network [25], because we need to prepare the data on which it will learn.

The online platform Kaggle can help with this. Here you can find a large collection of various datasets for any purpose. There is also a proprietary development environment that can also be used to build neural networks. At the same time, the platform unites a community of people engaged in machine learning. Here you can ask anything on this topic or discuss something. Various competitions are also often held.

A dataset named IMDB Dataset of 50K Movie Reviews was found in Kaggle [26]. This is a collection of 50,000 movie reviews, each of which is marked with a sentiment (positive or negative) as shown in the figure 2.

The dataset is well balanced, so we have an almost equal number of positive and negative reviews.

Since the bot, which is being developed, must communicate in Ukrainian, the dataset was translated using the googletrans library for Python, which provides access to Google Translate from a python code file. Of course, the translation cannot be called exact, but

| review | sentiment |
|---|---|
| A wonderful little production. <br /><br />The filming technique is very unassuming- very old-time- B... | positive |
| I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air con... | positive |
| Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his par... | negative |

Figure 2. Content of the dataset

translating 50 thousand reviews by hand would take too much time, so it was used.

The data also includes a list of stop words in the Ukrainian language. Fortunately, there are already ready-made lists, so one of these was used.

In addition, files were prepared with certain variants of appeals to the bot and probable reactions to such appeals. Scripts include greetings, farewells, thanks, and questions. And separately, of course, there is all the other text, which can be positive, negative or neutral, depending on which the bot will have a different reaction.

## 3. Description of application implementation

The developed application, with the help of libraries [27,28], consisted of three Google Colab[29] notebooks and three Python files.

The first notebook is used to translate the dataset into Ukrainian.

The second notebook is the main neural network, which is supposed to evaluate the emotional coloring of messages. First, the data is obtained from the dataset, the entire text is divided into training and test data, then the text is processed (removal of redundant characters using regular expressions and lemmatization using the pymorphy2 library).

The neural network is not able to work with the text, so the text is further tokenized. For this, all unique words that occur in the dataset are determined. A dictionary is formed from them, in which each word is given a unique token. This allows us to further represent the text as a list of tokens, where each token replaces a specific word. The dictionary is saved in a text file for later use. The data is also converted to the most suitable data format.

After that, the SentimentRNN neural network itself is created and trained on already processed data . Its structure is shown in Figure 3.

```
SentimentRNN(
  (embedding): Embedding(2001, 64)
  (lstm): LSTM(64, 256, num_layers=2, batch_first=True)
  (dropout): Dropout(p=0.3, inplace=False)
  (fc): Linear(in_features=256, out_features=1, bias=True)
  (sig): Sigmoid()
)
```

Figure 3. Structure of the SentimentRNN model

During training, a loss value is calculated for each epoch, and if it is less than the previous epochs, then the current state of the neural network is considered the best. The best state among all eras is saved to a file for later use.

The third notebook contains another neural network. Its task is to determine the intended scenarios in the user's messages, such as greetings, farewells, etc. In the data file for each scenario, certain patterns are defined, as well as responses to them.

Word processing is similar to the previous notebook, but word lemmatization is not used here.

The structure is much simpler as can be seen in Figure 4.

```
NeuralNet(
  (l1): Linear(in_features=30, out_features=8, bias=True)
  (l2): Linear(in_features=8, out_features=8, bias=True)
  (l3): Linear(in_features=8, out_features=4, bias=True)
  (relu): ReLU()
)
```

Figure 4. Structure of the NeuralNet model

Since it is not possible to create applications with a graphical interface in Google Colab, this part of the project was developed in Python IDLE, so we have three more files with the extension .py.

The first file contains the application interface, and the second and third are used to load the two models trained in Google Colab[29] and saved to files, and also contain functions for these models to predict the results.

## 4. Results of work and testing

If you look at the graphs of training and validation accuracies and losses, you can see that the results with the training data were constantly improving, while the best result on the validation data was achieved by the model at epoch 3, after which the losses only increased. This is indicative of overtraining and is clearly worth fixing, however 85% accuracy is a pretty good result, so the decision was made to leave this trained neural network as it is for epoch 3.

Then this is used in chatbot. An example of the conversation with it can be seen in figure 5.

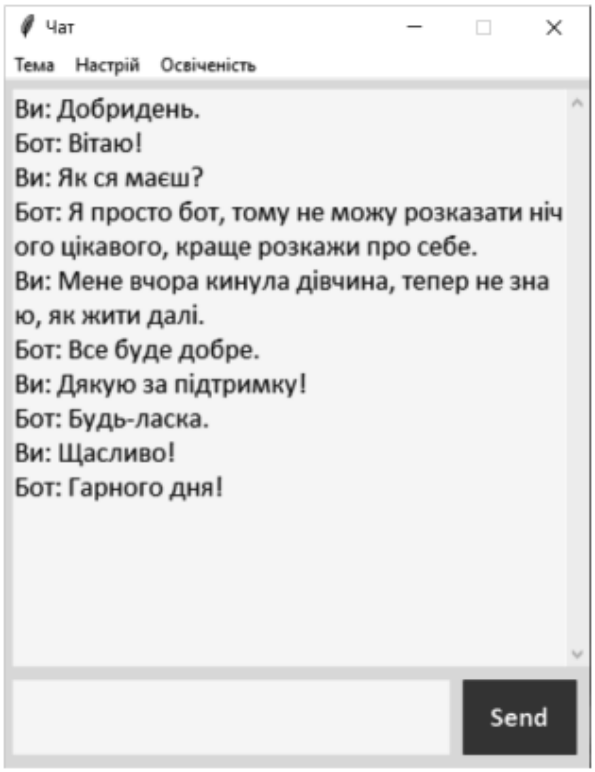


Figure 5. An example of a conversation with a developed chat bot

As you can see, the bot can recognize such common scenarios as greetings, thanks and goodbyes. When the user asks the bot about bot itself, it explains that it is an ordinary artificial intelligence and offers the user to tell what is on his mind.

The following is an example of what an average user could write to a bot. In this case, girl broke up with him, and we see that he is upset. And the bot, or rather the neural network behind it, is also able to recognize negative emotions, so it encourages the unhappy user in response.

## Conclusions

In this article, a recurrent neural network with modification named long short-term memory was developed to analyze sentiments in messages.

A large dataset, 50 thousand comments, containing different reviews and their sentiments was collected and annotated to successfully train and validate the model. It was also translated into Ukrainian language.

The resulting model demonstrated accuracy 85% in determining sentiments. One more small model was developed to improve make chatbot recognise the intended scenarios in the user's messages.

And finally, chatbot application designed like an ordinary chat was created. There user can communicate with chatbot which uses pre-trained models to understand how user feels and responds reespectively. Also it has some customization such as changing design and bot behavior.

Further improvement of the model's classification quality can be achieved by collecting a larger and better organised dataset or by researching other RNN architectures.

## References

1. Rodríguez-Ibánez, M., Casánez-Ventura, A., Castejón-Mateos, F., & Cuenca-Jiménez, P. M. (2023). A review on sentiment analysis from social media platforms. Expert Systems with Applications, 119862. https://doi.org/10.1016/j.eswa.2023.119862
2. Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis: tasks, approaches and applications. Knowledge-based systems. Vol. 89, pp. 14-46. https://doi.org/10.1016/j.knosys.2015.06.015
3. Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowledge-Based Systems. Vol. 226, 107134. https://doi.org/10.1016/j.knosys.2021.107134
4. Yadav, A., & Vishwakarma, D. K. (2020). Sentiment analysis using deep learning architectures: a review. Artificial Intelligence Review. Vol. 53(6), pp. 4335-4385. https://doi.org/10.1007/s10462-019-09794-5
5. Do, H. H., Prasad, P. W., Maag, A., & Alsadoon, A. (2019). Deep learning for aspect-based sentiment analysis: a comparative review. Expert systems with applications. Vol. 118, pp. 272-299. https://doi.org/10.1016/j.eswa.2018.10.003
6. Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science. Vol. 2(6), 420. https://doi.org/10.1007/s42979-021-00815-1
7. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), pp. 436-444. https://doi.org/10.1038/nature14539
8. Shymkovych V., Telenyk S., Kravets P. (2021) Hardware implementation of radial-

basis neural networks with Gaussian activation functions on FPGA. Neural Computing and Applications. vol. 33, no.15, pp. 9467-9479. https://doi.org/10.1007/s00521-021-05706-3

9. Harumy, T. F., Zarlis, M., Effendi, S., & Lidya, M. S. (2021, August). Prediction using a neural network algorithm approach (a review). 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM), Pekan, Malaysia, IEEE, pp. 325-330. https://doi.org/10.1109/ICSECS52883.2021.00066
10. Shymkovych, Volodymyr, Anatoliy Doroshenko, Tural Mamedov, and Olena Yatsenko (2022) Automated Design of an Artificial Neuron for Field-Programmable Gate Arrays Based on an Algebra-Algorithmic Approach. International Scientific Technical Journal "Problems of Control and Informatics" vol. 67, no. 5, pp. 61-72. https://doi.org/10.34229/2786-6505-2022-5-6
11. Perera, N. N., & Ganegoda, G. U. (2023). A Comprehensive Review on Speech Synthesis Using Neural-Network Based Approaches. 2023 3rd International Conference on Advanced Research in Computing (ICARC), Belihuloya, Sri Lanka, IEEE, pp. 214-219 https://doi.org/10.1109/ICARC57651.2023.10145741
12. Bezliudnyi Y., Shymkovysh V., Doroshenko A.( 2021) Convolutional neural network model and software for classification of typical pests. Prombles in programming. Vol.4, pp. 95-102. https://doi.org/10.15407/pp2021.04.095
13. Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. Multimedia tools and applications. Vol. 82(3), pp. 3713-3744. https://doi.org/10.1007/s11042-022-13428-4
14. Kravets P., Nevolko P., Shymkovych V., Shymkovych L. (2020) Synthesis of High-Speed Neuro-Fuzzy-Controllers Based on FPGA. 2020 IEEE 2nd International Conference on Advanced Trends in Information Theory (ATIT). pp. 291-295. https://doi.org/10.1109/ATIT50783.2020.9349299
15. Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications. Vol. 6, 100134. https://doi.org/10.1016/j.mlwa.2021.100134
16. Kravets, P., Novatskyi, A., Shymkovych, V., Rudakova, A., Lebedenko, Y., Rudakova, H. Neural Network Model for Laboratory Stand Control System Controller with Parallel Mechanisms. Lecture Notes on Data Engineering and Communications Technologies. Springer, Cham. 2023. Vol 181. pp. 47-58 https://doi.org/10.1007/978-3-031-36118-0_5
17. Y.S. Hryhorenko, V.M. Shymkovysh, P.I. Kravets, A.O. Novatskyi, L.L. Shymkovysh, A.Yu. Doroshenko. A convolutional neural network model and software for the classification of the presence of a medical mask on the human face. Problems in programming. 2023. Vol. 2. pp. 59-66. https://doi.org/10.15407/pp2023.02.059
18. Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation. Vol. 31(7), pp. 1235-1270. https://doi.org/10.1162/neco_a_01199
19. Bezliudnyi Y., Shymkovych V., Kravets P., Novatsky A., Shymkovych L. Pro-russian propaganda recognition and analytics system based on text classification model and statistical data processing methods. Адаптивні системи автоматичного управління: міжвідомчий науково-технічний збірник. 2023. № 1 (42), с. 15-31. https://doi.org/10.20535/1560-8956.42.2023.278923
20. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation. Vol. 9(8), pp. 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
21. Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. Artificial Intelligence Review. Vol. 53, pp. 5929-5955 https://doi.org/10.1007/s10462-020-09838-1
22. Kader, N.I.A., Yusof, U.K., Khalid, M.N.A., Husain, N.R.N. (2023). A Review of Long Short-Term Memory Approach for Time Series Analysis and Forecasting. Lecture Notes in Networks and Systems. Vol 573. Springer, Cham. pp. 12-21 https://doi.org/10.1007/978-3-031-20429-6_2
23. Long, F., Zhou, K., & Ou, W. (2019). Sentiment analysis of text based on bidirectional LSTM with multi-head attention. IEEE Access,7, pp. 141960-141969. https://doi.org/10.1109/ACCESS.2019.2942614

24. Olah, C. (2015). Understanding LSTM networks. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
25. R. Yu, S. Liu, X. Wang (2024) Dataset Distillation: A Comprehensive Review. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 46, no. 1, pp. 150-170. https://doi.org/10.1109/TPAMI.2023.3323376
26. IMDB Dataset of 50K Movie Reviews URL: https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
27. TensorFlow. (n.d.). TensorFlow: An end-to-end open source machine learning platform. Retrieved from https://www.tensorflow.org/
28. Keras. (n.d.). Keras: The Python deep learning API. Retrieved from https://keras.io/
29. Google Colab https://colab.research.google.com/

Одержано: 16.01.2024

***Про авторів:***

Кобченко Владислав Русланович,
студент магістр НТУ України
«КПІ імені Ігоря Сікорського»

Шимкович Володимир Миколайович,
кандидат технічних наук, доцент НТУ України «КПІ імені Ігоря Сікорського».
Кількість наукових публікацій
в українських виданнях – понад 30.
Кількість наукових публікацій
в зарубіжних виданнях – понад 10.
Індекс Хірша – 4. https://orcid.org/0000-0003-4014-2786

Новацький Анатолій Олександрович,
кандидат технічних наук,
доцент НТУ України
«КПІ імені Ігоря Сікорського».
Кількість наукових публікацій
в українських виданнях – понад 30

Кравець Петро Іванович,
кандидат технічних наук, доцент НТУ України «КПІ імені Ігоря Сікорського».
Кількість наукових публікацій
в українських виданнях – понад 40.
Кількість наукових публікацій
в зарубіжних виданнях – понад 10.
Індекс Хірша – 4. https://orcid.org/0000-0003-4632-9832

Шимкович Любов Леонідівна,
асистент кафедри інформаційних систем та технологій НТУ України
«КПІ імені Ігоря Сікорського».
Кількість наукових публікацій
в українських виданнях – 4.
Кількість наукових публікацій
в зарубіжних виданнях – 1.
https://orcid.org/0000-0002-1291-0373

Дорошенко Анатолій Юхимович,
доктор фізико-математичних наук,
професор, професор НТУ України
«КПІ імені Ігоря Сікорського».
Кількість наукових публікацій
в українських виданнях – понад 200.
Кількість наукових публікацій
в зарубіжних виданнях – понад 90.
Індекс Хірша – 7. http://orcid.org/0000-0002-8435-1451

***Місце роботи авторів:***

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», проспект Перемоги 37 та Інститут програмних систем НАН України, 03187, м. Київ-187, проспект Академіка Глушкова, 40.
E-mail:
vladik020402@gmail.com,
v.shymkovych@kpi.ua,
a.novatskyi@kpi.ua,
peter_kravets@yahoo.com,
L.shymkovych@gmail.com,
doroshenkoanatoliy2@gmail.com