https://orcid.org/0000-0002-9976-0266
https://orcid.org/0000-0002-6209-5661
https://orcid.org/0000-0002-0884-6516

UDC 681.518.5

**O.O. DRUZHYNIN[*], V.V. NEKHAI[*], O.A. PRILA[*]**

# FACEBOOK TEXT POSTS CLASSIFICATION WITH TENSORFLOR

[*]Chernihiv National University of Technology, Chernihiv, Ukraine

***Анотація.*** *Сьогодні однією з основних тенденцій розвитку Інтернету є зростаюча популярність соціальних мереж. Все більше людей «створюють зв'язки» один з одним, і в результаті кількість користувачів соціальних мереж зростає в геометричній прогресії. Тепер важко знайти людину, яка не має власних Facebook, Twitter чи Instagram сторінок. Люди все частіше взаємодіють у віртуальному просторі через різні канали комунікації і соціальні мережі – один з найпопулярніших видів. У цих умовах використання соціальних мереж як джерела інформації для запобігання кібератакам чи визначення ступеня загрози (настроїв, відношення) щодо досліджуваного об'єкта стає дуже актуальним. Більшість людей не уявляють свого життя без соціальних мереж: ми переглядаємо останні новини, дізнаємося про останні інновації на ринку, читаємо навчальні статті, ділимося інформацією з нашими друзями, стежимо за останніми подіями, що відбулися в їхньому житті, і т.д. Обробка природних мов (ОПМ) є однією з найважливіших технологій XXI століття. Машинне розуміння є дуже цікавим, але складним завданням як в обробці природних мов (ОПМ), так і в дослідженні штучного інтелекту (ШІ). ОПМ можна застосовувати там, де потрібна взаємодія людини з машиною (людино-машинна взаємодія). Останнім часом глибокі методи навчання показують вражаючі результати в вирішенні завдань, що стосуються ОПМ. Стандартні моделі глибокого навчання часто можуть використовуватися для вирішення цілого ряду завдань без необхідності застосування традиційних аналітичних інженерних методів, що потребують надзвичайно багато ресурсів. У цій статті ми розглянемо завдання класифікації текстів по відношенню до досліджуваного об'єкта за допомогою фреймворка «TensorFlow».*

***Ключові слова:*** *машинне навчання, глибоке навчання, обробка природних мов, інтелектуальний аналіз тексту, TensorFlow.*

***Аннотация.*** *Сегодня одной из основных тенденций развития Интернета является растущая популярность социальных сетей. Все больше людей «создают связи» друг с другом, и в результате количество пользователей социальных сетей растет в геометрической прогрессии. Сейчас трудно найти человека, который не имеет собственных Facebook, Twitter или Instagram страниц. Люди все чаще взаимодействуют в виртуальном пространстве через различные каналы коммуникации и социальные сети – один из самых популярных видов. В этих условиях использование социальных сетей как источника информации для предотвращения кибератакам или определения степени угрозы (настроений, отношений) по поводу исследуемого объекта становится очень актуальным. Большинство людей не представляют своей жизни без социальных сетей: мы просматриваем последние новости, узнаем о последних инновациях на рынке, читаем учебные статьи, делимся информацией с нашими друзьями, следим за последними событиями, произошедшими в их жизни, и т.д. Обработка естественных языков (ОЕЯ) является одной из важнейших технологий XXI века. Машинное понимание очень интересное, но сложное задание как в обработке естественных языков (ОЕЯ), так и в исследовании искусственного интеллекта (ИИ). ОЕЯ можно применять там, где требуется взаимодействие человека с машиной (человеко-компьютерное взаимодействие). В последнее время глубокие методы обучения показывают впечатляющие результаты в решении задач, касающихся ОЕЯ. Стандартные модели глубокого обучения часто могут использоваться для решения целого ряда задач, без необходимости применения традиционных аналитических инженерных методов, требующих очень много ресурсов. В этой статье мы рассмотрим задачи классификации текстов по отношению к исследуемому объекту с помощью фреймворка «TensorFlow».*

***Ключевые слова:*** *машинное обучение, глубокое обучение, обработка естественных языков, интеллектуальный анализ текста, TensorFlow.*

***Abstract.*** *Today, one of the main trends in the development of the Internet is the growing popularity of social networks. More and more people «create ties» with each other, and as a result, the number of users* *of social networks grows in geometric progression. Now, it's hard to find a person who does not have his own page on Facebook, Twitter or Instagram. Increasingly, people interact in virtual space due to different circumstances. In these circumstances, the use of social networks as a source of information, in addition, to prevent cyber-attacks or to determine the degree of threat (sentiment, attitude) towards the object under study is becoming very relevant. Many people do not imagine their lives without social networks: there we review the latest news, learn about the latest innovations in the market, read educational articles, share information with our friends, follow the latest events that have occurred in their lives, etc. Natural language processing (NLP) is one of the most important technologies of the XXI century. Machine Comprehension is a very interesting but challenging task in both Natural Language Processing (NLP) and artificial intelligent (AI) research. NLP can be applied wherever human-machine interaction is needed. Recently, deep learning methods show good results in tasks involving NLP. Standard models can often be used to solve a range of tasks, without the need to apply traditional analytical engineering techniques. The widespread distribution of social networks and the large number of users could give us impressive results, which can further build system interests analysis with a large number of established trust relationships. In this article, we will consider the task of classifying texts in relation to the object under study using the TensorFlow framework.*

***Keywords:*** *Machine Learning, Deep Learning, Natural Language Processing, Text Mining, TensorFlow.*

DOI: 10.34121/1028-9763-2019-3-47-54

## 1. Introduction

We are at an interesting stage in the development of artificial intelligence. The years of research in this area have yielded tangible results, in particular in the field of in-depth learning.

Today, the popularity of social networks (Facebook, Instagram, Twitter) and messengers (What’s Up, Facebook messenger, Viber, Telegram) remains high and still shows a positive user boost [1]. The popularity of such means of communication between people is growing at a rather high pace, and this is not surprising, since people are able to negotiate more easily and quickly with meetings, exchange news, discuss and solve problems, develop business, etc. Thus, the time consuming time for communication in Internet increases [1].

Cybercriminals are also humans, therefore, they often use social networks to coordinate their actions – because it is easier to find people who can support their ideas, and even take an active part in it. Some even have official pages on social networks, where such groups share the latest achievements and successes in cyberspace. From such correspondence (posts) it is easy for a person to understand who sent a post and the relation of the contributor to the organization or person. Researchers who try to track the mood of other people in social networks need a lot of time to read such posts and arrive at a final conclusion. Moreover, many of them (posts) have nothing to do with the topic of the researcher, and therefore he spends a lot of time on filtering just the right ones. There is a problem finding relevant information that can be used to assess the attitude of people to the organization or groups of other people. Coordination takes place in the form of messages in social networks. Of course, many of these messages are available only to a close circle of people, but it happens otherwise – the discussion is conducted in an open form in order to attract as large an audience as possible.

There is a problem of automatic processing of such messages. This is precisely one of the branches of artificial intelligence research – NLP (Natural Language Processing) – the general direction of artificial intelligence and mathematical linguistics. This study focuses on the problems of natural languages computer analysis and synthesis. In the case of artificial intelligence, analysis means understanding the language, and synthesis generates the correct text. Solving these problems will mean creating a more convenient form of interaction between the computer and the person.

## 2. Deep learning popularity

At the beginning, NLP problems were solved by systems that were built on the basis of rule systems [2]. For example, by writing grammar rules or developing heuristic rules to determine the root of a word, etc. However, such decisions were unsustainable and did not solve the problem as it was required, and support for such systems is very complicated, since natural languages are not fully described by the rules (needing a further component of awareness) and always have exceptions [3].

Recent research [4] which is published by O'Reilly Media shows that companies are already interested in solving a wide range of problems using Machine Learning and Deep Learning approaches.

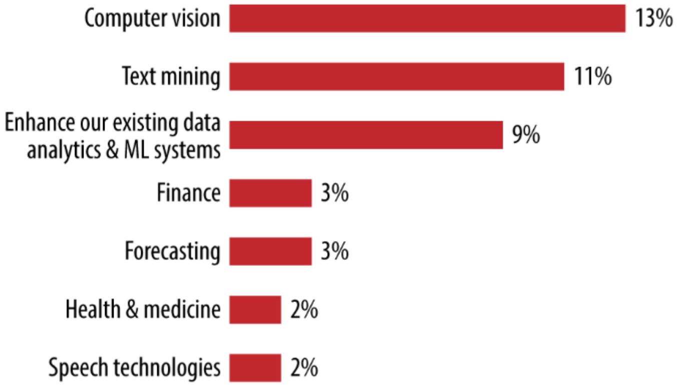


Figure 1 – What application of deep learning are you interested in? [4]

Figure 1 shows that Text mining is second only to the applications in which interested companies surveyed, lagging behind only 2% of Computer vision. This may perhaps be explained by greater awareness of Computer vision solutions, unlike Text mining. Already there are cars with autopilot, which are issued for the mass market, phones capable of recognizing their owners, etc.

In accordance with Figure 2 – TensorFlow is the most popular among open source frameworks with Keras in second place. All the listed tools have appended not so long ago: Caffe in 2014, 2015 – Torch and TensorFlow in 2016.

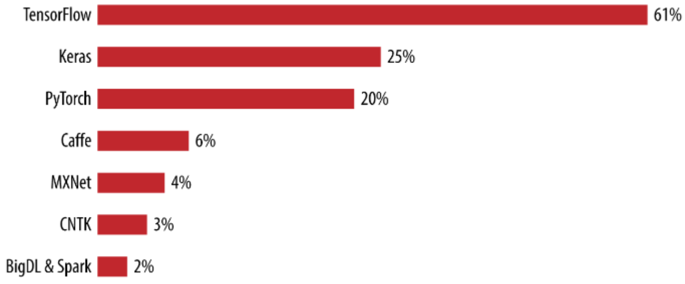


Figure 2 – What deep learning frameworks or tools are you using? [4]

In early 2017, two new frameworks were released. PyTorch quickly attracted attention from researchers and teachers; BigDL applies deep learning on Spark clusters. Also, Amazon and Microsoft are collaborating to create new tools that will be easier to use. Most likely, such frameworks like TensorFlow, Keras, and PyTorch remain popular for some time, others, like MXNet, CNTK, and BigDL, is getting their popularity day by day.

## 3. The aim of research

As a component of the expert analysis system for assessing user sentiment in relation to the subject under study, a solution to the problem is considered using a solution based on Deep Learning approaches. Since the user's attitudes on the Internet can testify to the relation of cyberspace to the investigated object, it is considered expedient to automate the process of gathering and processing information to obtain a general analytical picture. So, the purpose of the study is to explore the possibility of using deep learning models to determine the mood of cyberspace.

## 4. Related works

The basic concepts of Deep Learning, discussed in [5] which provides a lot of information for thinking about solving business problems in their future. In the article [5] it is noted that “Deep learning so far is difficult to engineer with”, it is still relevant now, although the entry threshold into the ML sphere is falls due to new frameworks that make it easier to test their abilities in this area.

Neural networks use mostly mathematical concepts and models, so it's hard to understand, at least at the basic level, why these models work. Article [6] considers the use of various neural networks, and algorithms of optimization with an emphasis on the mathematical concepts that are hidden behind them.

Other traditional NLP tasks such as: part-of-speech tagging, chunking, named entity recognition, and semantic role labeling are discussed in [7], which uses similar principles for models construction. [8] offers modified Convolutional Networks for solving tasks related to the classification of texts. The authors increased the number of layers in such neural networks up to 29. Such neural networks, according to the authors, can solve NLP problems more accurately. An alternative approach can be found in [9], which proposes to solve the problem of classifying texts using 2 Convolutional Networks, one of which has 9 layers, the other – 6 convolutional and 3 fully-connected layers.

Another paper [10] discusses whether to use the TensorFlow framework. It is being compared with other frameworks (Caffe, CNTK, Theano, Torch) to solve various tasks.

In [11], the framework for TensorFlow is considered in detail, and is an example of how appropriate it is to be used in scientific research.

In [12], I/O performance studies were performed for TensorFlow, and [13] focuses on the optimal use of CPU resources during research and in the production phase of the model.

Due to the fact that a large amount of data is often needed for constructing models that will produce a good result, there is also the problem of the rapid processing of such a large amount of input data, so [14] proposes to use the traditional distributed computing model in projects using TensorFlow, on based on the open source framework Horovod.

## 5. Research methods and data collection

The most difficult and meaningful way in working with text information is the need to teach the machine to understand the meaning of sentences and their relationships. But for this, we must first consider the task of understanding words and phrases.

The traditional approach to NLP involves a lot of domain knowledge of linguistics itself. Understanding terms such as phonemes and morphemes were pretty standard, as there are whole linguistic classes dedicated to their study. Traditional NLP approaches in order to understand the word use approaches to break the word into parts-prefix, root suffix, etc. Table 1 shows a list of suffixes in English for nouns.

Table 1 – Noun suffixes [15]

| suffix | meaning | example |
|---|---|---|
| -acy | state or quality | democracy, accuracy, lunacy |
| -al | the action or process of | remedial, denial, trial, criminal |
| -ance,<br>-ence | state or quality of | nuisance, ambience, tolerance |
| -dom | place or state of being | freedom, stardom, boredom |
| -er,<br>-or | person or object that does a specified action | reader, creator, interpreter, inventor, collaborator, teacher |
| -ism | doctrine, belief | Judaism, skepticism, escapism |
| -ist | person or object that does a specified action | Geologist, protagonist, sexist, scientist, theorist, communist |

Continuation of table 1

| | | |
|---|---|---|
| -ity,<br>-ty | quality of | extremity, validity, enormity |
| -ment | condition | enchantment, argument |
| -ness | state of being | heaviness, highness, sickness |
| -ship | position held | friendship, hardship, internship |
| -sion,<br>-tion | state of being | position, promotion, cohesion |
| -ity,<br>-ty | quality of | extremity, validity, enormity |
| -ment | condition | enchantment, argument |
| -ness | state of being | heaviness, highness, sickness |
| -ship | position held | friendship, hardship, internship |
| -sion,<br>-tion | state of being | position, promotion, cohesion |

It has been estimated that the vocabulary of English includes roughly 1 million words, but if we count all the forms (participles and plural) of the words we will end up around to 13 million number.

Words must be presented in a vector form so that it is possible to apply algorithms of machine learning. Such coding should take into account the interrelations between words – in this way, words that have a similar meaning should be closer to one another than words having different meanings. That is, the concept of the distance between vectors that can be calculated using the Euclidean distance or the cosine similarity becomes a key to understanding the machine of interconnections between words.

## 6. Word2Vec

Word embedding is a feature learning technique in which each word or phrase from the vocabulary is mapped to a N dimension vector of real numbers. A word vector, by itself, is a row of numbers. Each this number captures a dimension of the word's meaning and where semantically similar words have similar vectors. This eventually leads us to the fact that words such as «apple» and «orange» should have similar word vectors to the word «fruit» (because of the similarity of their meanings), whereas the word «engine» should be quite distant (in terms of meaning and vector distance). Put differently, words that are used in a similar context will be mapped to a proximate vector space.

The Word2Vec uses shallow NN with two hidden layers: continuous bag of words (CBOW) and the Skip-Gram model in order to create a vector for each given word. The Skip-Gram model dives a corpus of words $\omega$ and context C [16]. The goal is to maximize the probability:

$$\operatorname{argmax}_{\theta} \Pi_{\omega \in \mathrm{T}} [\Pi_{C \in C(\omega)} \rho(c \mid \omega; \theta)] . \tag{1}$$

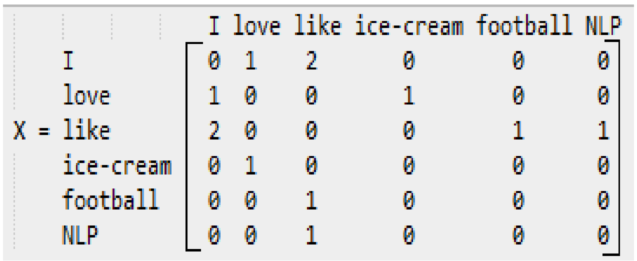
Where T refers to Text, and $\theta$ is parameter of $\rho(c \mid \omega; \theta)$.

In Skip-gram model, we take a center word and a window of context words and we try to predict context words out to some window size for each center word. So, our model is going to define a probability distribution i.e. probability of a word appearing in context given a center word and we are going to choose our vector representations to maximize the probability.

Figure 5 shows a simple CBOW model which tries to find the word based on previous words (by summing vectors) while Skip-Gram tries to find words that might come in the vicinity of each word.

The weights between the input layer and output layer represent $v \times N$ [17] as a matrix of $w$.

$$h = W^T c = W^T_{k_r} := \sigma^T_{\omega I} . \qquad (2)$$



$$X = \begin{array}{c|cccccc} & \text{I} & \text{love} & \text{like} & \text{ice-cream} & \text{football} & \text{NLP} \\ \hline \text{I} & 0 & 1 & 2 & 0 & 0 & 0 \\ \text{love} & 1 & 0 & 0 & 1 & 0 & 0 \\ \text{like} & 2 & 0 & 0 & 0 & 1 & 1 \\ \text{ice-cream} & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{football} & 0 & 0 & 1 & 0 & 0 & 0 \\ \text{NLP} & 0 & 0 & 1 & 0 & 0 & 0 \end{array}$$

Figure 3 – Co-ocurence matrix

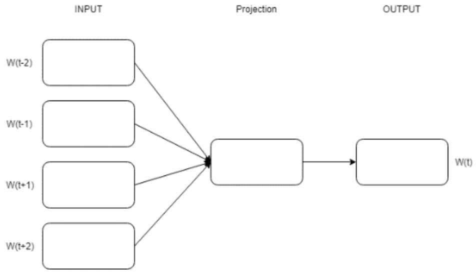


Figure 4 – Skip-gram predicts surrounding words given the word

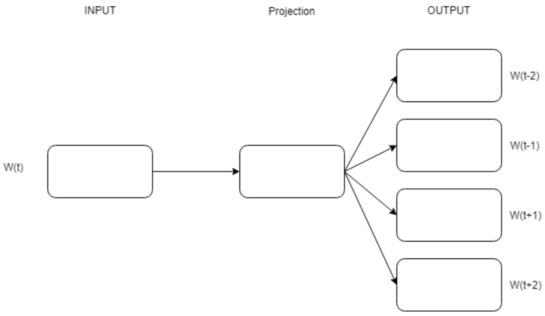


Figure 5 – CBOW predicts the current word based on the context

Our predictive model learns the vectors by minimizing the loss function. In Word2vec, this happens with a feed-forward neural network and optimization techniques such as Stochastic gradient descent. We have a large matrix with each column for the «context» and row for the «words». The number of «contexts» is of course large, since it is essentially combinatorial in size. SVD is applied to reduce the dimensions of the matrix retaining maximum information.

Figure 3 shows how the co-ocurence matrix will look like for the example below.

1. I love ice-cream.
2. I like football.
3. I like NLP.

In summary, converting words into vectors, which deep learning algorithms can ingest and process, helps to formulate a much better understanding of natural language.

## 7. Experiments

TF-Hub text embedding module for TensorFlow was used to train a simple sentiment classifier with a reasonable baseline accuracy.

We have our own dataset which consists of Facebook user’s posts that have been collected for 2 month using Graph API provided by Facebook. Each message has its own rank that indicate its relevance to the healthy lifestyle and sports activity. All the messages that were collected were manually ranked and those that had no relevance or against to the sports and healthy lifestyle or for it. At the end we end up in 50k messages. The resulting dataset were split into 2 halves – 25k each. Each of them contain 10k positive messages and 15k negative ones. In order to avoid unfair evaluation and training, all these messages were shuffled before training and evaluation stages. The first part of the dataset is used for training purposes while the other one – for model evaluation. All the messages were preprocessed to word2vec format, in order to use them for model training.

The model was trained on machine with Intel CPU i7 2630QM 2.0Ghz and 16G RAM.

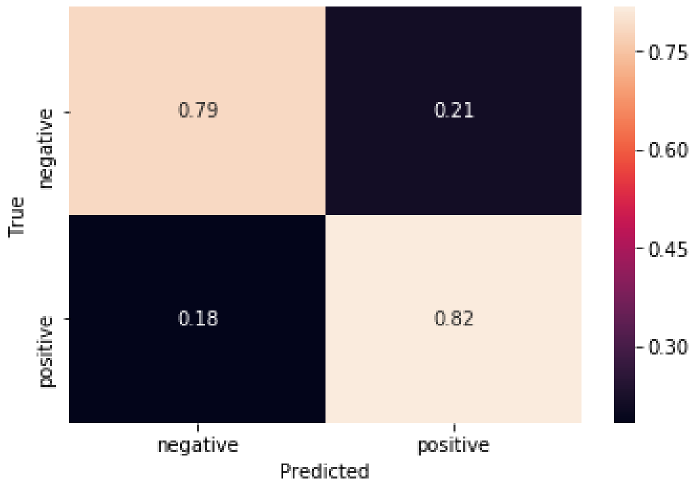


Figure 6 – Result confusion matrix

As for the classifier, DNN Classifier with loss reduction calculated by using Softmax cross entropy function was used.

After training the DNN Classifier we got the following accuracy of predictions:

Training set accuracy: 0.8018800020217896.

Test set accuracy: 0.7921199798583984.

The final classification results are shown on the figure 6 that shows the confusion matrix which can visualize the distributions o misclassifications that we have in our model.

As the result, we can see that the vast majority of the messages were correctly classified: positive messages have only 16 present of misclassification while negative ones – 21 present. We can see that our model has shown quite good results – in average 80 present of the messages is classified correctly.

## 8. Conclusions and perspectives

The classification task is one of the most indispensable problems in the machine learning. As text and document datasets proliferate, the development and documentation of supervised machine learning algorithms becomes an imperative issue, especially for text classification. Having a better document categorization system for this information requires algorithms. However, the existing text classification algorithms work more efficiently if we have a better understanding of feature extraction methods and how to evaluate them correctly.

The results of the study show that the use of the TensorFlow framework takes less time to solve the problem of ranking posts in the social network Facebook with sufficient accuracy of 80 percent. To improve the accuracy of the created model, the number of epochs during its training can be increased. Another more complex option is to use more complex models, such as [8] and [9] or more advanced techniques for classification which is described in [18].

## REFERENCES

1. Kept S. We Are Social. 2019. URL: https://wearesocial.com/global-digital-report-2019.

2. Kang N. Using rule-based natural language processing to improve disease normalization in biomedical text. *J Am Med Inform Assoc*. 2013. N 20. P. 876–881.

3. Cole T. Comprehensive List of Rules and Exceptions. The Article Book. The University of Michigan Press, 2000. P. 109–113.

4. Lorica B., Loukides M. How Companies Are Putting AI to Work Through Deep Learning. O'Reilly *Media Inc*. Gravenstein Highway North, Sebastopol, 2018. P. 4–20.

5. Marcus G. Deep Learning: A Critical Appraisal. *arXiv preprint arXiv:1801.00631*. 2018. P. 2–6.

6. Goldberg Y. A Primer on Neural Network Models for Natural Language Processing. *arXiv preprint arXiv:1510.00726*. 2015. P. 350–406.

7. Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., Kuksa P. Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2494–2524.

8. Conneau A., Schwenk H., Cun Y. Very Deep Convolutional Networks for Text Classification. *arXiv preprint arXiv:1606.01781*. 2017. P. 3–8.
9. Zhang X., Zhao J., Cun Y. Character-level Convolutional Networks for Text Classification. *Courant Institute of Mathematical Sciences*. New York University, 2016. P. 2–8.
10. Schrimpf M. Should I use TensorFlow? An evaluation of TensorFlow and its potential to replace pure Python implementations in Machine Learning. *Augsburg University*. 2016. P. 2–11.
11. Goldsborough P. A Tour of TensorFlow. *arXiv preprint arXiv:1610.01178*. 2016. P. 1–14.
12. Steven W.D. Chien, Stefano M., Chaitanya P.S., Santos L., Herman P., Narasimhamurthy S., Laure E. Characterizing Deep-Learning I/O Workloads in TensorFlow. *arXiv preprint arXiv:1810.03035*. 2018. P. 2–10.
13. Hasabnis N., Clara S. Auto-tuning TensorFlow Threading Model for CPU Backend. *arXiv preprint arXiv:1812.01665*. 2018. P. 1–9.
14. Sergeev A., Balso D.M. Horovod: fast and easy distributed deep learning in TensorFlow. *arXiv preprint arXiv:1802.05799*. 2018. P. 2–9.
15. Manik J. Dictionary of prefixes & suffixes. 2014. Vol. 5. P. 6–86.
16. Goldberg Y., Levy O. Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*. 2014. P. 1–5.
17. Rong X. Word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*. 2014. P. 4–6.
18. Kowsari K, Meimandi J. K., Heidarysafa M., Mendu S., Barnes E. L., Brown E.D. Text Classification Algorithms: A Survey. *arXiv preprint arXiv:1904.08067*. 2019. P. 2–58.

*Стаття надійшла до редакції 18.06.2019*