UDC 004.85 **doi: 10.32620/reks.2024.3.05**

Olesia BARKOVSKA[1], Anton HAVRASHENKO[1], Vitalii SERDECHNYI[1], Vladyslav KHOLIEV[1], Patrik RUSNAK[2]

[1] ***Kharkiv National University of Radio Electronics, Kharkiv, Ukraine***
[2] ***University of Žilina, Zilina, Slovakia***

ANALYSIS OF THE IMPACT OF THE CONTEXTUAL EMBEDDINGS USAGE ON THE TEXT CLASSIFICATION ACCURACY

This study aims to improve the accuracy of text classification, which is critical in fields such as medical diagnostics and law. In addition, accuracy requirements are increasing constantly with the development of information technologies and increasing volume of text data. ***The subject*** *of this paper is to study the impact of text vectorization methods on the accuracy of text data classification.* ***The goal*** *of this paper is to evaluate the effectiveness of different word vectorization methods (Word2Vec, GloVe, BERT, and GPT) in the context of text classification based on different embedding strategies - Word and Contextual Embedding. The primary focus was on studying the effect of the number of training epochs (by systematically increasing the number of training epochs) on text classification accuracy.* ***The task*** *of this study was to systematically compare the effectiveness of each type of embedding following the formed matrix of experiments, which controls the equality of the conditions of the experiment, and to further evaluate the key metrics of text classification (on the example of the IMDB dataset) using a neural network classifier (LSTM) with a recurrent architecture. Machine learning* ***methods****, including neural network methods, methods of vector representation of words, and statistical analysis, were used in this study.* ***The results*** *demonstrate that the best Word Embedding model was GloVe, which demonstrated a final accuracy of 87.73%. In the context of Contextual Embedding, BERT proved to be more effective than GPT, with a final accuracy of 92.97% compared to 91.65% of GPT. In general, the results demonstrate the superiority of Contextual Embedding in natural language processing tasks and confirm its potential use in modern applications and text analysis systems.* ***Conclusions.*** *The results demonstrate that no universal model is suitable for all types of NLP tasks. It is important to choose an embedding method that matches the specific task, available resources, and specific research goals. The results of this study can be extended to other NLP tasks, such as tone analysis, named entity detection, and machine translation.*

Keywords: *classification; NLP; context; model; neural network; Word2Vec; GloVe; embedding; BERT; GPT.*

1. Introduction

Modern requirements for language processing challenge us to develop more efficient systems that can analyze and generate natural language. This applies not only to textual information but also to voice communications. Speech understanding and processing [1,2] are important components of intelligent systems that can interact with people in natural language. In this context, it is important to research and develop methods and technologies to improve the quality of natural language recognition and understanding [3]. One such technology is the use of contextual embeddings, which allow us to better capture the semantic context of words in a text. The expression "embedding" comes from the idea of "embedding" words in a numerical space, where each word or phrase is represented as a vector of numbers [4, 5]. These vectors can reflect the semantics of words, their meanings, and relationships with other words in the text. The analysis of the impact of these embeddings on the accuracy of text classification is a relevant task in the field of language processing and machine learning.

Paper structure. Section 1, "Introduction", provides the background and motivation for the research, emphasizing the growing importance of accurate text classification across multiple industries, such as medical diagnostics, finance, and social media. This section introduces the core concepts of word and contextual embeddings, highlighting their relevance to natural language processing (NLP) tasks, and reviews existing literature on various text vectorization methods, including Word Embeddings (Word2Vec, GloVe) and Contextual Embeddings (BERT, GPT). This section identifies the gaps in the current research that the study seeks to address.

Section 2, "Methodology of the experiments", outlines the experimental setup, detailing the IMDB

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

dataset, the preprocessing steps (such as tokenization and normalization), and the LSTM neural network architecture used for text classification. This section also describes the embedding methods (Word2Vec, GloVe, BERT, GPT) and explains how the number of training epochs was adjusted in the experiments.

Section 3, "Results discussion", presents the findings from a systematic comparison of Word and Contextual Embedding techniques. This section presents a detailed analysis of the impact of different vectorization methods on classification accuracy, precision, recall, and F1-score. It also includes visual representations of training dynamics (e.g., validation loss) for each method. This section also interprets the results and compares the advantages and disadvantages of Word2Vec, GloVe, BERT, and GPT. It discusses the practical implications of using these models for NLP tasks and identifies which methods perform best under specific circumstances.

The paper ends with the "Conclusion" section, which summarizes the key findings and reinforces the superiority of Contextual Embeddings over Word Embeddings for complex NLP tasks. It also offers suggestions for future research, including potential improvements to the LSTM model and exploration of other neural network architectures.

1.1. Motivation

Text classification plays an important role in the field of natural language processing (NLP) and is central to other NLP tasks [6]. Text is an important medium of information, and its classification allows solving various tasks, ranging from automatic spam filtering to improving search engine results and analyzing large volumes of documents [7].

Accurate classification helps improve search engines, allowing users to find the information they require faster and more efficiently. Accordingly, improving the accuracy of text classification contributes to improving search results and the quality of information available on the web.

Second, accurate text classification is important in areas where security and filtering play crucial roles. For example, in email, classification tools help identify spam and phishing messages and protect users from fraud and cyberattacks.

Third, the accuracy of text classification is of great importance in industries related to the analysis of public opinion and sentiment. For example, in social networks and media companies, accurate text classification allows them to analyze public reactions to events, resolve reputation issues, and predict trends.

For some industries, such as medicine and law, the accuracy of text classification is critical. For example, in medical diagnostics, automatic text classification helps users quickly find and analyze scientific articles and medical information to improve diagnosis and treatment.

The accuracy of text classification is tangible in various aspects of life and different industries, and its importance is constantly growing with the development of information technology and the increase in text data. Accuracy is a key performance indicator in text classification systems and affects the quality of decisions and user satisfaction.

Text classification using contextual embeddings has wide application prospects (Fig. 1) [8]:

– text classification with contextual embeddings can be used to analyze medical documents, articles, and patient records. This helps in the automatic diagnosis of diseases and detection of medical anomalies [9, 10];

– in the financial sector, contextual embedding can be used to analyze news, financial reports, and articles to predict stock market movements. They are also useful for detecting fraud and deception in financial transactions [11];

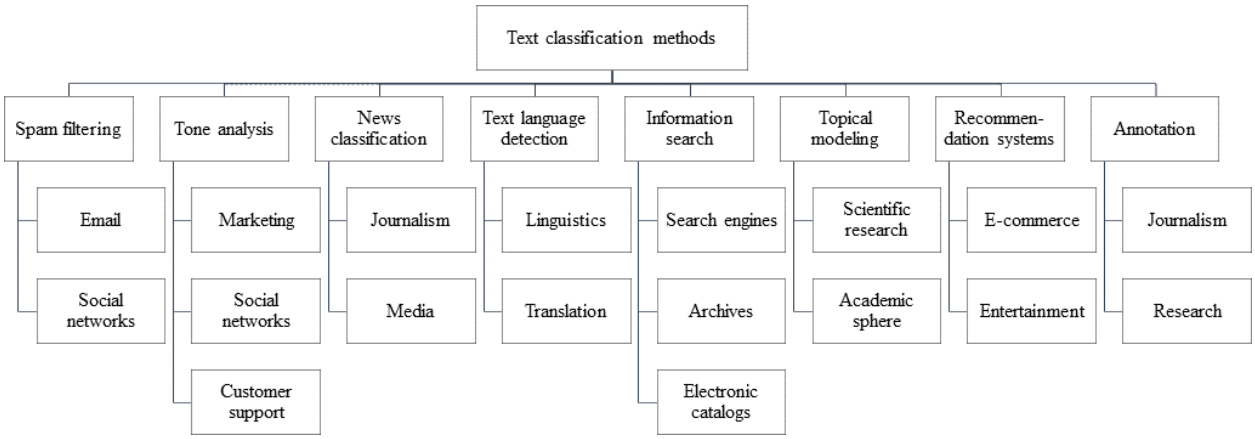


Fig. 1. Areas of application of text classification

– text classification with contextual embeddings helps to analyse reviews, comments, and public statements posted on social media. This is important for business and marketing because it allows us to understand public opinion and reactions to products and services [12];

– the use of contextual embeds allows for more accurate personalization of advertising messages and analysis of consumer responses to advertising campaigns;

– text classification based on contextual embeddings helps identify spam and fraudulent messages in email, thereby ensuring user safety and selectivity of users [13];

– in education, text classification based on contextual embeddings is used to evaluate educational texts, automatically-grade assignments, and even develop individualized curricula;

– in political studies, text classification with contextual embeddings is used to analyze political programs, election promises, and the public statements of politicians.

In general, text classification opens up opportunities for automation and optimization in many areas and continues to grow in importance with the development of information technologies.

1.2. Research task rationale. Related works

Working with text has its own peculiarities that are important to consider when classifying text data [5]. Some of the main characteristics of this model are as follows:

– heterogeneity of the text. A text can be heterogeneous, which includes different writing styles, lexical diversity, and the use of synonyms. This can create difficulties in analysis and classification;

– polysemy and homonymy. Words can have multiple meanings (polysemy) or be identical to spelling but have different meanings (homonymy); Understanding the context is important for correct classification;

– language dependency. Different languages have their own syntactic and semantic features. Text classification models should be adapted to a specific language;

– class imbalance. Some text classification tasks may have an uneven distribution of classes, which requires attention to data balancing methods;

– text patterns. Text can contain various patterns, such as keywords, structured data (e.g., tables or lists), or features related to language grammar.

For text classification, various tools are used to achieve a high level of accuracy, often combining existing methods that have proven themselves. Machine learning methods (Naïve Bayes, Support Vector Machine (SVM) [7], Decision Trees, etc.) can be used to create models that automatically recognize patterns in text data and perform classification based on a set of training data. Neural networks are a special type of machine learning method because they allow automatic detection of internal patterns in text data.

The basic stages of text array classification, from their initial input to the final results, are separate processes - from preprocessing and data normalization to their vectorization and further analysis using neural network algorithms (Fig. 2). The diagram also identifies the input and output streams, and the resources and tools involved at each stage.

The choice of a text data analysis method affects the accuracy and reliability of classification; thus, one should analyze the most common modern neural network models Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM), which represent two different approaches to text data analysis and have their own features that can be useful for different types of texts and classification tasks [5, 14, 15].

The CNN model is unique and specialized for recognizing patterns in images; however, it can also be successfully used to process text data, where it recognizes local dependencies and important features of the textual context. The LSTM model, on the other hand, is a part of recurrent neural networks and has the ability to store and use information from previous steps, which allows it to work effectively with sequential data, especially text sequences, and to consider the context in texts (Table 1).

Table 1

Comparative analysis of LSTM and CNN models for text classification

Feature	LSTM Model	CNN Model
Core architecture	Recurrent neural network with LSTM layer	Convolutional neural network with Conv1D layers
Typical tasks	Sequential data analysis, text data	Sequential data analysis, images
Special features	Takes into account the context of word dependencies	Identifies local patterns in data
Learning algorithms	Backward error propagation, Adam optimizer	Backward error propagation, Adam optimizer
Activation functions	Tanh, Sigmoid	ReLU
Memory usage	Uses short-term and long-term memory	Does not use memory
Applications	Sequential text analysis, language translation	Image and video analysis
Implementation libraries	TensorFlow, Keras	TensorFlow, Keras

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

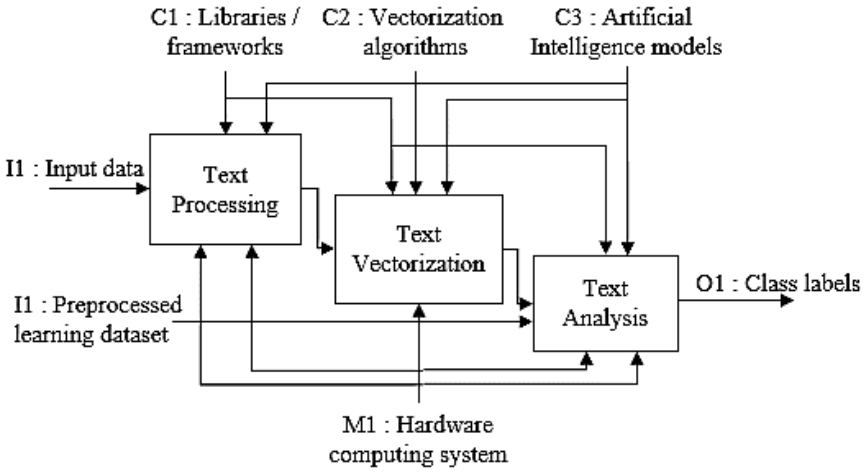


Fig. 2. Basic stages of text data classification

Therefore, an important characteristic of CNNs is their ability to learn useful features and scalability on different tasks automatically. To use CNNs in text analysis, text data are converted into vector form, for example, using word embeddings such as Word2Vec or GloVe. Thus, local features or patterns are extracted in a sequence of word vectors.

The ability to learn features automatically from data is especially useful for tasks in which text has an important context or local features play an important role. However, for more complex text processing tasks, such as meaning recognition, the use of recurrent neural networks (RNNs) or transformers may be more effective because they take into account the sequential context and relationships between words in the text. However, CNNs remain a useful tool for text processing in many cases, particularly when local features play a key role.

A linear recurrent neural network (LSTM, Long Short-Term Memory) is a type of recurrent neural network proposed by Jürgen Schmidguber and Frederik Gagerster. LSTMs have special internal structures called gates that regulate the flow of information through the network.

An important advantage of LSTMs is their ability to store and use information over a long time and learn useful dependencies in the data automatically.

The use of the LSTM model for text processing has its own peculiarities:

– working with sequences. LSTMs are ideal for processing sequential data, such as texts, where the context and dependencies between words are important. They can efficiently model sequences and understand long-term relationships in the text;

– ability to recurrence. LSTMs have a recurrent structure that allows them to remember previous states and take them into account when analyzing new data. This method is especially useful for text analysis, where words can have different meanings in the context;

– analysis of long-term dependencies. LSTMs can understand long-term dependencies in text, which makes them effective for tasks in which history and context are important. For example, in language modeling, it is important to analyze how previous words influence the choice of the next word;

– text classification. LSTMs can be used for text classification, where the category or class to which the text belongs is determined. For example, we determine the sentiment (positive, negative, neutral) in user reviews;

– text generation. LSTMs can be used for text generation, where the model creates new texts with a similar style or context to the input data. This can be useful when creating texts such as news articles;

– Time-series analysis in text. LSTMs can be used to analyze time series in text, such as predicting and recognizing events in large texts or news;

– working with neural word vectors. LSTMs can use vector representations of words, such as Word2Vec or GloVe, to better understand the semantics of words in a text.

In general, the use of LSTMs for text processing makes it possible to effectively analyse word sequences, consider the context, and understand complex dependencies, making LSTMs a powerful tool for many tasks in the field of text processing and sequence analysis.

When analyzing Word Embedding methods, we can distinguish three basic types: Word2Vec, GloVe, and FastText (Fig. 3).

Word2Vec and GloVe are tools to create embeddings that capture semantic relations between words by generating vector representations of them based on structural relations in large text corpora. While Word2Vec uses the local context of surrounding words, GloVe is based on global statistics of word co-occurrence across the entire corpus.

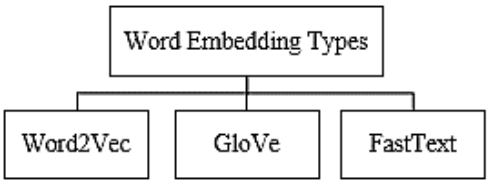


Fig. 3. Word Embedding Types

When analyzing the Contextual Embedding types, BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) and XLNet were identified, which are characterized by the ability to analyze the context of words and generate meaningful text (Fig. 4).

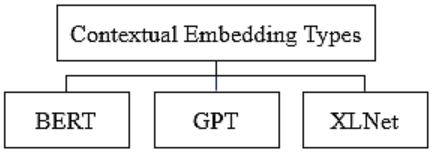


Fig. 4. Contextual Embedding Types

BERT and GPT are evolutions of the concept of embeddings, and they can generate contextual vectors for words by considering all relationships within a whole sentence or even larger text fragments. This provides a more accurate and deeper understanding of language structures, which in turn increases the accuracy of text classification models.

In this study, we used the open-source generative language model GPT-3.5 based on deep neural networks with a transformer architecture.

The advantages of multilayer transformers with a self-attention mechanism are that they allow a model to effectively work with long text sequences, which was used in this study.

In other words, using contextual embeddings to analyze the impact on text classification accuracy is justified because of their ability to better consider the semantic context of words. This can lead to improved NLP solutions and expanded applications in which classification accuracy is critical.

1.3. Text corpus selection

There are many different corpora for training text classification models, which have become an important resource for research and applications in natural language processing [16, 17]. The corpora represent different types of text data from different sources and cover a wide range of topics. To ensure successfully compare neural network models for text classification, it is important to carefully select an appropriate corpus that meets the research goals and task characteristics. Table 2 shows the characteristics of the most common corpora: the IMDB movie review corpus, Reuters news corpus, PubMed scientific article corpus, and Twitter sentiment analysis corpus.

Table 2

Text corpora for text classification in NLP tasks

№	Corpus	Description	Classification task
1	IMDB corpus	Contains movie reviews from IMDB, divided into positive and negative categories.	Binary text classification (positive/negative reviews)
2	Reuters news corpus	Contains news articles organized by topic.	Categorical text classification
3	PubMed scientific article corpus	Contains scientific articles from medical sources for classification by topic or importance.	Categorical text classification
4	Twitter sentiment analysis corpus	Contains Twitter messages categorized by sentiment (positive, negative, or neutral).	Classifying sentiment in short texts

The IMDB corpus was chosen for the paper because of such advantages as diversity, accessibility, representativeness, and the presence of text reviews with emotional coloring, which is an important feature for solving the binary classification problem. The variety of text lengths in the IMDB corpus allows us to test the ability of the models to work with sequences of varying complexity and length, which is crucial for a realistic analysis of neural network performance on a variety of input data.

The use of the IMDB movie review corpus for training and testing allows us to determine research potential aimed at improving the accuracy of text data classification results in the following areas and problems:

– sentiment analysis of product and service reviews. This helps companies understand customer satisfaction, identify problem areas, and improve their products;

– content classification in web services – this will help organizations and platforms automatically filter content to ensure safety and positive user experience;

– analysis of emotions in social networks – this will help marketers understand reactions to news,

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

events, and publications, which are important for advertisers and marketers;

– monitoring brands and companies to track and analyze public opinion and helping managers respond to changes in the perception of goods and services.

1.4. Aims and tasks of the work

The goal of this study is to analyze the impact of using Contextual and Word embeddings on the accuracy of text array classification.

The achievement of the set aim requires solving the following tasks:

– comparative analysis of text document vectorization methods;

– review of existing text classification pipelines, including preprocessing and direct analysis;

– the impact of text vectorization methods (Word2Vec and GloVe) on the accuracy of text classification in Word Embedding based on an LSTM neural network model;

– studying the influence of text vectorization methods (similar to vectorization in the BERT and GPT models) on the accuracy of text corpora classification in Contextual Embedding based on an LSTM neural network model;

– analysis of the accuracy of the obtained results.

The structure of the article is designed to systematically explore and present the impact of contextual embeddings on text classification accuracy. It begins with an Introduction. Introduction provides the context and significance of text classification in various fields, the concept of embeddings, and the goals and scope of the study. This is followed by the Motivation section, which underscores the importance of accurate text classification across different industries and highlights the challenges associated with achieving high accuracy.

The Related work section offers a comprehensive review of the existing literature, summarizing previous studies on Word Embedding and Contextual Embedding methods, and identifying the research gaps addressed by this study. In the Text corpus selection process, existing text corpora are analyzed and the characteristics of the selected corpus are described in detail. The Methodology section provides a detailed account of the experimental setup, including the data description, preprocessing steps, and embedding methods. This section also explains the architecture of the LSTM neural network model and the evaluation metrics employed. The experiments and results section presents the findings from the comparative analysis of Word2Vec, GloVe, BERT, and GPT models, and they discuss their performance across different epochs. The Discussion interprets these results, providing insights and implications for future research and practical applications, while also acknowledging the study's limitations. Finally, the Conclusions summarize the key findings and suggest directions for future work. The article is supported by a comprehensive list of references, ensuring the credibility and context of the research.

2. Methodology of the experiments

Given the complexity of modern NLP tasks, an effective functional model is critical to achieve high results accuracy. The conceptual model for classifying text corpora visualizes the main processes and their interrelationships and identifies research factors (Fig. 5).

The input data is the publicly available IMDB dataset, which is commonly used for natural language processing and sentiment analysis tasks. This dataset contains 50,000 reviews evenly distributed between positive and negative categories. Positive reviews are rated 7 out of 10 or higher, and negative reviews are rated 4 out of 10 or lower. This markup provides a noticeable sentiment difference, contributing to the classification accuracy. The dataset contains only the text of the reviews without any additional meta-information, such as movie ratings or review data, which allows us to focus on text analysis. Reviews can vary in size and provide several data for analysis, from short comments to detailed reviews.

To organize machine learning processes, the IMDB dataset was divided into three main parts - trainings, validation, and test:

– training dataset (70%) to provide sufficient examples for effective learning and adaptation to a variety of textual feedback;

– validation dataset (15%) to periodically evaluate the model during training, which allows monitoring and preventing model overtraining, and helps in fine-tuning hyperparameters;

– a test dataset (15%) was used to evaluate the final accuracy of the model after training on data that were not used during training.

Text pre-processing includes normalization and tokenization of reviews from the IMDB dataset. Normalization includes converting all text to lowercase letters and removing unnecessary characters and punctuation that do not convey important information. Tokenization is the process of dividing a text into individual elements or tokens to create a clear map for vectorization. The final preprocessing step is the removal of stop words, which are the most frequently used words that do not carry a significant semantic load [18]. This sequence of preprocessing methods provides a structured, noise-free dataset for the following stages of analysis and classification [19, 20].

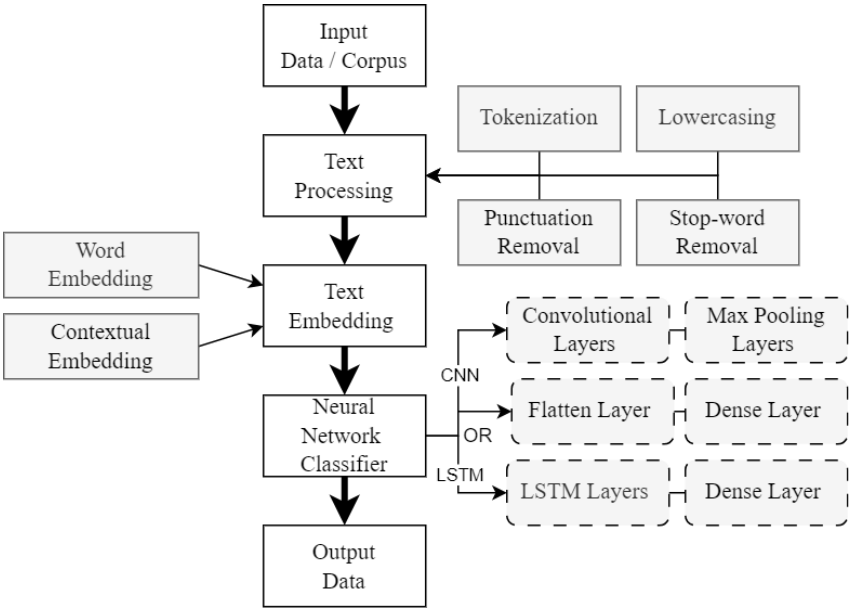


Fig. 5. A conceptual model for classifying text corpora

The text vectorization stage is the step in which text is converted into a format suitable for machine learning. For this study, we consider Word Embedding models, such as Word2Vec and GloVe, and Contextual Embedding models, such as BERT and GPT.

For the task of text classification, the LSTM (Long Short-Term Memory) model was chosen, which is a type of recurrent neural network that is effectively used to analyze sequences of data, such as text.

Using embedding methods, each word in the text corpus is converted into a fixed-length vector. The resulting embedding matrix is input to the LSTM network. The model structure includes the following layers:

– the first layer in the architecture is LSTM, which comprises 100 units. The use of a hyperbolic tangential activation function (tanh) helps support nonlinearity in internal data structures. This layer provides the model with the ability to memorise information for longer periods and to work more efficiently with sequences;

– after the LSTM layer, the Dense Layer is used to derive the final forecast. The sigmoid function is ideal for binary classification because it outputs a value between 0 and 1, which is interpreted as the probability of belonging to a particular class;

– the model is compiled using the adam optimizer, which is effective for various tasks, and the binary_crossentropy loss function, which is a standard binary classification method. As a metric, we used accuracy, which evaluates classification accuracy.

All these elements form a model for solving the problem of analyzing the sentiment of large text corpora (Fig. 6), in our case, for the task of binary classification (positive, negative) of movie reviews.

The research methodology is based on a matrix of experiments that includes a systematic comparison of the effectiveness of each embedding type. Each model was evaluated based on several key metrics, including accuracy, confusion matrix, F1 measure, and ROC-AUC Score. Experimental studies were planned as a systematic approach to compare four types of vector word representations: Word2Vec and GloVe for Word Embedding, and BERT and GPT for Contextual Embedding.

In the matrix of experiments, each embedding method was tested on a predetermined LSTM architecture, which ensured the equality of the experimental conditions (see Table 3).

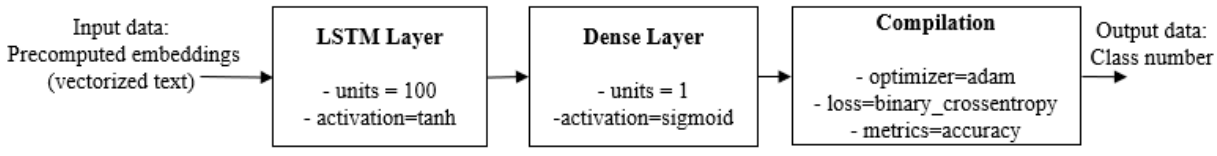


Fig. 6. LSTM model architecture

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Table 3

Planning matrix of a multivariate experiment to determine the influence of the number of training epochs for the selected types of text array vectorization on the accuracy of input data classification within Word Embedding or Contextual Embedding

Experiment number	Factors of influence			Purposes for which dependencies are assessed
	Variable number of training epochs	Method of text data vectorization # 1	Method of text data vectorization # 2	Train accuracy, Test accuracy, Test loss, Training time
1	1	+	-	+
2	1	-	+	+
3	2	+	-	+
4	2	-	+	+
…	…	…	…	…
2N-1	N	+	-	+
2N	N	-	+	+

The purpose of the experiments was to determine the method of text corpus vectorization and the number of training epochs of the neural network model of text classification for which the following indicators will be analyzed:

– train accuracy measures the error that the model makes on the training set but does not guarantee high prediction accuracy on new data that it has not seen before;

– test accuracy provides an objective assessment of the model's performance, as it indicates the percentage of correctly classified samples in the test dataset. The higher the test accuracy, the better the model was trained on the training dataset and could make accurate classifications on new data;

– validation loss is a measure of the error generated by the machine learning model on the validation set. The validation loss is used to evaluate the performance of a model during training, indicating how well the model generalises to new data. The lower the validation loss, the better the model generalises and makes accurate predictions on data it has not seen before. A high validation loss indicates overfitting on the training set. A significant increase in validation loss may indicate that the model has started to overlearn and should be stopped. Test loss, unlike validation loss, measures the error that the model makes on the test set, i.e., the part of the IMDB dataset that is not used to train the model and is not used to select model hyperparameters. In other words, the test loss is used only once to obtain a final estimate of the model's performance;

– training time (shows the total time spent on model training).

The number of experiments was not predetermined and depended on the number of epochs N, starting from which the validation accuracy of the classification began to decrease.

As part of matrix formation, we first consider a comparison of Word2Vec and GloVe, determine the parameters that will vary or remain unchanged, and determine the metrics used to evaluate each embedding. Then BERT and GPT are analyzed in a similar manner.

3. Results discussion

To analyze the impact of such Word Embedding models as Word2Vec and GloVe, on the accuracy of text data classification using a neural network with LSTM architecture, some experiments were performed (Experiment 1). The goal is to determine the effect of the number of training epochs of the neural network on the accuracy, training speed, validation loss, and validation accuracy for different vectorization methods (Word2Vec and GloVe) within the Word Embedding approach.

Table 4 provides a detailed overview of the changes in precision, validation accuracy, and validation loss for each Word Embedding method over 15 training epochs.

The results in the table show a decrease in accuracy for the validation set and an increase in loss of accuracy after the 10th training epoch, which may indicate that the model was over trained. The results for the 15th epoch demonstrate that further training reduced the model's ability to generalize to new data and increased computational costs without improving the results.

After determining the required number of training epochs to achieve the highest validation accuracy, the following indicators were evaluated: test accuracy, ROC-AUC Score, precision, recall, and F1-score for each text vectorization method (Table 5).

To visualize the dynamics of model training using the Word2Vec and GloVe methods, we plotted the loss change during all training epochs (Fig. 7).

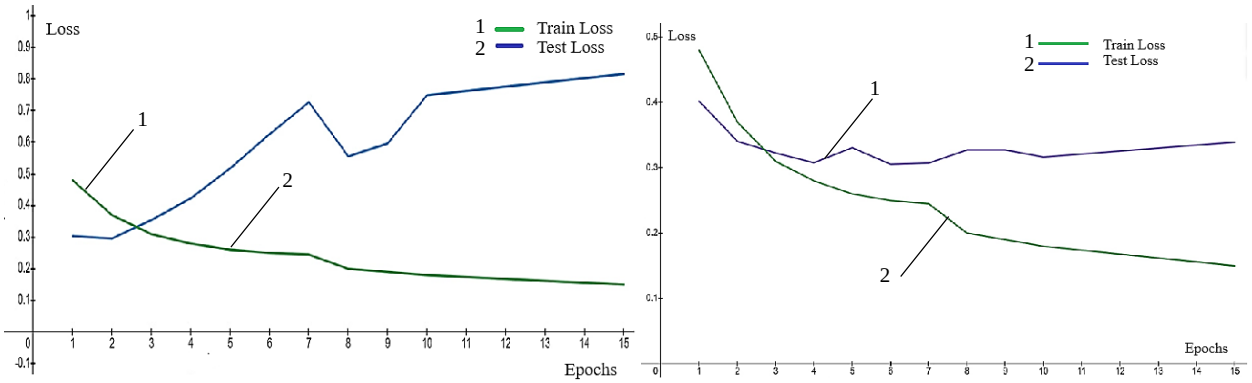
The graph shows the losses on the training dataset (Train Loss) and the validation dataset (Test Loss), which allows us to assess the overall stability of the model and is an important indicator for analyzing and optimizing the training process (see Fig. 7).

The analysis of the results of Experiment 1 revealed that each method had its advantages and disadvantages. Word2Vec demonstrated high accuracy, but with a certain level of loss, especially in later epochs,

Table 4

Table of metrics for each of the 15 epochs for Word2Vec and GloVe

Epoch	Word2Vec Accuracy, %	Word2Vec Test Accuracy, %	Word2Vec Test Loss	GloVe Accuracy, %	GloVe Test Accuracy, %	GloVe Test Loss
1	84.94	88.77	0.3037	76.97	82.33	0.4019
2	94.10	88.19	0.2963	83.79	85.41	0.3406
3	96.82	86.80	0.3543	86.53	86.43	0.3227
4	97.02	87.40	0.4230	87.76	87.06	0.3077
5	98.94	87.13	0.5181	88.77	86.61	0.3308
6	99.22	86.47	0.6262	89.54	87.30	0.3056
7	99.26	83.90	0.7275	90.63	87.98	0.3073
8	98.74	85.93	0.5562	91.23	87.73	0.3033
9	99.37	86.55	0.5966	92.12	86.86	0.3273
10	99.75	88.77	0.7497	93.17	87.72	0.3166
15	99.75	86.03	0.8173	93.00	87.21	0.3392



a) b)

Fig. 7. Loss trends in model training and testing using vectorization methods: a) Word2Vec, b) GloVe

Table 5

Table of the experimental results

Embedding Type	Word2Vec	GloVe
Number of Epochs	10	8
Test Accuracy, %	88.77	87.73
Test Loss	0.7497	0.3033
ROC-AUC Score	0.9333	0.9457
Precision (avg)	0.87	0.88
Recall (avg)	0.87	0.88
F1-Score (avg)	0.87	0.88
Training Time, sec	6340	3830

which may indicate overfitting. GloVe, on the other hand, performed more consistently on the validation dataset, although its overall accuracy was slightly lower than Word2Vec. These differences emphasize the importance of selecting an appropriate vectorization method for a particular task and dataset.

To analyze the impact of Contextual Embedding models, such as BERT and GPT, on the accuracy of text array classification by a neural network with LSTM architecture, some experiments were performed (Experiment 2). The goal was to determine how the number of training epochs of the neural network affects the accuracy, training speed, validation loss, and validation precision for different vectorization methods (BERT and GPT) within the Contextual Embedding approach.

The table provides a detailed overview of the changes in accuracy, validation accuracy, and validation loss for each Contextual Embedding method over 10 training epochs (Table 6).

To illustrate the learning progress of the BERT and GPT models, we constructed graphs that show changes in the loss rate over all epochs. The graphs provide a more visual comparison between the performance of the considered contextualization methods (Fig.8).

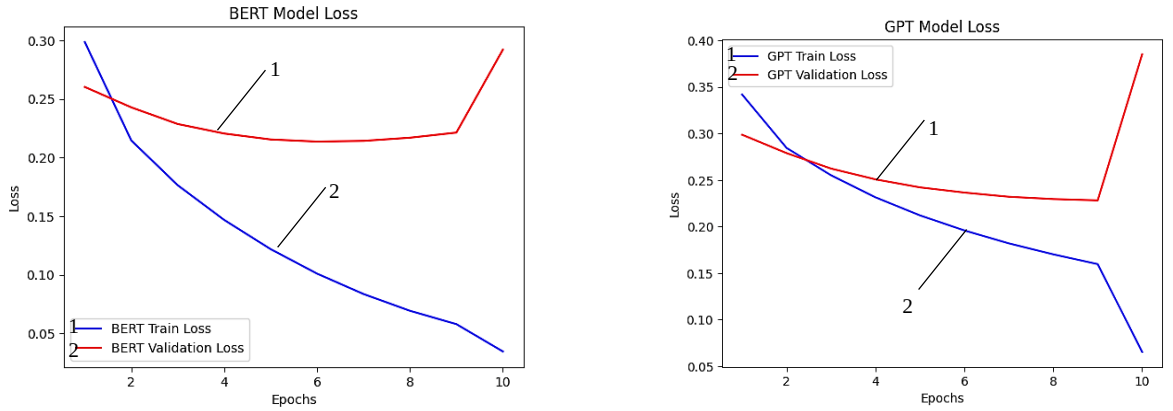
The analysis of the results obtained by BERT and GPT demonstrates different aspects of the effectiveness of these models in terms of text classification. BERT, with a final precision of 0.9123 and a relatively low loss of 0.2919, demonstrated high ability to analyze and understand the context of the text. The high ROC-AUC score (0.951) indicates an advanced understanding of the nuances of language. In contrast, GPT with final precision of 0.8945 and a loss of 0.3851, proved to be slightly less accurate but still effective, particularly in

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Table 6

Table of metrics for each of the 10 epochs for BERT and GPT

Epoch	BERT Accuracy, %	BERT Test Accuracy, %	BERT Test Loss	GPT Accuracy, %	GPT Test Accuracy, %	GPT Test Loss
1	87.62	90.14	0.2601	85.50	88.92	0.2987
2	91.53	90.98	0.2427	88.04	89.27	0.2789
3	93.27	91.45	0.2285	89.31	89.84	0.2623
4	94.52	91.82	0.2204	90.28	90.29	0.2507
5	95.58	92.07	0.2153	91.12	90.65	0.2421
6	96.41	92.31	0.2135	91.76	90.97	0.2365
7	97.12	92.56	0.2141	92.39	91.20	0.2320
8	97.65	92.79	0.2168	92.87	91.43	0.2295
9	98.14	92.97	0.2212	93.34	91.65	0.2281
10	98.91	91.23	0.2919	97.79	89.45	0.3851



a) b)

Fig. 8. Loss trends in model training and testing using vectorization methods: a) BERT, b) GPT

the context of text generation and processing more creative tasks. In general, both models demonstrate significant text classification efficiency; however, the choice between them depends on the specific task.

After comparing the performance estimates of the models on the test sample based on the test loss, the final assessment of the classification accuracy of the selected model was performed based on the test accuracy (Table 8).

The GPT-3.5 language model has several speech-processing capabilities despite the lack of the ability to train on multimodal data, such as GPT-4. However, the experiment demonstrated that the BERT language model outperformed the GPT-3.5 model by 1.42% in test accuracy over 9 epochs, so conducting an experiment with the GPT-3 language model for text data classification is not advisable.

Although GloVe has lower accuracy than BERT. it is less complex and requires fewer computational resources. On the other hand, BERT with its high accuracy and low loss. has proven to be effective in more complex NLP tasks where a deeper understanding of the context and language nuances is required.

Table 7

Table of the experimental results

Embedding Type	BERT	GPT
Epochs	9	9
Test Accuracy, %	92.97	91.65
Test Loss	0.2212	0.2281
ROC-AUC Score	0.951	0.939
Precision (avg)	0.91	0.89
Recall (avg)	0.91	0.90
F1-Score (avg)	0.91	0.89
Training Time, sec	11340	14030

Table 8

Table of the experimental results

Embedding Type	Word Embedding (GloVe)	Contextual Embedding (BERT)
Test Accuracy, %	87.73	92.97

Conclusion

This study conducted a comprehensive analysis of the impact of using contextual and verbal embeddings on the accuracy of text data classification. The study managed to achieve the set goals and solve the task set.

We conducted a detailed study and compared Word2Vec and GloVe. and BERT and GPT vectorization methods. This allowed us to determine the best techniques for different types of text data.

Different approaches to preprocessing and text analysis have been studied and described. which emphasizes the importance of each stage in building an effective classification system.

The application of Word2Vec and GloVe in the context of LSTM neural network model demonstrated that both methods effectively vectorize text for classification. Word2Vec and GloVe have different approaches to vectorization. both demonstrate the ability to capture semantic relationships between words. A comparison of the accuracy between the two methods showed that GloVe has a slight advantage in classifying text corpora. which may be due to its ability to integrate global word co-occurrence statistics across the entire corpus.

The use of BERT and GPT in the context of the LSTM neural network model significantly improved the classification results compared to the Word Embedding methods. This confirms that they are more effective at capturing complex language relations and contexts. Contextual embeddings demonstrated the ability to better understand linguistic nuances. which contributed to improved classification accuracy. especially in complex cases where the context of the word is crucial. The results emphasize the importance of using contextual embeddings for complex text processing tasks. including classification tasks.

We compared the results obtained by the best Word Embedding and Contextual Embedding models. The best Word Embedding model. based on our evaluation criteria. was GloVe. which demonstrated a final accuracy of 87.73%. In the context of Contextual Embedding. BERT proved to be more effective than GPT. with a final accuracy of 92.97% versus 91.65% for GPT.

Given the results, we recommend BERT and GPT as the most effective tools for text classification tasks. especially when adequate computing resources are available. For situations with limited resources or processing speed requirements. Word2Vec and GloVe are reliable options.

The scientific significance of this study is to deepen our understanding of the mechanisms underlying the vector representation of words and their impact on classification processes. These findings can serve as a basis for further research in the field of machine learning and the development of intelligent systems. which will contribute to advances in the field of artificial intelligence.

To further develop this study and increase its practical value, several promising areas can be considered. Experiments with alternative models of contextual embedding. In this paper, we consider an LSTM model using Word Embedding and Contextual Embedding. However, other contextual models, such as BERT and GPT, also deserve attention. The comparison of these models can expand our understanding of the impact of their models on text classification accuracy. To improve the classification accuracy, additional training data can be considered. The addition of more diverse and large corpora may have a positive impact on the results.

Contribution of authors:

Formulation of the problem **Olesia Barkovska. Anton Havrashenko, Patrik Rusnak;** hardware platform investigation and development tools selection **Vitalii Serdechnyi, Vladyslav Kholiev.**

Methodology of the experiments **Olesia Barkovska;** word and contextual embeddings methods realization **Anton Havrashenko, Vitalii Serdechnyi;** analysis and processing of the obtained results **Olesia Barkovska, Patrik Rusnak;** finalization of the draft article version **Olesia Barkovska;** corrections and postediting **Anton Havrashenko, Vladyslav Kholiev.**

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Financing

This research was conducted without financial support.

Data availability

The manuscript contains no associated data.

Use of Artificial Intelligence

The authors confirm they did not use artificial intelligence methods while creating the presented work.

All the authors have read and agreed to the published version of this manuscript.

References

1. Chowdhary, K. R. Natural Language Processing. In: *Fundamentals of Artificial Intelligence*.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Springer. New Delhi, 2020, pp. 603-649. DOI: 10.1007/978-81-322-3972-7_19

2. Khurana, D., Koli, A., Khatter, K., & Singh, S. Natural language processing: state of the art, current trends and challenges. *Multimed Tools* Appl., 2023, vol. 82, pp. 3713–3744. DOI: 10.1007/s11042-022-13428-4.

3. Habib, M. A., Ema, R. R., Islam, T., Arafat, M. Y., & Hasan, M. Automatic text summarization based on extractive-abstractive method. *Radioelectronic and Computer Systems*, 2023, no. 2, pp. 5-17. DOI: 10.32620/reks.2023.2.01.

4. Patil, R., Boit, S., Gudivada, V., & Nandigam, J. A Survey of Text Representation and Embedding Techniques in NLP. *IEEE Access*, 2023, vol. 11. pp. 36120-36146. DOI: 10.1109/ACCESS.2023.3266377.

5. Birunda, S. S., & Devi, R. K. A Review on Word Embedding Techniques for Text Classification. In: Raj, J. S., Iliyasu, A. M., Bestak, R., Baig, Z. A. (eds) Innovative Data Communication Technologies and Application. *Lecture Notes on Data Engineering and Communications Technologies*, Springer, Singapore, 2021, vol. 59, pp. 267-281. DOI: 10.1007/978-981-15-9651-3_23.

6. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao. J. Deep learning--based text classification: a comprehensive review. *ACM computing surveys*, 2021, vol. 54, iss. 3, article no. 62, pp. 1-40. DOI: 10.1145/3439726.

7. Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., & Ijaz, M. F. A complete process of text classification system using state-of-the-art NLP models. *Computational Intelligence and Neuroscience*, 2022, vol. 2022, article no. 1883698. 26 p. DOI: 10.1155/2022/1883698.

8. Rakib Mollah, M. A., Jahangir Kabir, M. M., Reza, M. S., & Kabir, M. Adapting Contextual Embedding to Identify Sentiment of E-commerce Consumer Reviews with Addressing Class Imbalance Issues. *2024 International Conference on Advances in Computing. Communication. Electrical. and Smart Systems (iCACCESS)*, Dhaka, Bangladesh, 2024, pp. 1-6. DOI: 10.1109/iCACCESS61735.2024.10499554.

9. Lavanya, P., & Sasikala, E. Deep Learning Techniques on Text Classification Using Natural Language Processing (NLP) In Social Healthcare Network: A Comprehensive Survey. *2021 3rd International Conference on Signal Processing and Communication (ICPSC*), Coimbatore, India, 2021, pp. 603-609. DOI: 10.1109/ICSPC51351.2021.9451752.

10. Spasic, I., Nenadic, G. Clinical Text Data in Machine Learning: Systematic Review. *JMIR Med Inform*, 2020; vol. 8, iss. 3, article no. e17984. DOI: 10.2196/17984.

11. Gupta, A., Dengre, V., Kheruwala, H. A., & Shah, M. Comprehensive review of text-mining applications in finance. *Financ Innov*, 2020, vol. 6, article no. 39. DOI: 10.1186/s40854-020-00205-1.

12. Lan, L., Huang, T., Li, Y., Song, Y. A Survey of Cross-Lingual Text Classification and Its Applications on Fake News Detection. *World Scientific Annual Review of Artificial Intelligence*, 2023, vol. 1, article no. 2350003. DOI: 10.1142/S2811032323500030.

13. Padalko, H., Chomko, V., Yakovlev, S., & Chumachenko, D. Ensemble machine learning approaches for fake news classification. *Radioelectronic and Computer Systems*, 2023, no. 4, pp. 5-19. DOI: 10.32620/reks.2023.4.01.

14. Li, R., Liu, M., Xu, D., Gao, J., Wu, F., & Zhu, L. A review of machine learning algorithms for text classification. In: Lu, W., Zhang, Y., Wen, W., Yan, H., Li, C. (eds) *Cyber Security. CNCERT 2021. Communications in Computer and Information Science*, Springer, Singapore, 2022, vol. 1506, pp. 226-234. DOI: 10.1007/978-981-16-9229-1_14.

15. Zhou, Y. A review of text classification based on deep learning. *3rd international conference on geoinformatics and data analysis*, New York, USA, Association for Computing Machinery Publ., 2020, pp. 132-136. DOI: 10.1145/3397056.3397082.

16. Nandwani, P., & Verma, R. A review on sentiment analysis and emotion detection from text. *Soc. Netw. Anal. Min.*, 2021, vol. 11, article no. 81. DOI: 10.1007/s13278-021-00776-6.

17. Pandya, S., & Mehta, P. A review on sentiment analysis methodologies, practices and applications. *International Journal of Scientific and Technology Research*, 2020, vol. 9, iss. 2, pp. 601-609. Available at: https://www.researchgate.net/publication/344487215_A_Review_On_Sentiment_Analysis_Methodologies_Practices_And_Applications (accessed 12/02/2024).

18. Barkovska, O. Performance study of the text analysis module in the proposed model of automatic speaker's speech annotation. *Computer systems and information technologies*, 2022, no. 4, pp. 13-19. DOI: 10.31891/csit-2022-4-2.

19. Barkovska, O., Kholiev, V., Havrashenko, A., Mohylevskyi, D. & Kovalenko,A. A Conceptual Text Classification Model Based on Two-Factor Selection of Significant Words //*COLINS* (2). – 2023. – С. 244-255.

20. Barkovska, O., Khomych, V., & Nastenko, O. Research of the text processing methods in organization of electronic storages of information objects. *Innovative technologies and scientific solutions for industries*, 2022, no. 1(19), pp. 5-12. DOI: 10.30837/ITSSI.2022.19.005.

Received 17.05.2024. Accepted 20.08.2024

АНАЛІЗ ВПЛИВУ ВИКОРИСТАННЯ КОНТЕКСТУАЛЬНИХ ЕМБЕДИНГІВ НА ТОЧНІСТЬ КЛАСИФІКАЦІЇ ТЕКСТУ

О. Ю. Барковська, А. О. Гаврашенко, В. С. Сердечний, В. О. Холєв, П. Руснак

Робота присвячена проблемі підвищення точності класифікації текстових масивів. що має критичне значення у таких галузях. як медична діагностика. юриспруденція тощо. Окрім того. вимоги до точності постійно зростають з розвитком інформаційних технологій та збільшенням обсягів текстових даних. **Предметом** статті є дослідження впливу методів векторизації тексту на точність класифікації текстових масивів. **Метою** даного дослідження є оцінка ефективності різних методів векторизації слів (Word2Vec. GloVe. BERT та GPT) у контексті класифікації тексту на основі різних стратегій до використання ембедингів - Word та Contextual Embedding. Основна увага приділяється вивченню впливу кількості епох навчання (шляхом систематичного збільшення числа епох тренування) на точність класифікації текстових даних. **Задачею роботи є** систематичне порівняння ефективності кожного виду ембедингу у відповідності до сформованої матриці експериментів. яка контролює рівність умов виконання експерименту. та подальша оцінка ключових метрик класифікації тексту (на прикладі IMDB датасету) нейромережевим класифікатором (LSTM) із рекурентною архітектурою. В робі використані **методи** машинного навчання. зокрема нейромережеві методи. методи векторого представлення слів. методи статистичного аналізу. За **результатами** дослідження було встановлено. що найкращою моделлю Word Embedding є GloVe. яка продемонструвала кінцеву точність на рівні 87.73%. У контексті Contextual Embedding. BERT виявився ефективнішим порівняно з GPT. з кінцевою точністю 92.97% проти 91.65% в GPT. **В цілому**. отримані результати свідчать про перевагу Contextual Embedding у завданнях обробки природної мови і підтверджують їхню перспективність для сучасних додатків та систем текстового аналізу. Результати демонструють. що немає універсальної моделі. яка підійде для всіх типів задач NLP. Важливо вибирати метод ембедингу. орієнтуючись на специфіку задачі. доступні ресурси та конкретні цілі дослідження.

Ключові слова: класифікація; NLP; аналіз; контекст; модель; нейронна мережа; Word2Vec; GloVe; embedding; BERT; GPT.

Барковська Олеся Юріївна – канд. техн. наук, доц., доц. каф. Електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна

Гаврашенко Антон Олегович - асп. каф. Електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна

Сердечний Віталій Сергійович - асп. каф. Електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна

Холєв Владислав Олександрович - асп. каф. Електронних обчислювальних машин, Харківський національний університет радіоелектроніки, Харків, Україна

Руснак Патрік – д-р філос. з комп'ютерних наук, доц. каф. інформатики Жилінського університету, Жиліна, Словаччина.

Olesia Barkovska - PhD, Associate Professor, Associate Professor at the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
e-mail: olesia.barkovska@nure.ua, ORCID: 0000-0001-7496-4353. Scopus Author ID: 24482907700.

Anton Havrashenko – PhD Student of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
e-mail: anton.havrashenko@nure.ua, ORCID: 0000-0002-8802-0529.

Vitalii Serdechnyi – PhD Student of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
e-mail: vitalii.serdechnyi@nure.ua, ORCID: 0009-0007-8828-5803.

Vladyslav Kholiev – PhD Student of the Department of Electronic Computers, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine,
e-mail: vladyslav.kholiev@nure.ua, ORCID: 0000-0002-9148-1561.

Patrik Rusnak – PhD (Computer Sciences), Assistant Professor at the Department of Informatics, University of Zilina, Zilina, Slovakia,
e-mail: patrik.rusnak@fri.uniza.sk, ORCID: 0000-0001-9683-5376, Scopus Author ID: 57195920975.