Intellectual Information System for Supporting Text Data Rephrasing Processes Based on Deep Learning

Nickolay Rudnichenko[a], Vladimir Vychuzhanin[a], Natalia Shibaeva[a], Svetlana Antoshchuk[a] and Igor Petrov[b]

[a] *Odessa National Polytechnic University, Shevchenko Avenue, 1, Odessa, 65001, Ukraine*
[b] *National University "Odessa Maritime Academy", Didrichson str., 8, Odessa, 65029, Ukraine*

Abstract

This paper focuses on the specifics of the an intelligent information system development for supporting the processes of large volumes text data analysis and rephrasing based on deep learning models. An overview of the key problems in the field of content analysis on the Internet, in particular, text data, is carried out, ways of solving existing difficulties are indicated, the artificial neural networks models using necessity to automate the process of natural language processing is substantiated. Paper presents the results of LSTM and Transformer models advantages analysis, software implementation within the framework of an intelligent information system to support dialogue and text paraphrasing functions. System's design diagrams in UML are presented, its functionality is described and an artificial neural network model's effectiveness experimental study using the generated test data sets is provided. As a result the paraphrases generation quality metrics were assessed based on the use of the Loss and BLEU indicators. In the conclusion the authors describe ways for the further system's development and possible researches in related areas of natural language processing based on deep learning.

Keywords

Deep learning, artificial Intelligence, data analysis, text rephrasing, intellectual information systems, Big data, information systems design.

1. Introduction

Currently, there is an active growth in the amount of data created and posted in the Internet space on various applied topics. This is largely due to the high demand from consumers for various goods and services, in useful content. In this regard, modern business is actively developing and uses technologies and methods for creating high-quality materials, including for marketing promotion of its products, attracting experts in the required fields to develop targeted and authoritative content, implementing existing and innovative technologies for generating the required information [1-4].

The largest volume of content circulating on the Internet belongs to multimedia materials, including video clips and photographs, located on various platforms, due to a wide range of reproduction and presentation of information in a graphic form, dynamism and interactivity of its display [5].

One of the most critical shortcomings of studying and analyzing this kind of content is the high time spent on its full viewing, as well as the need to use both visual and auditory perception, which is not always effective and justified. In this regard, the most accessible, valuable and convenient for

IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine

EMAIL: nickolay.rud@gmail.com (N. Rudnichenko); vint532@yandex.ua (V. Vychuzhanin); nati.sh@gmail.com (N. Shibaeva); asgonpu@gmail.com (S. Antoshchuk); firmn@list.ru (I. Petrov)
ORCID: 0000-0002-7343-8076 (N. Rudnichenko); 0000-0002-6302-1832 (V. Vychuzhanin); 0000-0002-7869-9953 (N. Shibaeva); 0000-0002-9346-145X (S. Antoshchuk); 0000-0002-8740-6198 (I. Petrov)

analysis and study by a person (and economically less costly for a business) is historically textual content, nowadays often published on websites in the form of articles, essays or comments [6].

This content can be educational, reference, advertising and other in nature, but for its successful creation and use, it is required to ensure its clarity, logical coherence and uniqueness, which enhances the chances of increasing the position of the website on which the content is posted in the search engines used by given relevant words or phrases [7].

This, in turn, enhance the number of potential customers viewing content, helping to increase overall conversions and profit for the business. The laboriousness of creating such content is associated with the need to search and attract experts in the described applied field to writing it, which implies the time and material costs for a preliminary assessment of the performer's capabilities, the quality and consistency of his academic or journalistic style of the target audience, the subject of the article under consideration, the cost of his work, and other, difficult to formalize factors [8-10].

In this regard, tools, technologies, methods and systems for automating the processes of generating text content, its syntactic, punctuation and grammatical processing, assessing the level of uniqueness and other indicators of the quality of the materials obtained are becoming increasingly important. The most promising direction for solving such problems is the theory of artificial intelligence (AI), which includes such areas as data mining (DM), machine (ML) and deep learning (DL) based on the use of artificial neural networks (NN) [11-14].

2. Description of problem

Due to the development of AI concepts, a variety of application systems, tools and means are being created to provide the process of intelligent processing and analysis of textual information in natural language [15]. However, their efficiency for large amounts of text data is often low, including due the impossibility of providing functions for generating complete and coherent sentences or phrases for a given subject area, taking into account its specificity and syntactic rules of a particular language [16]. Full automation of the listed processes is possible only under the condition of multi-criteria perception of the text by the model at the level of human heuristics, which requires the creation of a full-fledged AI, which is still an unattainable task for modern science [17-20].

Nevertheless, the task of paraphrasing textual content is feasible due to the use of the DL approach, the effectiveness of which has significantly increased in recent years through the development of learning algorithms for artificial neural network models and their sets that underlie DL[21]. The first difficulty in using this approach to develop efficient natural language processing tools is the great computational complexity. The most efficient DL algorithms require several days, weeks and months of continuous operation of computational clusters using video cards sets for parallelizing data processing and training models [22].

The second problem is the lack of open, complete, non-contradictory, integral and labeled data for training DL models located and structured in one place. Due to the specifics of each specific task, data must be collected from various sources, which leads to their heterogeneity, the appearance of anomalies, omissions, and errors in syntax, forcing significant resources to be spent on their preprocessing and cleaning. To solve natural language processing problems (paraphrasing in particular) based on the use of NNs, various architectures are used, the most popular and effective in practice are recurrent NN models (in particular LSTMs) and networks of the Transformer type [23].

The essential advantages of these NN architectures for the problem under consideration are:

- support for internal loops, through which the model builds a prediction of the result based on previous input data, which is a kind of memory;
- the possibility of ranking data by importance with the subsequent removal of less significant and preserving more critical in the framework of long-term memory;
- deleting and adding data to the state of a model element by using gate mechanisms (point multiplication operation) [24,25].

It should be noted that additional advantages of Transformer models over LSTMs, critical from the point of view of paraphrasing text content, are [26]:

- implementation of the internal attention mechanism to identify and link relevant words in the text, which helps to speed up the learning process;

- support for the parallelization of computing processes within the framework of building a distributed system;
- no need to process text sequences in a strictly ordered form, which also allows us to speed up the construction of the model [27].

In this regard, having analyzed the possibilities of using the described NN architectures, it is expedient to compare and study the possibilities of using the LSTM and Transformer architectures within the framework of the problem we are solving. Due to the lack of full-fledged and functional ready-made software solutions in the open access and the purpose of the obtained results practical use it is necessary to develop our own applied software that implements the selected NN architecture to solve the considered problem.

The purpose of this article is to develop and study an intelligent information system for supporting the analysis and paraphrasing processes of heterogeneous big text data based on the use of deep learning capabilities (NN models in particular) to improve the quality of the created paraphrases.

3. System development

To study the possibilities of solving the problem under consideration, a decision was made in the development of an intelligent information system (IIS) in the form of a client-server software application based on the use of the Python 3.7 programming language and the Django framework to implement the server-side logic of the system, the PyCharm integrated development environment, Postgress database management system for storage user data, Anaconda distribution kit with Jupiter Notebook (conducting computational experiments to study the effectiveness of NN models), as well as software libraries sklearn (using functionality to create the necessary objects, validate and evaluate the numerical values of metrics), matplotlib (visualization of the obtained data dependencies), Pytorch (creation of NN models and their tuning, setting hyperparameter values and starting computational processes), Pandas and NumPy (processing input data arrays and structuring them for input to NN models). The IIS project is implemented based on the use of the UML language and is a chat application for communicating with real people (registered users of the system) and virtual interlocutors (chat bots) based on the created NN models, as well as for issuing tasks to chat bots to generate paraphrases. A diagram of the main use cases of the system is shown in Fig. 1.

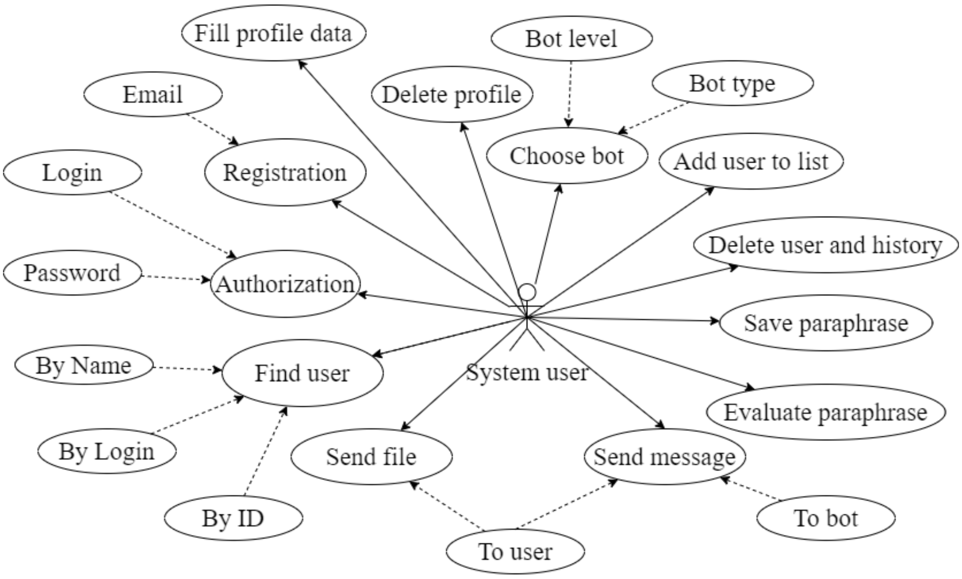

Figure 1: Diagram of system's main use cases

According to the developed use case diagram, the IIS user has next abilities:

- registration in the system to create an account and further work in the application;
- authorization in the system under an account (login and password) to enter user's personal account in order to use its functionality;

- exit and delete the created account;
- search for users registered in the system for dialogue;
- adding the found user account to the contact list;
- sending files and messages to other users;
- removing a user from the list of contacts, chat history;
- choosing a chat bot from the list of available ones;
- conducting communication with a virtual interlocutor;
- sending a command to one or several chat bots to build a set of paraphrases;
- selection and saving of the required variant of the paraphrased sentence from those issued;
- assessment of each of the received paraphrases according to the criterion of acceptability (0 - incorrect paraphrase, 1 - correct).

The general process of user interaction with the system consists of several stages.:

1. Going to the web page of the system through a browser at the URL with subsequent registration. The data entered by the user during registration are checked for compliance with the necessary criteria (complexity and length of the password, the uniqueness of the login) are saved in the database, and a message with a generated one-time link to activate the account is sent to the specified email address.
2. User authorization in the system by login and password. In this case, the data is validated on the client side (profile activity, correctness of the entered data). In case of data errors, a corresponding information message is issued. After successful authorization in the database using the developed control trigger, empty tables of lists of contacts, chats, messages and sent files of the user are created, which are filled in during the use of the system.
3. Selecting operations. After the user is authorized in the system, his profile is loaded, from which it is possible to go to the page for searching for interlocutors, managing personal data created by chats (opening or deleting an existing chat, creating a new chat), viewing the communication history, choosing a chat bot.
4. Chatting with a person. After selecting the desired user or existing chat, a corresponding request is made to the server side through the POST method, after processing which the necessary data is loaded from the database and displayed on the correspondence page, which consists of a text field, a message feed, an interface settings panel and text display formatting, a panel stylized graphic images, buttons for sending files and saving correspondence to an external file in txt format.
5. Interaction with the chat bot. When choosing the function of communicating with a chatbot, the user can specify an option from the available list of existing presets of NN models, on the basis of which the bots were created, view their brief description and characteristics, as well as their functional purpose (the possibilities of simple communication and generation of paraphrases are available). For a more convenient configuration in the mode of correspondence via a text line, a number of commands are available aimed at setting restrictions to the bot, set using the key “-cb -% key =% value”, where key is a control command, and value is a value for this command (maybe number or string). For example, the following options are available: limiting the number of words used in the construction of a paraphrase; entering stop words that are prohibited for use; selection of case and additional language; setting the number of generated paraphrases.

The main components implemented for the implementation of the designated functional are summarized in Fig. 2.

In this diagram, we can see that the system user accesses the main Chat App component, which is responsible for the relationship and interaction of the Chat, Chat Profile, Chat History components with each other.

The Chat component is responsible for sending and receiving messages, Chat Profile implements the functionality of the chat state according to the user's profile, Chat History provides serialization, deserialization and storage of correspondence data.

All components are linked to the data security component (cryptography dependency from the Python open repository) and the database (django.db.backends.postgresql dependency).

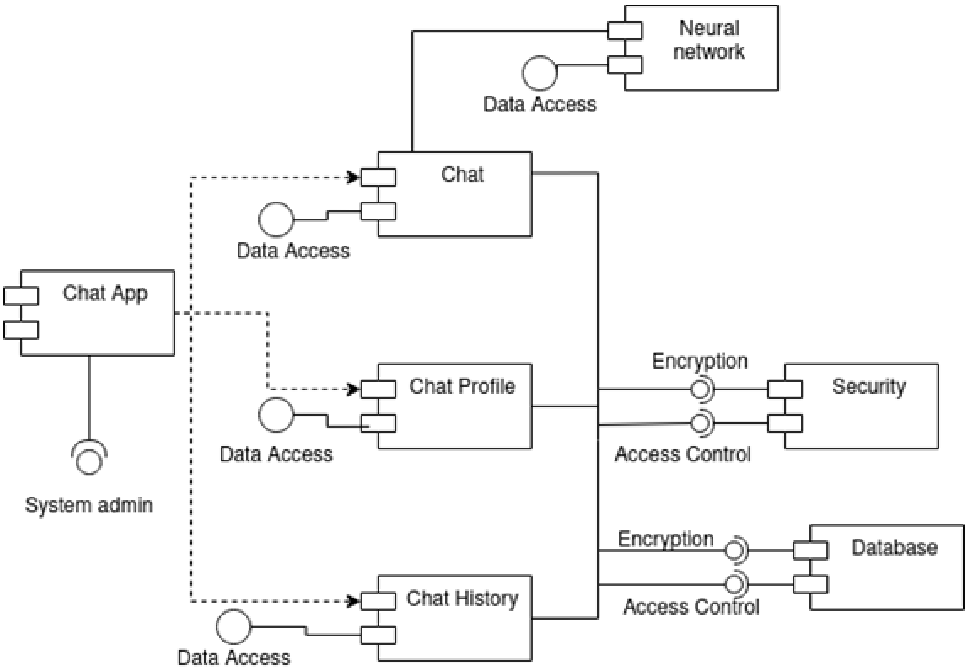

Figure 2: Diagram of key system components

The Neural Networks component provides the functionality for loading the created LSTM and Transformer models into the business logic of the chat bot's work for conducting correspondence and generating paraphrases of the Chat component.

4. Experiments and results analysis

The process of collecting reliable and adequately labeled data for building NN models is a difficult task, therefore, for conducting this study, it is advisable to use existing data sets as a training data sample for NN used to solve the problem of generating paraphrases.

Among the available datasets are Quora Question Dataset [28] and Para-NMT-50m [29]. The first of these datasets is designed to detect duplicate questions on Quora sites. It consists of a series of questions and an end mark that indicates whether they are meaningfully identical. In view of the limited nature of this set, its full use for the problem being solved is not optimal, since as a result of training, the NN model will be able to generate only interrogative sentences. However, it can be partially adapted for the task of maintaining a conversation with the user.

Para-NMT-50m is more suitable for the task of generating paraphrases, because compiled with the help of automatic translation systems and includes 50,000,000 pairs of sentences, which are paraphrases of each other. The complexity of full-fledged processing of this data set lies in the high volume and laboriousness of performing calculations on available hardware. In this regard, by generalization and aggregation by means of the used libraries functional the sample size was reduced to 7,500,000 sentence pairs. They were divided as follows: 6,000,000 for the training sample; 1,000,000 for a test sample; 500,000 for the validation sample.

To use the data correctly and submit them to the input of the NN LSTM and Transformer models, it was necessary to tokenize them (using the WordTokenizer class of the used library) and compose a data structure in the form of a dictionary in which a number is assigned to each token. After that, the procedure was carried out to translate sentences into a sequence of numbers.

The performance of each architecture will be tested on a test dataset that was created previously using the BLEU metric. During the training of the models, 2 optimizers were used - Adam and SGD. Adam is based on using the mean of the second moments of the gradients, its advantages are efficiency for large amounts of data, invariance to gradient scaling, and ease of implementation. The SGD optimizer performs stochastic gradient descent, its ease of use is due to the ability to ensure convergence avoiding local minima and the absence of the need to simultaneously load all training data in the memory of the computing device [30]. Adam hyperparameters that changed during computational experiments: params (implements registration of model weights NN), lr (controls the learning rate of the model), betas (used to calculate the mean value of the gradient and its square), eps (excludes the possibility of division by 0), weight_decay (controls the procedure for reducing the model weights to ensure regularization), n_epochs (limiting the number of model training epochs). In the SGD optimizer, in addition to the hyperparameters indicated above, the following are also used: momentum (acceleration of processing gradient vectors), damping (control of the process of accelerating data processing in time). To compare the operation of the models, a number of computational experiments were carried out, 4 of which are shown in Table 1.

Table 1
The values of the specified hyperparameters of the LSTM model and optimizers

№	NN LSTM		Optimizer	
	Name	Value	Name	Value
1	hidden_size	2*inp_sz	optim	SGD
	num_layers	1	lr	0.01
	bias	False	momentum	0
	batch_first	True	dampening	0.01
	dropout	0	weight_decay	0
	bidirectional	False	n_epochs	10 000
2	hidden_size	2*inp_sz	optim	SGD
	num_layers	1	lr	0.0001
	bias	True	momentum	0.05
	batch_first	True	dampening	0
	dropout	0.3	weight_decay	0.01
	bidirectional	False	n_epochs	10 000
3	hidden_size	3*inp_sz	optim	SGD
	num_layers	2	lr	0.001
	bias	False	momentum	0.1
	batch_first	True	dampening	0
	dropout	0.4	weight_decay	0.05
	bidirectional	False	n_epochs	10 000
4	hidden_size	3*inp_sz	optim	Adam
	num_layers	2	lr	0.001
	bias	True	betas	0.9; 0.99
	batch_first	True	eps	0.000001
	dropout	0.35	weight_decay	0.1
	bidirectional	False	n_epochs	20 000

To build NN models based on the LSTM architecture, the following hyperparameters were changed: input_size (specifies the number of features in the input data), hidden_size (the dimension of the hidden layer of neurons), num_layers (the total number of NN model layers), bias (accounting for deviation parameters), batch_first (places the size batch of data before measuring time), dropout (regulates the process of prohibiting weights on the next layer of the NN model), bidirectional (provides a choice of two-sided LSTM type, to account for sequences both before and after a word from the sample), vocab_sz (determines the size of the dictionary), emb_sz (sets the size of the embedding matrix). To build NN models based on the Transformer architecture, the following hyperparameters were changed: d_model (the number of parameters in the encoder and decoder), nhead (the value of the number of threads), num_encoder_layers and num_decoder_layers (the number of neuron layers in the encoder and decoder), dim_feedforward (the dimension of fully connected layers of the NN model), activation (relu or gelu activation function), src, tgt (sequences for using encoder and decoder models). The obtained dependences of the learning errors of LSTM models are shown in Fig.3.

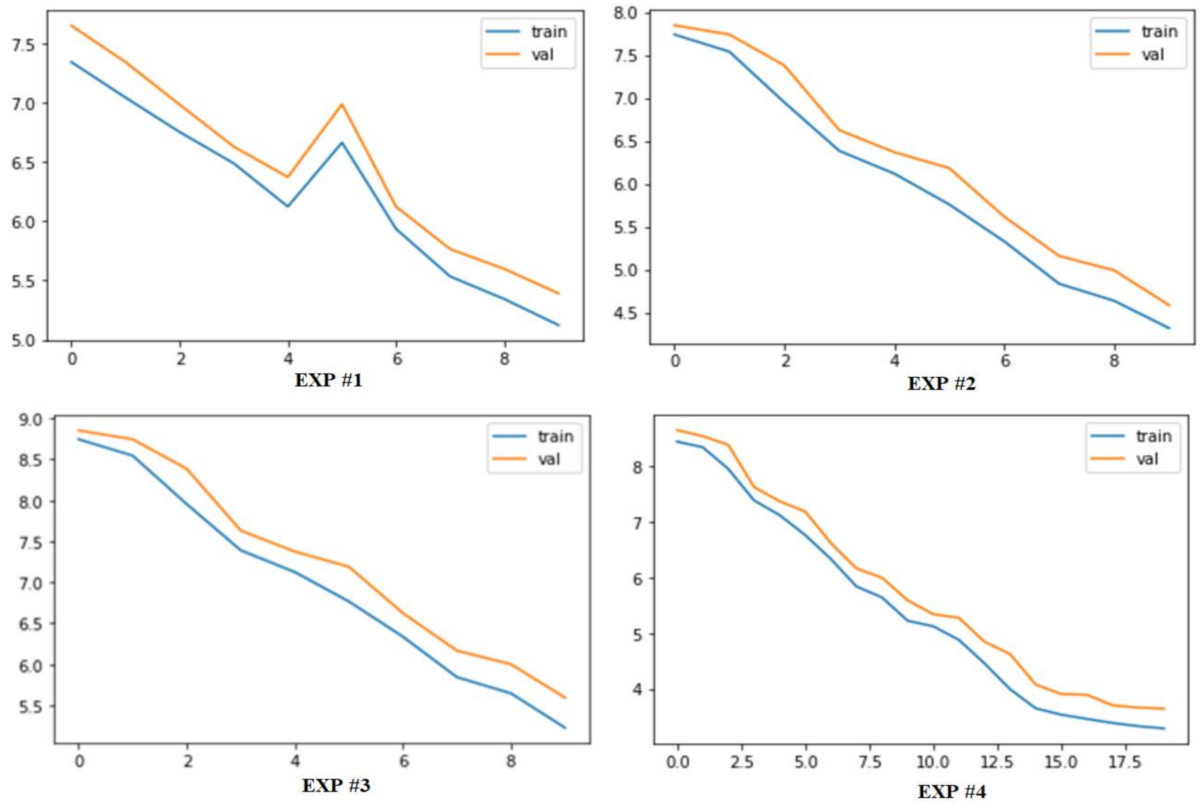

Figure 3: LSTM models learning errors dependences

Based on the results of the first experiment, it can be answered that the LSTM model was trained for 10,000 epochs (the values on the abscissa were scaled 1 to 1000). The value of the loss function gradually dropped from 7.5 to 5.3 on the validation set, and it should be noted that, starting from the 5000th epoch, the indicators of the loss function sharply deteriorated, which may be due to the high value of the model learning rate. For the second experiment, it is characteristic that the value of the loss function gradually (more smoothly compared to the previous case) dropped from 7.7 to 4.5, which means a more appropriate selection of the model's hyperparameters. Analyzing the results of the third experiment, it should be noted that the model's performance deteriorated, since the value of the loss function decreased from a higher value and stopped at 5.3. A similar effect is caused by the fact that the NN model has become more complex and lost its effectiveness due to the increase in the number of hidden layers. The fourth experiment showed that with a significant increase in the number of training epochs (up to 20,000) and a change in the optimizer to Adam, the value of the loss function gradually dropped from 8.4 to 3.6 on the validation set, which is the smallest value, providing the highest accuracy of the obtained LSTM models. As shown by a number of other experiments, a further increase in the number of epochs and a decrease in the betas parameter allows the final value of the function to be reduced to 2.6, but the learning process becomes very time-consuming and laborious in terms of computational costs, and the time costs begin to increase exponentially. The values of the specified hyperparameters and the results of the experiments carried out on the use of the Transformer model are shown in Table 2 and Fig. 4, respectively. According to the experiments, the best loss function in terms of indicator is the Transformer model, the level of which was reduced to a minimum value of 0.6.

Table 2
Values of Specified Transformer Model Hyperparameters and Optimizers

	NN Transformer		Optimizer	
№	Name	Value	Name	Value
1	d_model	512	optim	Adam
	nhead	4	lr	0.0001
	num_encoder_layers	6	betas	0.9; 0.99
	num_decoder_layers	6	eps	0.000001
	dim_feedforward	512	weight_decay	0.1
	dropout	0.2	n_epochs	10 000
	activation	relu		
2	d_model	512	optim	Adam
	nhead	8	lr	0.01
	num_encoder_layers	12	betas	0.88; 0.95
	num_decoder_layers	12	eps	0.000001
	dim_feedforward	1024	weight_decay	0.05
	dropout	0.3	n_epochs	10 000
	activation	gelu		

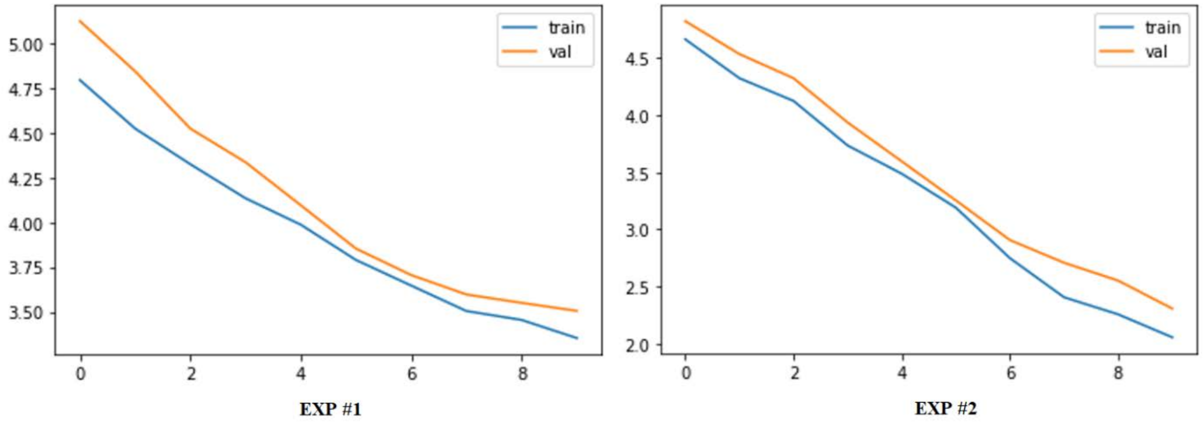

Figure 4: Transformer models training errors dependencies

For a more holistic check paraphrases generation quality by models, it is necessary to use the BLEU metric, which in practice has some dependence on the loss function (the lower the value of the loss function, the higher the BLEU). The results of evaluating the BLEU metric for the created NN models are shown in Table 3. As we can see the Transformer model shows the best result in solving the problem of generating paraphrases. The BLEU metric shows how similar the generated sentence is to the one in the dataset as a reference.

Table 3
BLEU metric values

	BLEU	
Experiment	LSTM	Transformer
1	12.23	29.81
2	21.56	46.73
3	14.02	
4	28.39	

Based on the practical use of the developed system in various scenarios for different texts, it should be noted that on the basis of Transformer models, on average, in 7 out of 10 cases, paraphrase is generated correctly and satisfies expectations, for LSTM models this value is 3-4 out of 10. It should be also noted that all generated sentences within the framework of the study are complete and complete, they have a meaning close to the initial sentences, which is a confirmation of the correctness of the aggregation of the taken dataset.

5. Conclusion

The studies carried out made it possible to establish the adequacy of using DL for solving the generating paraphrases problems. The experimental results obtained allowed us to reveal a greater accuracy of the Transformer models, compared to LSTM (approximately 1.5-2 times). In addition to the described business tasks related to the promotion of goods or services on the Internet, the developed system, due to its modular structure, can be further supplemented with a number of functional capabilities and used to solve problems in various applied areas, in particular:

- to increase the efficiency of the search for borrowings in texts, not only by the direct occurrence of given shingles (as is implemented in existing systems for checking uniqueness based on parsing search results for relevant phrases, word combinations and sentences), but also by possible variations in the order of placing words and taking them into account synonyms;
- to automate the detection of duplicates in text data by indirect occurrences;
- to automate the process of supplementing training data in machine translation systems, expanded by adding paraphrased sentences, datasets become more flexible, increasing the efficiency of DL trained models.

Possible area of further research within the framework of the considered problem is the semantic analysis of text content entered by the user in the form of files or messages, according to various criteria for assessing the difficultly formalized characteristics of the text, including sentiment, emotional coloring or complexity of perception.

6. References

[1] N. Rudnichenko, V. Vychuzhanin, N. Shybaieva, D. Shybaiev, T. Otradskaya, I. Petrov, The use of machine learning methods to automate the classification of text data arrays large amounts. Information management systems and technologies. Problems and solutions. Ecology, Odessa, (2019) 31-46.

[2] V.V. Vychuzhanin, N.D. Rudnichenko, Metody informatsionnykh tekhnologiy v diagnostike sostoyaniya slozhnykh tekhnicheskikh sistem. Monografiya, Odessa (2019).

[3] D.S. Shybaiev, T.V. Otradskaya, M.V. Stepanchuk, N.O. Shybaieva, N.D. Rudnichenko, Predicting system for the estimated cost of real estate objects development using neural networks. ZhSTU Herald. Technical science 83, (2020) 154-160.

[4] N. Rudnichenko, V. Vychuzhanin, I. Petrov, D. Shibaev, Decision Support System for the Machine Learning Methods Selection in Big Data Mining. Proceedings 0f The Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020): session 6 “Intelligent Information Technologies (2020) 872-885.

[5] N. Rudnichenko, S. Antoshchuk, V. Vychuzhanin, A. Ben, I. Petrov, Information System for the Intellectual Assessment Customers Text Reviews Tonality Based on Artificial Neural Networks. Proceedings of the 9th International Conference "Information Control Systems & Technologies" (2020) 371-385.

[6] N. D. Rudnichenko, V.V. Vychuzhanin, N.O. Shibaeva, D.S. Shibaev, T.V. Otradskaya, I.M. Petrov, Software development for interactive big data mining based on machine learning methods. Actual problems of information systems and technologies: monograph (2020) 59-71.

[7] U. Sivarajah, M. Mustafa Kamal, Z. Irani, V. Weerakkody, Critical analysis of Big Data challenges and analytical methods. Journal of Business Research 70 (2017) 263-286.

[8] P. Bala, Introduction of Big Data With Analytics of Big Data. Advanced Deep Learning Applications in Big Data Analytics (2021) 110-125.

[9] W. Lidong, A, Cheryl, Machine Learning in Big Data. International Journal of Mathematical, Engineering and Management Sciences 1 (2016) 52-61. doi:10.33889/IJMEMS.2016.1.2-006.

[10] Q. Junfei, Wu. Qihui, D. Guoru, Xu. Yuhua, F. Shuo, A survey of machine learning for big data processing. EURASIP Journal on Advances in Signal Processing (2016). doi: 10.1186/s13634-016-0355-x.

[11] J. Won, K. Gun, L. Jiwon, L. Hyunwuk, H. Dongho, R. Won, Deep learning with GPUs. Advances in Computers (2021) doi: 10.1016/bs.adcom.2020.11.003.

[12] H. Ryan, K. Panagiotis, P. Charalampos, T. Lazaros, L. Xing, L. George, A Prototype Deep Learning Paraphrase Identification Service for Discovering Information Cascades in Social Networks. IEEE International Conference on Multimedia and ExpoAt (2020). doi: 10.1109/ICMEW46912.2020.9106044.

[13] S. Will, Neural Networks in Big Data and Web Search. Neural networks in web search, 4 7 (2018) doi:10.3390/data4010007.

[14] L. Zichao, J. Xin, S. Lifeng, L. Hang, Paraphrase Generation with Deep Reinforcement Learning. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018) 3865-3878. doi:10.18653/v1/D18-1421.

[15] B. O. Bliznyuk, L.V. Vasilieva, I.D. Strelnikov, D.S. Tkachuk, Modern methods of natural language processing. Visnik of Kharkiv National University of the Name of V.N. Karazin (2017).

[16] W. Qiang, L. Bei, X. Tong, Z. Jingbo, L. Changliang, W. Derek, C. Lidia, Learning Deep Transformer Models for Machine Translation (2019) 1810-1822. doi:10.18653/v1/P19-1176.

[17] V.S. Yuskov, I. V. Barannikova, Comparative Analysis of Natural Language Processing Platforms. Mining information and analytical bulletin 3 (2017) 272-278.

[18] H. Hassan, C. Beneki, S. Unger, M. Mazinani, M. Yeganegi, Text Mining in Big Data Analytics. Big Data and Cognitive Computing 4 1 (2020). doi:10.3390/bdcc4010001.

[19] S. Zhou, Research on the Application of Deep Learning in Text Generation. Journal of Physics: Conference Series 1693 (2020). doi:10.1088/1742-6596/1693/1/012060.

[20] A.O. Soshnikov, The role of the functional-semantic type of speech in the process of automatic processing of natural language. Eurasian Union of Scientists 15 (2015) 134-135.

[21] N. I. Widiastuti, Deep Learning – Now and Next in Text Mining and Natural Language Processing. IOP Conference Series: Materials Science and Engineering, 407 (2018). doi:10.1088/1757-899X/407/1/012114.

[22] H. Li, Deep learning for natural language processing: Advantages and challenges. National Science Review 5 (2018). doi:10.1093/nsr/nwx110.

[23] C. Hou, C. Zhou, K. Zhou, J. Sun, S. Xuanyuanj, A Survey of Deep Learning Applied to Story Generation (2019). doi:10.1007/978-3-030-34139-8_1.

[24] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent Trends in Deep Learning Based Natural Language Processing. IEEE Computational Intelligence Magazine 13 (2018) 55-75.

[25] A.Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). NY, USA (2017).

[26] X. Zhang, M. Chen, Y. Qin, NLP-QA Framework Based on LSTM-RNN (2018) 307-311. doi:10.1109/ICDSBA.2018.00065. 6000–6010.

[27] K. Adam, K. Smagulova, A. James, Memristive LSTM Architectures (2020). doi:10.1007/978-3-030-14524-8_12.

[28] Quora Question Dataset 2020, URL: https://www.kaggle.com/c/quora-question-pairs.

[29] ParaNMT-50M 2020, URL: https://www.aclweb.org/anthology/P18-1042/.

[30] P.D.Kingma, J.Ba, Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).