Text processing using large language models for automatic generation of knowledge bases

Andreev Aleksandr Vladimirovich
Kotsenko Anton Aleksandrovich
Varlamov Oleg Olegovich*
Kim Radmir Ilshadovich
Marushchenko Aleksei Viktorovich
Bauman Moscow State Technical University, 2-ya Baumanskaya Street, 5/1, Moscow, 105005, Russian Federation
*ovarlamov@gmail.com

Abstract

The complexity of subject areas in which intelligent information systems operate is steadily increasing. Tasks assigned to such systems are increasingly focused on automating and robotizing spheres of human activity. Solving such tasks requires adaptive and flexible methods capable of accommodating dynamic changes in the environment in real-time. The mivar approach to creating intelligent decision-making systems enables working with adaptive discrete structures and provides methods for managerial decision-making based on adaptive active logical inference from the mivar rule knowledge base. The mivar logical inference machine forms the core of expert systems based on the mivar approach. As a result of the development of the mivar approach across various subject areas, different versions of mivar logical inference machines with their algorithms for rule traversal in the knowledge base have been created. Recent advancements in artificial intelligence and machine learning have opened new opportunities for enhancing the mivar approach. Integrating large language models for automating text processing in mivar systems significantly enhances the accuracy and efficiency of decision-making processes. This paper demonstrates the feasibility of using automated text processing, intended for human training, through large language models, and its subsequent application in action planning tasks within technical systems. The proposed methodology aims to create extensive knowledge bases based on textual information.

Keywords

mivar, mivar network, artificial intelligence, large language model, DSS, Wi!Mi, Razumator.

INTRODUCTION

Currently, artificial intelligence is developing at a rapid pace, and advanced developments in AI are capable of providing strong competition to established methods. The business sector is actively interested in automating processes involved in the handling of various text documents and regulations. The advancement of automation in solving such tasks opens up vast prospects for the widespread use of artificial intelligence in creating various knowledge bases applied in diverse fields of human activity. An example of using such a methodology can be the creation of mivar knowledge bases in areas where there is insufficient data to create models based on neural networks.

The development of mivar technologies in logical artificial intelligence has been going on for quite some time. These technologies have proven effective in solving tasks within the framework of production networks, as they enable finding solutions with linear computational complexity [1]. Mivar technologies are widely used for creating expert systems in various fields, such as the management of educational programs at universities [2], the detailed description of knowledge in a scientific discipline [3], the detection of bank check fraud [4], intelligent plant care systems [5], decision-making on the safety of thermolabile blood components [6], optimization of the process of preparing fresh frozen plasma for transfusion [7], diabetes diagnosis [8], automated assembly [9], mechanical engineering [10], and many other areas. Additionally, mivar technologies have found their place in the creation of hybrid artificial intelligence (AI) systems. They can be implemented in a wide range of activities, such as creating a brief overview of judicial practice [11], detecting energy theft in smart grids using explainable attention maps [12], using metagraphs to represent data sets [13] to overcome limitations [14] and improve existing knowledge bases [15], for tasks of analysis and classification [16] of equivalent logical operations [17], and solving first-order logical equations with exhaustive search for solutions [18].

The hybrid AI approach with the application of mivar technologies includes neural network methods. Neural networks can be used for sentiment analysis based on text and audio data [19], for processing media information [20] and its optimal encoding [21], for working with satellite images [22] and text queries [23]. Hybrid AI can also be applied in the development of polymer microstructures [24] and composites [25], in modeling heat exchangers [26], as well as in the formation of intelligent technological units [27]. The use of mivar technologies in conjunction with a neural network approach simplifies the process of analyzing LiDAR data for the task of finding trees and estimating their diameter [28], and for measuring active gases affecting the climate at carbon landfills [29]. Mivar technologies are capable not only of decision-making but also of performing intelligent analysis of pulsed EPR for recognizing 3D objects by an optical location system [30]. One of the most promising areas of application of the mivar approach and hybrid AI is the development of robotics [31], as well as the navigation systems of robotic complexes [32], and the creation of intelligent vehicle control systems [33]. One of the complex tasks in this area is pathfinding [34] with planning the shortest route for a robotic complex [35], which is especially important when moving autonomous transport on public roads, where many conditions need to be taken into account [36].

In this work, we will conduct a study of the subject area and analyze existing large language models to create an algorithm for generating a mivar knowledge base. With this algorithm, it will be possible to significantly reduce the time required to process existing knowledge about the subject area in the form of text intended for training specialists or representing an instruction or set of rules and subsequently transferring it to the mivar space. An example of a field of human activity where such a tool could be useful is medicine, where creating sufficiently effective models based on neural networks may be difficult or impossible due to a lack of data for training.

FORMULATION OF THE PROBLEM

The use of large language models (LLMs) for processing text documents is a classical approach based on prompts, known as Retrieval-Augmented Generation (RAG). Before the era of LLMs, models were often supplemented with new data simply by fine-tuning them. However, now that the models being used have become much larger, training them on significantly larger datasets is suitable for only a few usage scenarios. Fine-tuning is particularly effective in cases where the model needs to interact with users using a style and tone of speech different from the original. However, fine-tuning is not as effective when new data needs to be added to the model, which I have found to be a much more common business scenario. Additionally, fine-tuning LLMs requires large volumes of high-quality data, substantial computational resources, and significant time. For most LLM users, these are considered limited resources.

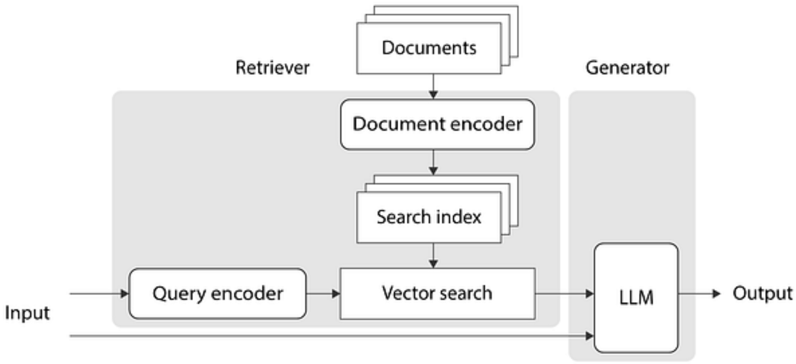

Fig. 1. Architecture of an AI system using RAG.

The main principle involves breaking down large documents that cannot fit into a prompt into smaller parts or chunks. When a question is posed, the system searches for the most relevant pieces of text where the answer might reside. These are then combined into a final prompt. The implementation of this mechanism includes several stages:

1. Break the text into chunks.
2. Convert these chunks into vector form for subsequent selection of the most relevant using cosine similarity.
3. Formulate a prompt from these smaller parts.
4. Send the prompt to the selected model and receive a response.

Thus, to create mivar knowledge bases using a large language model, the following steps are necessary:

1. Select an algorithm for dividing the text into chunks.
2. Choose a tokenizer to obtain embeddings for these chunks.
3. Formulate a prompt.
4. Select a model.

A classic description of this mechanism involves:

- Splitting the text by line breaks, trying to keep words intact, with a limit on the number of characters or tokens.
- For tokenization, using the ada-002 model from OpenAI is recommended.
- Using OpenAI for generating responses.

This approach works well unless there is a need for local solutions or any restrictions on the use of source data, such as trade secrets. Let's take the GPT4-Turbo (10K context) model from OpenAI with the native ada-002 tokenizer as a reference. Based on the task at hand, various models and tokenizers need to be systematically tested to select those closest to the reference. As a result, for comparison, the following were selected:

1. One chunking algorithm.
2. Four types of tokenizers: ada-002, RuBert, YandexEmbeding, RuBert Finetuned.
3. One prompt.
4. Seven types of models: GPT4, GPT3.5, YandexGPT, GigaChat, Saiga, FineTuned Saiga.

TEXT PROCESSING ALGORITHM

In the classical approach, it is suggested to divide the text into paragraphs or, where this is not possible, into words with some overlap. This method works well when each paragraph contains a complete thought, and paragraphs are similar in size and structure. However, in the case of legal documents or medical instructions, some paragraphs may be subsections of larger sections. For example, section 3.3.3 describes cases where temperature transport is prohibited, with sub-sections 3.3.3.1 and 3.3.3.2 detailing specific instances. It is important not to separate such related sections.

Additionally, testing revealed that the longer the chunk, the lower the quality of tokenization—resulting in more generalized tokenization and parameter loss during the final generation of the mivar knowledge base. Therefore, it was decided that the optimal chunk size is not merely a technical decision but should primarily preserve context. As a result, chunk sizes were selected up to 150 words, grouped according to the hierarchy of the document or other source text. A script was developed for this purpose, which analyzes the document structure, creating chunks that are as large as possible within the constraints.

TOKENIZER, PRODUCTS, MODELS

Each chunk needed to be transformed into a vector, which was then stored in PSQL. Various options were considered: cloud services OpenApi (ada-002) and YandexEmbeding (with a limit of 10 requests per second), as well as numerous local options including Word2Vec, RuBert, ruGPT, and even embedding layers from Llama. The cloud services performed well, although Yandex had a low default quota. RuBert-Large proved most suitable for our task, using mean pooling. The hidden layer of Llama turned out to be entirely useless. Models like RuGPT, Fred, and others from ai-forever performed worse.

The prompt size was limited to 10,000 characters, as this size was suitable across all models considered, despite varying numbers of tokens occupied by Russian characters in different models (e.g., 2-3 characters per each of Llama's 4096 tokens or 1 token = 1 character for GPT4).

The study utilized multiple sources: access to OpenAI, YandexGPT, GigaChat, and a powerful 4090 GPU for running and training local models. Llama and Saiga were chosen as base models. For cloud models, the response generation temperature was set to zero to enhance accuracy and reduce response creativity. Access to models was obtained via API.

TEST RESULTS

For testing, 31 queries related to text processing used for creating knowledge bases were used, with the system responding using different combinations of models and tokenizers. Each response was then manually evaluated and recorded in Table 1 with the following grading: correct (green), debatable (yellow), incorrect (red). Questions are listed horizontally, and parameter combinations are listed vertically. Each cell in the table contains a color corresponding to the evaluation of the response.

From the test results recorded in Table 1, it is evident that GPT4-Turbo with the native ada-02 tokenizer showed the best results: 71% of the responses were rated as correct and only 23% as incorrect. The least effective model was GigaChat with the local tokenizer RuBert-Large. YandexGPT's results were slightly better, but the difference was insignificant.

Table 1. Comparison of the received answers

Model	gpt-4-1106-preview	gpt-3.5-turbo-1106	gpt-4-1106-preview	gpt-3.5-turbo-1106	YandexGPT2	YandexGPT2
Tokenizer	Ada-002	Ada-002	RuBert-Large	RuBert-Large	RuBert-Large	Ada-002
Truth	71%	61%	32%	35%	32%	29%
Controversial	6%	3%	13%	0%	6%	26%
False	23%	35%	55%	65%	61%	45%

It is important to note that OpenAI models also performed poorly with the RuBert tokenizer. This confirms the assumption that incorrect selection of text parts (tokenizer error) for prompt formation leads to incorrect answers. In almost all cases where OpenAI models gave incorrect answers, other models also showed poor results.

Based on the initial data obtained, it is clear that efforts should be focused on developing task-specific localized tokenizers and using ada-002 as the reference tokenizer. This highlights the importance of choosing the right tools for processing and preparing text before using language models for the task of creating mivar knowledge bases..

CHOOSING A TOKENIZER

In RAG, the prompt task is dynamically formed. Question (A) is converted into a vector, and the most closely related mathematical vectors of regulation fragments are selected and sorted by decreasing "closeness." A query is then formed from these vectors. For example, for question A, the closest text fragments are P, D, F, G, with the distance between vector A-P being greater than A-D. ADA-02 from OpenAI performs best for this task.

Thus, you can create your own tokenizer and compare which text fragments it produces for the same query. If their sequence matches ADA-02, that's excellent. Otherwise, this error can be measured and analyzed.

As a reference for comparison and finding the best tokenizer, approximately 5000 text fragments of various content were used: sample questions, excerpts from regulations and instructions, etc. From this data, 250 pairs of texts were randomly formed to calculate cosine similarity for the chosen tokenizer. Then, the deviation between the value of the ADA-02 standard and the selected tokenizer was calculated. The fluctuation of this deviation is our parameter. If the tokenizers are identical, then the root mean square deviation between them equals 0.

The root mean square deviation for each point allowed us to determine the accuracy of the chosen tokenizer. RuBert-Large demonstrated an error of approximately 10-11% compared to ADA-02, sbert-nlu-ru-mt showed 9-11%, which is a slight improvement.

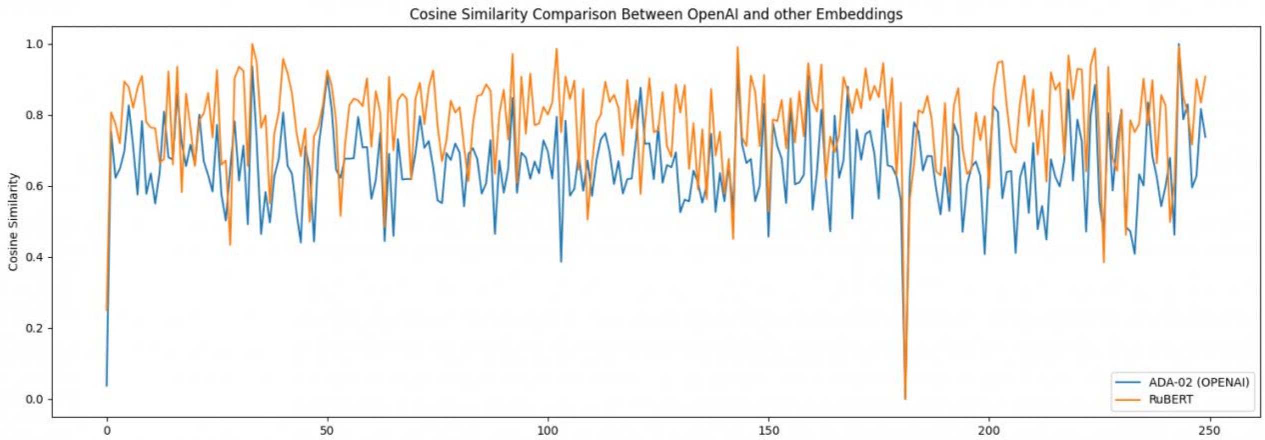

Fig. 2. Comparison of the best local tokenizer with the reference

After analyzing prompts in ChatGPT, a decision was made to use a three-layer TensorFlow network without special tuning of hyperparameters. Over 30 epochs of training, a local adapter was created that converts a RuBert vector (1024 parameters) into an OpenAI vector (1526 parameters). Its use improved the results of using the local tokenizer. Training continued for 30 epochs to create the adapter. It converts a RuBert vector (1024 parameters) into an OpenAI vector (1526 parameters). This adapter will effectively work only with texts closely related to a specific theme.

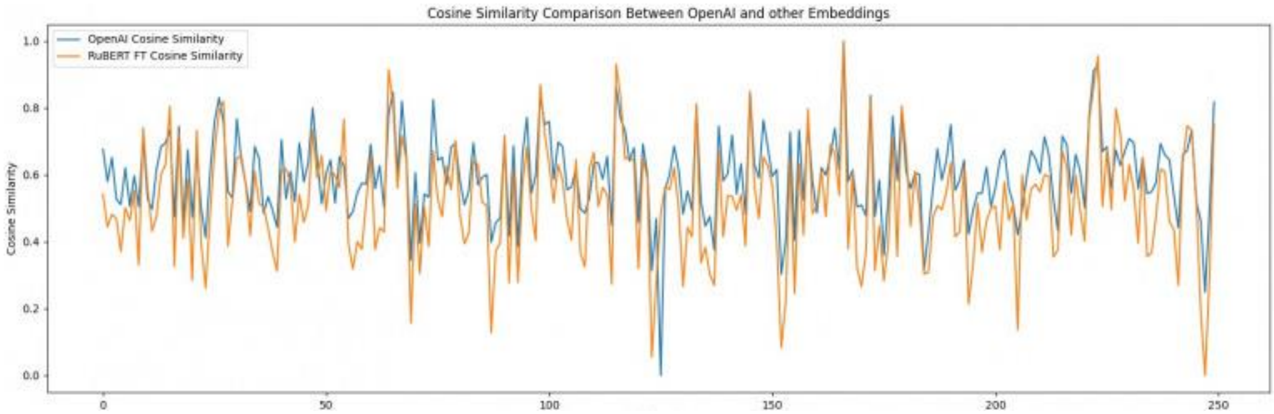

Fig. 3. Comparison of the best local tokenizer with an adapter against the reference

Synthetic tests showed that the error rate decreased from 9-11% to 6%. However, the use of sbert-nlu-ru-mt did not provide any improvement, so RuBert-large was chosen as the basis. Thus, a locally functioning tokenizer was obtained, nearly as effective as the cloud-based ADA-02, theoretically improving the quality of responses.

TESTING IN PRACTICE HOW THE FINETUNED TOKENIZER WORKS

Let's take our standard in the form of GPT-4 and run it through our question/answer machine and give it to experts for evaluation. The results were obtained, as noted in Table 2:

Table 2. Comparison of reference with RuBert – Large for GPT-4 model

GPT-4	ADA-02	RuBert - Large	RuBert-Large FineTuned
Truth	71%	32%	55%
Controversial	6%	13%	19%
False	23%	55%	26%

The values of correct answers on the benchmark model significantly increased (from 32% to 55%). Incorrect answers were halved and equaled the benchmark (reduced from 55% to 26%, with 23% being the benchmark). In other words, our strategy worked:

1. A proper tokenizer improves quality. Enhancing the accuracy of embeddings from 11% to 6% yields a significant result.
2. Fine-tuning of the tokenizer is possible.

We see similar results for YandexGPT models. The number of correct answers increases, and incorrect answersdecrease, as reflected in Table 3.

Table 3. Comparison of reference with RuBert – Large for YandexGPT model

YandexGPT	ADA-02	RuBert - Large	RuBert-Large FineTuned
Truth	29%	32%	45%
Controversial	26%	6%	13%
False	45%	61%	42%

LOCAL MODELS

Let's deploy local models based on Saiga2 (LLama2). The decision not to test with the tokenizer from OpenAI and regular RuBert was justified, as using a local model with a cloud-based tokenizer from OpenAI seemed absurd, and employing regular RuBert was impractical due to its low quality. For comparison, we consider models with two different quantizations: 5-bit and 8-bit. The former can fit on a GeForce 3060 GPU (12GB version), while the latter requires a GeForce 3090 or 4090 GPU (24GB version).

Both models showed nearly identical results, as presented in Table 4: 35% correct answers, 52% incorrect answers, and 13% disputed answers. This is slightly better than YandexGPT and GigaChat based on regular RuBert but inferior to RuBert fine-tuned models. Overall, YandexGPT slightly outperforms Saiga LLama2, which is a positive outcome.

Table 3. Comparison of reference with RuBert – Large for YandexGPT model

Верных ответов						
	gpt-4-1106-preview	**gpt-3.5-turbo-1106**	**YandexGPT2**	**GigaChat**	**Saiga2/LLama2-13BQ8_0**	**Saiga2/LLama2-13BQ5_K**
Ada-02	71%	61%	29%	26%	n/a	n/a
RuBert-Large	32%	35%	32%	29%	n/a	n/a
RuBert-FineTuned	55%	n/a	45%	n/a	35%	32%
Спорные ответы						
	gpt-4-1106-preview	**gpt-3.5-turbo-1106**	**YandexGPT2**	**GigaChat**	**Saiga2/LLama2-13BQ8_0**	**Saiga2/LLama2-13BQ5_K**
Ada-02	6%	3%	26%	13%	n/a	n/a
RuBert-Large	13%	0%	6%	0%	n/a	n/a
RuBert-FineTuned	19%	n/a	13%	n/a	13%	16%
Ошибочные						
	gpt-4-1106-preview	**gpt-3.5-turbo-1106**	**YandexGPT2**	**GigaChat**	**Saiga2/LLama2-13BQ8_0**	**Saiga2/LLama2-13BQ5_K**
Ada-02	23%	35%	45%	61%	n/a	n/a
RuBert-Large	55%	65%	61%	71%	n/a	n/a
RuBert-FineTuned	26%	n/a	42%	n/a	52%	52%

CONCLUSION

Currently, the RAG task using local models (in a closed circuit, without internet access or interfaces to other models) is not fully solvable due to the large number of incorrect answers. This task can be addressed if there is access to cloud solutions from OpenAI, such as GPT-4 with a wide context window, achieving 70-80% correct answers without much effort. The downside, apart from the lack of a security circuit, is the cost. At the time of writing, a request costs about 5-10 cents.

The main limitations of local models are primarily related to the correct formation of primary text parts by tokenizers and, to a lesser extent, to the models themselves. Model shortcomings can be compensated for with more accurate prompts and fine-tuning. Alternatively, one can wait for technological advancements.

Working on data slicing and vectorization is a task that requires attention and will not be resolved without additional research. Fine-tuning the tokenizer seems technically feasible. More experimentation with model hyperparameters is needed.

There are three possible approaches to data slicing (text parts). The first, somewhat primitive but correct from a product perspective, is to limit the RAG task to information that fits into 1-2 paragraphs, possibly with manual slicing. The second is to explore the possibilities and technologies of graph representation of data for more complex processing. The third is to wait for decent local models with a context length (effective length) of up to 128K characters or train them independently. However, this is not our approach.

The quality assessment of the answers was performed manually by experts, giving the study practical significance. One of the key findings is that the model creating the primary embeddings of the document played a significant role in the proportion of quality answers. A good embedding provided more quality improvement than a good LLM.

Automatic creation of knowledge bases using large language models is possible. If cloud solutions are available, using the RAG task can significantly speed up the processing of the source text. However, to create fully automatic knowledge base generation, a comprehensive methodology must be developed, including preprocessing of unstructured text, checking the preservation of information completeness, and testing the resulting model.

REFERENCES

[1] Varlamov, O. "Brains" for Robots: Application of the Mivar Expert Systems for Implementation of Autonomous Intelligent Robots. Big Data Research, 2021, 25, 100241. https://doi.org/10.1016/j.bdr.2021.100241

[2] Guzeva, T., Parsheva, A., Babin, V. at al. Management of Educational Programs at the University Based on Mivar Expert Systems. Lecture Notes in Networks and Systems, 2023, 509, pp. 651–659. https://doi.org/10.1007/978-3-031-11058-0_65

[3] Guzeva, T., Egorov, S., Smetankin, K. at al. Mivar's Approach to Detailed Description of Knowledge for the Academic Subject "Rocket and Space Manufacturing Technologies". Lecture Notes in Networks and Systems, 2023, 509, pp. 643–650. https://doi.org/10.1007/978-3-031-11058-0_64

[4] Grigorev, U., Shashkin, Y., Ploutenko, A., et al., Bank Checks Fraud Detection Based on the Analysis of Event Trends in Data-Flow Systems. International Conference on Internet of Things, Big Data and Security, IoTBDS - Proceedings, vol. 2023-April, pp. 97-104, 2023. https://doi.org/10.5220/0011727300003482

[5] Aladin, D.V., Aladina, E.V., Chuvikov, D.A. at al. Creating a logical intelligent plant care system in digital agriculture based on Mivar approach. IOP Conference Series: Earth and Environmental Science, 2022, 954(1), 012004. https://doi.org/10.1088/1755-1315/954/1/012004

[6] Varlamov, O.O., Chuvikov, D.A., Lemondzhava, V.N. at al. A Software Package Supporting Decision Making on the Safety of Thermolabile Blood Components. Biomedical Engineering, 2022, 55(5), pp. 355–359. https://doi.org/10.1007/s10527-022-10135-0

[7] Lemondzhava, V.N., Lemondzhava, T.Yu., Gudkov, A.G., et al., Technological optimization of the process of preparation of fresh frozen blood plasma to transfusion in devices for its thermal processing. AIP Conference Proceedings, vol. 2605, 2023. https://doi.org/10.1063/5.0110400

[8] Shamaev, D., Synthetic Datasets and Medical Artificial Intelligence Specifics. Lecture Notes in Networks and Systems, vol. 597 LNNS, pp. 519-528, 2023. https://doi.org/10.1007/978-3-031-21438-7_41

[9] Andreev A., Kotsenko A., Kim R. at al. Methodology for automated assembly and delivery of products using mivar expert systems. E3S Web of Conferences, 2024, 549, 08008. https://doi.org/10.1051/e3sconf/202454908008

[10] Volkov, A., Varlamov, O. Method of creation of a two-level neural network structure for solving problems in mechanical engineering. Journal of Physics: Conference Series, 2021, 2131(3), 032003. https://doi.org/10.1088/1742-6596/2131/3/032003

[11] Taran, M.O., Revunkov, G.I., Gapanyuk, Y.E., Creating a Brief Review of Judicial Practice Using Clustering Methods. Studies in Computational Intelligence, vol. 1064 SCI, pp. 466-475, 2023. https://doi.org/10.1007/978-3-031-19032-2_48

[12] Ishkov, D.O., Terekhov, V.I., Myshenkov, K.S., Energy Theft Detection in Smart Grids via Explainable Attention Maps. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086919

[13] Gapanyuk, Y.E., Extension of Metagraph Model on the Basis of Protograph Approach. Pattern Recognition and Image Analysis, vol. 33, no. 3, pp. 300-305, 2023. https://doi.org/10.1134/S1054661823030124

[14] Fedyukin, D., Gromozdov, D., Gapanyuk, Y., Creating a Metagraph Dataset Based on the ATOMIC Dataset. Studies in Systems, Decision and Control, vol. 457, pp. 207-218, 2023. https://doi.org/10.1007/978-3-031-22938-1_14

[15] Tsarev, R., Azizam, S.H., Sablinskii, A., et al., Gamification of the Graph Theory Course. Finding the Shortest Path by a Greedy Algorithm. Lecture Notes in Networks and Systems, vol. 723 LNNS, pp. 209-216, 2023. https://doi.org/10.1007/978-3-031-35317-8_18

[16] Gurianov, D.A., K.S. Myshenkov, V.I. Terekhov, Software Development Methodologies: Analysis and Classification. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086852

[17] Popov, V.S., Equivalence of logical operations and other operations in Python programming language. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086928

[18] Popov, V.S., First-Order Logical Equations with Parameter and their Exhaustive Search Solutions. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086751

[19] Verina, Y., Tolstoukhov, D., Egorov, D., et al., Hybrid model for sentiment analysis based on both text and audio data trained on MELD. AIP Conference Proceedings, vol. 2819, 2023. https://doi.org/10.1063/5.0137948

[20] Kanev, A., Nazarov, M., Uskov, D., Terentyev, V., Research of Different Neural Network Architectures for Audio and Video Denoising. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086862

[21] Belodedov, M.V., Fonkants, R.V., Safin, R.R., Development of an Algorithm for Optimal Encoding of WAV Files Using Genetic Algorithms. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086837

[22] Kanev, A.I., Tarasov, A.V., Shikhov, A.N., et al., Identification of logged and windthrow areas from Sentinel-2 satellite images using the U-net convolutional neural network and factors affecting its accuracy. Sovremennye Problemy Distantsionnogo Zondirovaniya Zemli iz Kosmosa, vol. 20, no. 3, pp. 136-151, 2023. https://doi.org/10.21046/2070-7401-2023-20-3-136-151
[23] Kim, R., Kotsenko, A., Andreev, A. at al. Evaluation of BERT and ChatGPT models in inference, paraphrase and similarity tasks. E3S Web of Conferences, 2024, 515, 03016. https://doi.org/10.1051/e3sconf/202451503016
[24] Kim R., Kotsenko A., Andreev A. at al. Comparison of ChatGPT and Bard for using in hybrid intelligent information systems. E3S Web of Conferences, 2024, 549, 08009. https://doi.org/10.1051/e3sconf/202454908009
[25] Tikhonova, D.S., Abramochkin, A.Y., Borodulin, A.S., et al., A Neural Network Approach to Processing Photos of Carbon-Fiber-Reinforced Polymer Microstructures for Their Further Diagnostics. Polymer Science - Series D, vol. 16, no. 2, pp. 307-312, 2023. https://doi.org/10.1134/S1995421223020442
[26] Valishin, A., Beriachvili, N., Applying neural networks to analyse the properties and structure of composite materials. E3S Web of Conferences, vol. 376, 2023. https://doi.org/10.1051/e3sconf/202337601041
[27] Lavrov, N.A., Khutsieva, S.I., Shananin, V.A., Use of Neural Networks for Dynamic Heat Exchanger Modeling. Chemical and Petroleum Engineering, vol. 58, no. 11-12, pp. 917-924, 2023. https://doi.org/10.1007/s10556-023-01183-8
[28] Bukhtoyarov, V.V., Tynchenko, V.S., Nelyub, V.A., et al., A Study on a Probabilistic Method for Designing Artificial Neural Networks for the Formation of Intelligent Technology Assemblies with High Variability. Electronics (Switzerland), vol. 12, no. 1, 2023. https://doi.org/10.3390/electronics12010215
[29] Grishin, I.A., Terekhov, V.I., Procedure for Locating Trees and Estimating Diameters Using LiDAR Data. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086843
[30] Dyachenko, V.V., Devisilov, V.A., Shemanin, V.G., Lidar Technology for Measuring Climate Impacting Active Gases for Carbon Polygons. Ecology and Industry of Russia, vol. 27, no. 6, pp. 30-35, 2023. https://doi.org/10.18412/1816-0395-2023-6-30-35
[31] Labunets, L., Intellectual analysis of pulsed RCS for 3D objects recognition by an optical location system. Proceedings of SPIE - The International Society for Optical Engineering, vol. 12564, 2023. https://doi.org/10.1117/12.2669242
[32] Varlamov, O., Aladin, D. A New Generation of Rules-based Approach: Mivar-based Intelligent Planning of Robot Actions (MIPRA) and Brains for Autonomous Robots. Machine Intelligence Research, 2024. https://doi.org/10.1007/s11633-023-1473-1
[33] Volgina, A.D., Kirillov, D.S., Kravtsov, A.N., et al., The Robot-Guide for Indoor Navigation. Proceedings of the 2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering, REEPE 2023. https://doi.org/10.1109/REEPE57272.2023.10086707
[34] Aladin D., Kotsenko A., Kim R. at al. Reasons for using parallel activation of logical rules in solving management tasks. E3S Web of Conferences, 2024, 549, 08007. https://doi.org/10.1051/e3sconf/202454908007
[35] Kotsenko, A., Andreev, A., Kim, R. at al. Route planning of autonomous robots in three-dimensional logic space using mivar technologies. E3S Web of Conferences, 2024, 515, 04018. https://doi.org/10.1051/e3sconf/202451504018
[36] Potashnikov, M., Shishkina, V., Muravev, A., Kartashov, A., Development of vehicle driving cycles based on the real traffic dataset. E3S Web of Conferences, vol. 402, 2023. https://doi.org/10.1051/e3sconf/202340204001