




Language Model Architecture Based on the Syntactic Graph of Analyzed Text

Roman Semenov^(✉) 

MIREA – Russian Technological University, Vernadsky Avenue 78, 119454 Moscow, Russia
9629790@gmail.com

Abstract. The methods and techniques of graph structures for text processing are considered. The task of processing Russian-language text and extracting semantic structures is an important stage in the development of artificial intelligence systems. Existing models of intelligent assistants are unable to handle a large volume of noisy information and take a long time to process requests. To solve this problem, the article proposes methods for working with graph structures for the analysis and classification of necessary data. By performing initial processing according to the proposed conceptual structure, it becomes possible to use a syntactic graph for a more accurate representation of each part of speech in the processed context. The results of the tested model provided data on the accuracy of word identification in Russian-language sentences. A table comparing the accuracy with existing natural language processing models is presented. The results were obtained based on the fact that 70% of the text volume is required for the training set, and analysis was conducted on the remaining portion, which is true for each of the compared model.

Keywords: Syntactic Graph · Conceptual Structure · Graphlet · Logical Structure · Actionable Construct · Analyser · Information Noise

1 Introduction

Abstract models are becoming more prevalent in scientific research due to the advancement and innovation of new computing techniques and algorithms. To conveniently present data, graphical methods are used with various visualization tools. Graph theory is one of the primary approaches to describing objects and situations. Graphical representations of models are essential for rapidly comprehending, absorbing, and conveying critical information. In mathematical terms, a graph consists of a finite set of vertex points that can be linked by edge lines. Today, no scientific field can function effectively without using graph models. The field of information technology relies on multifaceted object interactions, making broadcasting information through this kind of method vital [1]. The article will delve into graph structures and situational analysis, which present the foundation for developing models of logical and conceptual representation of proposals.

Understanding textual constructions in the Russian language is an intricate and multifaceted procedure. Most current intelligent assistants rely on conditional structures and

question-answer templates. These rudimentary algorithms underpin rapidly developing systems that now boast vast and regularly updated databases [2]. Artificial limitations in communication with AI reduce interest in interacting with such systems, resulting in the exclusive use of standard phrases. The processing of textual information is currently hindered by information noise, which refers to the excess of trivial information cluttering internet resources. Natural language processing models have advanced significantly and are applied in various fields. There are numerous tasks that analyzers can solve. To tackle the issue of semantic allocation of structures for future use, we employ conditional models, despite their low accuracy. It's worth noting that these methods are limited by specific language rules and conditions. To enhance the identification of further parameters and characteristics for each analyzed structure, it is advisable to employ graphical representations. Graph-based natural language processing models offer an intriguing solution for selecting additional criteria and promptly detecting relevant information.

In connection with the above areas, a number of studies of models and techniques will be conducted, according to which it will be possible to organize an automated analyzer of Russian language sentences based on graph structures.

2 Materials and Methods

2.1 Graph Core and Function Levels

Graph structures are commonly used in machine learning models, representing a specific approach to function development.

The graph kernel is a concept that describes graph behavior in relation to homomorphism. Homomorphism maps data between two graphs, while each graph's structure remains unchanged. It is possible to establish the similarity between graphs, as depicted in Fig. 1. A graph is considered a kernel when every homomorphism is an isomorphism. This allows the performance of selected operations within the graph, regardless of the space surrounding it. For the design of the proposed system, this method is essential to emphasizing the most significant aspects of language. It is crucial to ascertain the potency of every term within its context to produce a relevant and significant solution.

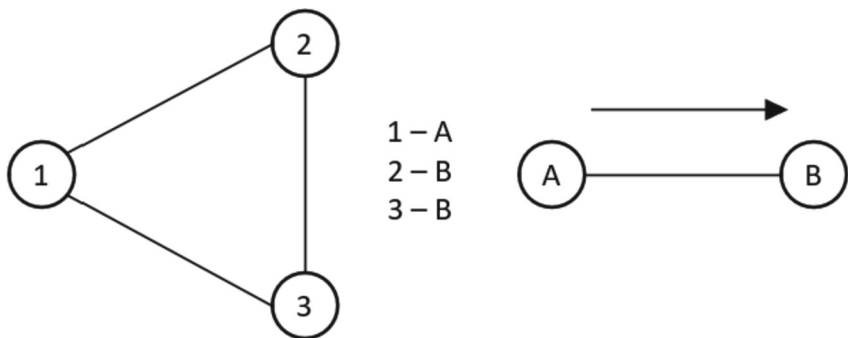


Fig. 1. The example of a graph with homomorphism.

Homomorphism enables simplification of the structure and reduction of adjacent vertices. They constitute a specific aspect of graph representation that emphasizes the most significant features of the subject domain [3]. Subsequently, we will examine graph isomorphism, which is one of the homomorphism instances.

An isomorphism between two graphs represents a one-to-one correspondence between their sets of vertices, such that the adjacency of any two vertices in graph A is the same as that of their corresponding vertices in graph B. An illustration of this concept is provided in Fig. 2. The graphs under consideration may be undirected and lack weights for both their vertices and edges. The isomorphism technique is valuable in processing semantic constructions, as it enables the identification of analogous sentences. It also supports the selection of appropriate data for training neural networks.

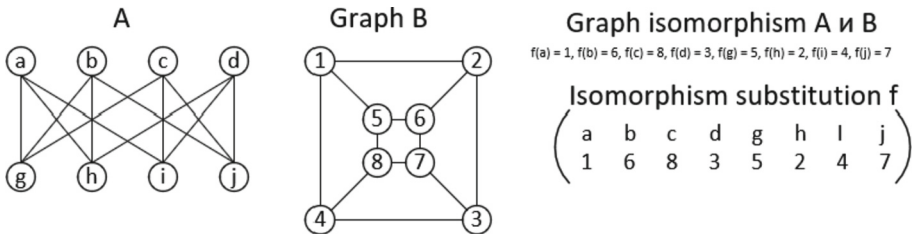


Fig. 2. Isomorphism on the example of a graph.

Isomorphism expresses the similarity or coincidence of two graphs in both directions. The use of homomorphism in algorithms for processing semantic constructions enables detailed processing and identification of the most significant sentences. Applying a preorder and identifying dependencies among vertices of graphs enables the creation of a system that uses predetermined knowledge bases to process incoming requests and provide an answer promptly [4–6]. However, defining the processing area is crucial in tackling the problem of isomorphism, as the algorithm’s search for such isomorphism may take too long and result in no response from the system.

The proposed system employs statistical aggregation at the node level, providing a straightforward method of defining a function at the graph level. The proposed system employs statistical aggregation at the node level, providing a straightforward method of defining a function at the graph level. This compiled information is then utilized to represent the graph at its highest level. Consequently, statistics can be conveniently consolidated at the node level. Frequency histograms and other summary statistics can then be calculated based on word usage, node centrality, and clustering coefficients in the graph. One drawback of this method is its reliance on local node data, potentially omitting critical global properties of the graph.

An iterative neighbourhood aggregation strategy can enhance the fundamental technique with a node set. These methodologies seek to extract node-level functions with more informative content beyond local data. Feature selection functions are combined in the graph-level representation. The aggregation process concludes upon achieving a perfect model, with different parts of speech assigned to each property type.

The Weisfeiler-Lehman algorithm and kernel are two of the most significant and widely recognized aggregation strategies. Typically, this label corresponds to a degree in most graphs, [7–9]. The algorithm’s main concept involves the assignment of an initial label to each node. Subsequently, every node is assigned a new label iteratively. This new label is generated by converting the multisets of current neighbouring node labels into a fixed-length bit string.

$$l^{(i)}(v) = \text{HASH} \left(\left\{ \left\{ l^{(i-1)}(v) \forall v \in N(v) \right\} \right\} \right). \quad (1)$$

When double curly brackets indicate a multiset, the HASH function assigns a unique new label to each unique multiset. The hash function transforms input data, of any length, into an output bit string of a specific length. After several rounds of highlighting properties, a label emerges for each node which summarises the structure of its surrounding area. Consequently, summary statistics can be calculated on these labels as a functional representation of the graph. The Weisfeiler-Lehman kernel is computed by quantifying the dissimilarity between the outcome sets of labels for two graphs. The Weisfeiler-Lehmann kernel is significant from a theoretical viewpoint [10]. To estimate graph isomorphism, one can examine whether two graphs share the same property set after several rounds of the mechanism. This technique resolves some isomorphism difficulties when processing sentence constructions on graphs.

2.2 Graphlet-Based Methods

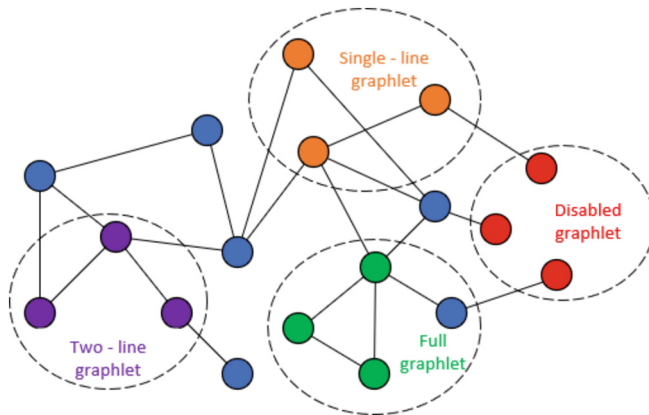


Fig. 3. Types of assigned graphlets.

When analysing functions at the node level, our approach involves counting the frequency of small subgraph structures present in the constructed simulations of each sentence. These structures are known as graphlets and are isomorphism classes of induced subgraphs in a graph. The graphite core formalizes this method by listing all possible graph structures of a specific size and determining the number of occurrences in a

complete graph. Figure 3 illustrates a range of graphlets for comprehending the allotted structures.

This technique encounters issues with intricate graphlet counting, necessitating computational power even after pre-processing each situation with a set of approximations.

2.3 Metric of Similarity of Neighbors

The examined approaches for extracting features or statistics of individual nodes and entire graphs have practical applications in the envisaged model for visually classifying vital parameters. Nonetheless, they possess limitations since they fail to quantify internode relationships [11–14]. This indicates that the aforementioned methods are insufficient for forecasting connections necessary for graphs that can be dynamically scaled and for modelling response sentences textually. The objective is to predict the existence of an edge between two nodes, as depicted in Fig. 4.

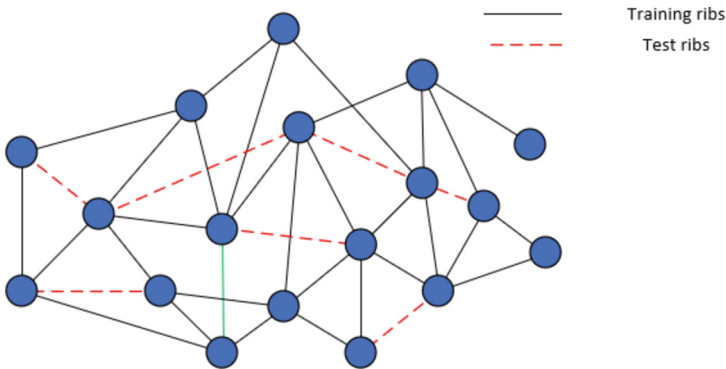


Fig. 4. Illustration of a graph with a subsample for training.

In the figure, the dotted edges on the training graph are removed when training the model or calculating similarity statistics. The model’s performance is assessed based on its aptitude to predict the existence of these evaluation edges. As words can be placed at any position in Russian sentence structures, it is imperative to anticipate the maximum feasible manoeuvres for positioning each node. One way to measure the similarity of neighbourhoods among pairs of nodes is by counting the number of neighbours they share.

$$S[u, v] = |N(u) \cap N(v)|. \quad (2)$$

where $S[u, v]$ represents the value defining the relationship between nodes u and v . To avoid overburdening the model with machine learning techniques, this approach enables the rapid and efficient prediction of most connections in a large data stream. The fundamental principle of the similarity statistics of neighbours $S[u, v]$ is to assume that the probability of an edge (u, v) is proportional to $S[u, v]$ [8].

$$P(S[u, v] = 1) \propto S[u, v]. \quad (3)$$

Thus, to address the issue of prediction of relationships using a measure of similarity of neighbours, you need to set a threshold to determine when the existence of an edge should be predicted.

2.4 Model-Parametric Representation

Graph theory is the most suitable tool for describing and investigating the structure of a space that contains models, parameters, and their relationships. One can refer to the subspace of parameters associated with models or the subspace of models associated with parameters. By using weighted digraphs, we can establish the structure of this space and apply specific actions based on the obtained results. It is feasible to identify additional subspaces within a specified space based on varied traits [15–17].

Based on the image of the model-parametric structure, we will create a model for situational analysis of the information representation using neighbourhood selection. The first-order spaces described above are the neighbourhood of parameters directly linked to the model. It is imperative to assign this neighbourhood as a distinct structure. All components of the model's vicinity, as asserted, are factors and are spaced apart by one unit.

A model-neighbourhood comprises all the model-parametric space elements that are connected via paths. The model neighbourhood boundary consists of space elements that are linked to the model in one direction. Figure 5 illustrates an example of such a space.

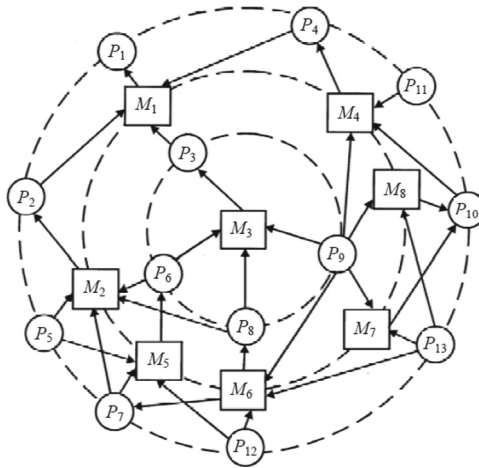


Fig. 5. Model-parametric space relative to the M3 model.

By analogy with model districts, parameter regions exist. Utilising regions is crucial in representing structures for subsequent analysis with graphs [18]. We will refer to those elements belonging to each graph region's neighbourhood as the intersection of the space under review. Figure 6 demonstrates an example of this intersection.

From the example, it is evident that the intersection of neighbourhoods does not result in a neighbourhood. To identify patterns and conduct automated analysis of proposals,

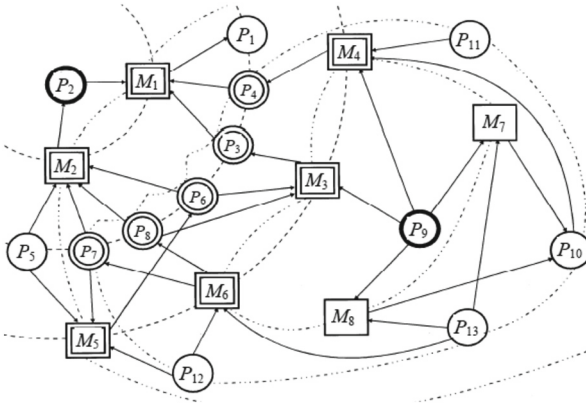


Fig. 6. Results of the operation of intersection of model neighbourhoods.

it is essential to recognise predetermined structures that are logical. Such structures are called graphlets, representing certain situations in the legal space of interaction between models and parameters of the designed environment. The graphlets chosen are demonstrated in Fig. 3.

This method struggles with intricate graphlet counting, requiring substantial computing power, and even pre-processing each situation with a set of approximations. Consequently, we present the foundation of the examined texts. We have produced a model of the abstract representation of our concept. Utilising the graph depicted in Fig. 6, we can establish the structure and connections of sentence parts in each sentence being appraised. According to the graph shown in Fig. 6, it is possible to determine the structure and interrelationships of parts of speech in each sentence under consideration. By comparing the parts of speech in a sentence and representing them in a graph, semantic elements can be combined to highlight graphlets.

When constructing model-parametric neighbourhoods at the logical level, a particular model or parameter is predominant. The rest of the space is relative to this element, and its ordering is carried out with respect to the parameter or model being studied [19–21]. Thus, the model-parametric space is the primary representation when constructing models for Russian language sentences. The neighbourhoods of each subject could serve as potential links between objects and parameters in a given situation.

Each sentence in this text acts as an independent model. The presented architecture suggests that some parameters of adjacent models may coincide or have a similar meaning. As a result, these parameters connect to form a fully connected graph. To highlight the current situations and identify the direction of the sentence and its semantic load, we use constructions predefined in meaning, called graphlets. This perspective can be enhanced and expanded in conjunction with the handling of new resources. The primary trait is the brevity and graphical depiction of the desired models' various versions. Additionally, there is a chance to conceive the required circumstance in any field of study, courtesy of a user-friendly design framework. The exploration of developing a system for presenting textual information using model-parametric models enables the

structuring of text through graphical representation. The architectures designed within the model-parametric space are applicable to software that analyses Russian sentences.

3 Results and Discussion

Based on the previously discussed methods, Fig. 7 demonstrates a created syntactic graph model. The syntactic graph abstractly represents the distribution of parts of speech within a sentence for logical purposes. The interpretation of each part of speech depends on its frequency of use in the text. The frequency of individual word usage is calculated during the training iterations. Afterward, the words are classified by their parts of speech and their frequency characteristics and graphlets are determined. After determining the semantic purpose of the words, the identifier of the logical role in the sentence is assigned.

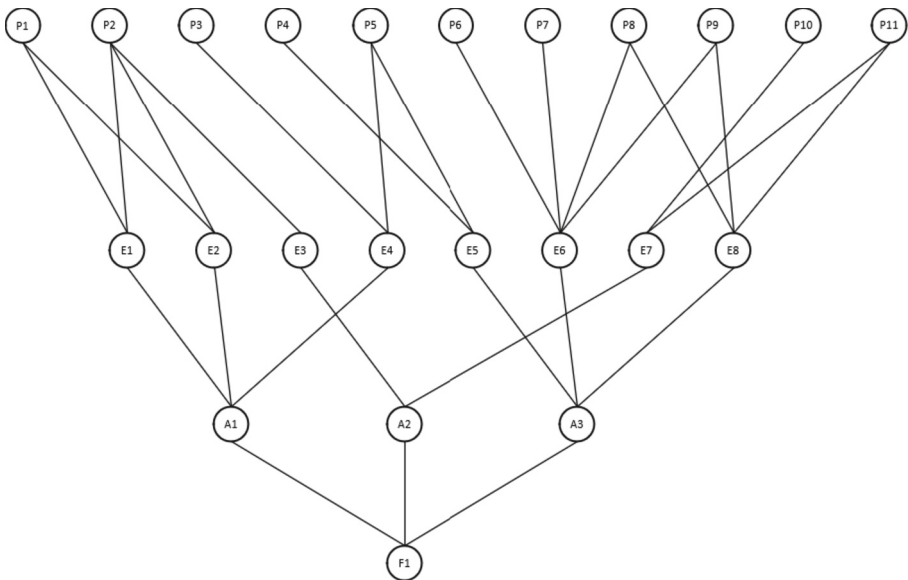


Fig. 7. Syntactic graph: In Fig. 7, P1 – noun; P2 – pronoun; P3 – verb; P4 – participle; P5 – adverb; P6 – adverb; P7 – preposition; P8 – particle; P9 – conjunction; P10 – adjective; P11 – numeral; E1 – subject of action; E2 – object of action; E3 – component of action; E4 – action; E5 – interaction; E6 – relation; E7 – set of properties; E8 – relation; A1 – main idea; A2 – description; A3 – logical structure. F1 is the resulting expression. The resulting expression will be an entire sentence. It is divided into valid constructions by meaning: the main idea (A1), description (A2), logical structures (A3). Each construction expresses a specific element of the graph (E1 – E8). The element is a representation of a part of speech (P1 – P11).

It is essential to handle each active construction in distinct ways and algorithms of varying complexities, depending on the text's saturation. The existing design indicates an element or elements of the conceptual structure displayed in Fig. 8.

The element portrays specific parts of speech in a sentence that are categorized for further scrutiny. The demonstrated interaction enables one to deconstruct propositions

into simpler constructions for easy handling. Additionally, this construction enables the generation of straightforward spoken expressions through a trained model. This has led to the suggestion of combining the produced statements into complete sentences based on the structure of the syntactic graph.

In order to verify the obtained graphs, the existing models of natural language processing were redesigned. The introduction of preliminary results showed that when using a syntactic graph, additional connections appear based on the graph of the conceptual structure. These improvements allow you to get a better result when processing small texts. To improve the accuracy of algorithms for analysing large texts, other, more significant changes should be made. Comparison with existing models of natural language processing was carried out using software implementation of processing algorithms and implementation of a modified model with additional criteria. When compiling the modified model, graph isomorphism, graphlets and the metric of similarity of neighbours, which were described in the second section, were taken into account. This allowed us to achieve better results when used on a text smaller than 10000 words. The test results are shown in Table 1.

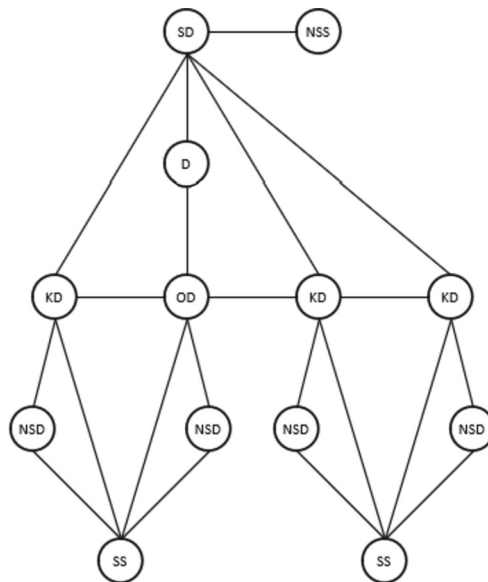


Fig. 8. Graph of the conceptual structure:

In Fig. 8, NSS is the set of properties of the subject; SD is the Subject of the action; D is the Action; OD is the Object of the action; KD is the Component of the action; NSD is the Set of properties of the action; SS is the ratio of properties.

The results shown in Table 1 demonstrate the outcomes of testing various natural language processors on Russian text. In order to compare results across other languages, it would be necessary to adjust the model and metrics accordingly; therefore, this study solely considers and tests architecture in Russian. The table data demonstrates that the

Table 1. Implementation of the results obtained.

The volume of the text Model	5000–10000 words, % accuracy	15000–50000 words, % accuracy	60000–150000 words, % accuracy
RoBERT	54.2	59.1	72.4
DistilBERT	56.3	57.2	73.9
XLM - RoBERTa	63.3	73.9	85.8
Modified model	69.6	54.4	57.3

modified model based on the syntactic graph delivers the best results in processing short text. With medium and large text, the accuracy is significantly reduced, for various reasons. There is an oversaturation of certain words and speech constructions, and words that are rarely used are lost against their background. Large speech models use many factors and conditions to avoid such situations.

4 Conclusions

As a result, it can be concluded that the use of various methods of simplifying complex structures allows you to unload the analysed model to speed up the processing time of information. A unique graph model of the distribution of significant sentence structures for subsequent analysis and textual modelling, called a syntactic graph, is proposed. The methods discussed in the article, at the initial stage of processing, will allow sorting words and phrases in text constructions according to a variety of criteria. The syntactic graph is modelled in accordance with the speech turns of the Russian language and allows you to uniquely determine the role of a specific part of speech in the context. Natural language processing models are currently very diverse and are used in various fields. There are many types of tasks that are solved by different types of analysers. To solve the problems of semantic allocation of structures for further work with them, conditional models are used, which have low accuracy. Also, such methods of obtaining a result are very limited by the rules that are capable of searching for speech constructions of a certain language under certain conditions. To increase the number of additional parameters and characteristics of each analysed structure, graphical representations of information can be used. Graph-based natural language processing models can be an interesting solution when selecting additional criteria and quickly detecting the necessary information.

According to the results of the research, it was possible to identify the optimum conceptual structure, upon which an algorithm for processing semantic constructions without utilising condition models was developed. The result of checking the designed model shows that the utilisation of a syntactic graph in the distribution of parts of speech at the initial stage of analysis enables you to identify logical constructions. Based on the obtained patterns, additional criteria are assigned to each word in order to enhance the accuracy of further text determination. However, as the number of processed words increases, the model's accuracy declines as words that are infrequent but still relevant to a particular text are assigned less weight. According to the results of the research, it

was revealed that the developed model can only work with texts in Russian because the logical constructions and features of constructing sentences in other languages will be very different. The rules that are included in the algorithm must be adjusted according to the syntax of the specific language under study. Further improvements of this algorithm allow you to determine the necessary structure for a different language and type of narrative. The designed analyser will become part of the system of semantic information processing and modelling of text structures. The architecture of further processing may differ from existing models to achieve the best results. Thus, using many simple methods of data ordering and analysis, you can get a fast-automated tool for working with text.

References

1. Anferov, M.A.: Genetic clustering algorithm. *Russ. Technol. J.* **6**(7), 134–150 (2019)
2. Sorokin, A.B., Lobanov, D.A.: Conceptual design of intelligent systems. *Inf. Technol.* **1**(24), 3–10 (2018)
3. Khurana, D., Koli, A., Khatter, K.: Natural language processing: state of the art, current trends and challenges. *Multimed. Tools Appl.* **82**, 3713–3744 (2023)
4. Sorokin A.B., Smolyaninova V.A.: Generalized integrated characteristic base of modular number system **9**(23), 634–641 (2017)
5. Krasnikov, K.E.: Mathematical modeling of some social processes using game-theoretic approaches and making managerial decisions based on them. *Russ. Technol. J.* **9**(5), 67–83 (2021)
6. Zhang, X., Zhao, H., Chen, D.-Y.: Semantic mapping methods between expert view and ontology view. *J. Softw.* **31**(9), 2855–2882 (2020)
7. Sydorenko, V., Kravchenko, S., Rychok, Y., Zeman, K.: Method of classification of tonal estimations time series in problems of intellectual analysis of text content. *Transp. Res. Procedia* **44**, 102–109 (2020)
8. Sorokin, A.B., Zheleznyak, L.M., Suprunenko, D.V., Kholmogorov, V.V.: Designing modules of system dynamics in decision support systems. *Russ. Technol. J.* **10**(4), 18–26 (2022)
9. Anferov, M.A.: Algorithm for searching subcritical paths on network graphs. *Russ. Technol. J.* **11**(1), 60–69 (2023)
10. Tomashevskaya, V.S., Yakovlev, D.A.: Methods of processing unstructured data. *Russ. Technol. J.* **9**(1), 7–17 (2021)
11. Tatur, M.M., Lukashevich, M.M., Pertsev, D.Y., Iskra, N.A.: Data mining and cloud computing. *Doklady BGUIR* **6**(124), 62–71 (2019)
12. Sobolevsky, S., Belyi, A.: Graph neural network inspired algorithm for unsupervised network community detection. *Appl. Netw. Sci.* **7**(63) (2022)
13. Kochkarov, R.A.: Research of NP-complete problems in the class of prefractal graphs. *Mathematics* **9**(21), 2764 (2023)
14. Liu, B., et al: Graph neural networks in natural language processing: a survey. *Now Found. Trends* (2023)
15. le Gorrec, L., Knight, P.A., Caen, A.: Learning network embeddings using small graphlets. *Soc. Netw. Anal. Min.* **12**, 12–20 (2022)
16. Wang, H., Li, J., Wu, H., Hovy, E., Sun, Y.: Pre-trained language models and their applications. *Engineering* **25**, 51–65 (2023)
17. Zhou, J., et al: Graph neural networks: a review of methods and applications. *AI Open* **1**, 57–81 (2020)

18. Belov, S.D., Matrelova, D.P., Matrelov, P.V., Korenkov, V.V.: Review of methods of automatic text processing in natural language. *Syst. Anal. Sci. Educ.* **3**, 8–22 (2020)
19. Sadovskaya, L.L., Guskov, A.E., Kosyakov, D.V., Mukhamediev, R.I.: Text processing in natural language: a review of publications. *Artif. Intell. Decis.-Mak.* **3**, 66–86 (2021)