УДК 528.013 **DOI** 10.34757/2413-7383.2023.30.3.001

Kharlamov Alexander [1, 2, 3, 4], Samaev Eugeny [5], Kuznetsov Dmitry [6], Pantiukhin Dmitry [3, 7]
[1]Institute of Higher Nervous Activity and Neurophysiology, RAS, Moscow; kharlamov@analyst.ru
[2] Moscow State Linguistic University, Moscow
[3] HSE University, Moscow
[4] Moscow Institute of Physics and Technology, Moscow Region, RF
[5] NPP Garant-Service-Universitet, Moscow
[6] Positive Technologies, Moscow
[7] MIREA - Russian Technological University, Moscow

SEMANTIC TEXT ANALYSIS USING ARTIFICIAL NEURAL NETWORKS BASED ON NEURAL-LIKE ELEMENTS WITH TEMPORAL SIGNAL SUMMATION

Харламов Александр[1, 2, 3, 4], Самаев Евгений[5], Кузнецов Дмитрий[6], Пантюхин Дмитрий [3, 7]
[1]Институт высшей нервной деятельности и нейрофизиологии РАН, Москва; kharlamov@analyst.ru
[2]Московский государственный лингвистический университет, Москва
[3]Университет ВШЭ, Москва
[4]Московский физико-технический институт, Московская область, РФ
[5]АЭС "Гарант-Сервис-Университет", Москва
[6]Позитивные технологии, Москва
[7]МИРЭА - Российский технологический университет, Москва

СЕМАНТИЧЕСКИЙ АНАЛИЗ ТЕКСТА С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ НА ОСНОВЕ НЕЙРОПОДОБНЫХ ЭЛЕМЕНТОВ С ВРЕМЕННЫМ СУММИРОВАНИЕМ СИГНАЛОВ

Text as an image is analyzed in the human visual analyzer. In this case, the image is scanned along the points of the greatest informativity, which are the inflections of the contours of the equitextural areas, into which the image is roughly divided. In the case of text analysis, individual characters of the alphabet are analyzed in this wayNext, the text is analyzed as repetitive language elements of varying complexity. Dictionaries of level-forming elements of varying complexity are formed, the top of which is the level of acceptable com-patibility of the root stems of words (names) in sentences of the text, that is, the semantic level. The level of semantics represented by pairs of root stems is virtually a homogeneous directed semantic network. Re-ranking the weights of the network vertices corresponding to the root stems of individual names, as occurs in the hippocampus, makes it possible to move from the frequency characteristics of the network to their semantic weights. Such networks can be used to analyze texts that represent them: one can compare them with each other, classify and use to identify the most significant parts of texts (generate abstracts of texts), etc.

Keywords: text analysis; language model; neural network; transformer model; semantic analysis of texts; artificial neural networks based on neurons with temporal summation of signals; language levels; semantic level; TextAnalyst technology for semantic text analysis; applications.

Текст как изображение анализируется в зрительном анализаторе человека. При этом изображение сканируется по точкам наибольшей информативности, которые являются перегибами контуров эквитекстурных областей, на которые грубо разбивается изображение. В случае анализа текста таким образом анализируются отдельные символы алфавита. Далее текст анализируется как повторяющиеся элементы языка различной сложности. Формируются словари уровнеобразующих элементов различной сложности, вершиной которых является уровень допустимой сочетаемости корневых основ слов (имен) в предложениях текста, то есть семантический уровень. Уровень семантики, представленный парами корневых основ, представляет собой однородную направленную семантическую сеть. Переранжирование весов вершин сети, соответствующих корневым корням отдельных имен, как это происходит в гиппокампе, позволяет перейти от частотных характеристик сети к их семантическим весам. Такие сети можно использовать для анализа текстов, которые их представляют: сравнивать их между собой, классифицировать и использовать для выявления наиболее значимых частей текстов (генерировать рефераты текстов) и подобное.

Ключевые слова: семантический анализ текстов; искусственные нейронные сети на основе нейронов с временным суммированием сигналов; языковые уровни; семантический уровень; языковая модель; нейронная сеть; трансформаторная модель; технология TextAnalyst для семантического анализа текстов; приложения.

1. Introduction and language models survey

Today, neural networks show resounding success in the analysis and synthesis of natural language, both written and spoken. One can have academic discussions with a computer assistant, quickly find the information needed or even voice own text in the voice of your favorite celebrity. Such advances have been made possible by neural network models obtained in natural language modeling that are able to learn from examples. The main task of language models is to predict any linguistic units, be it the next word in a sentence, the next character, part of speech, etc. To make such a prediction possible, it is necessary to split the text into a sequence of such units (their numerical representation is called tokens), group them with each other in some dependencies and evaluate the neural network confidence levels for the units of interest to us [1]. With the right dataset, models can be trained to make such predictions making the most of their capabilities. And an already trained model, if it has really learned, is able to predict new data. Almost all tasks of natural language processing are thus reduced to the task of processing a numerical sequence

1.1. Data sources for training language models

Language is a complex phenomenon; that’s why really good language models are large. They contain billions of trainable parameters [2]. To train and evaluate the performance of language models, large datasets are needed that contain text in natural and (sometimes) artificial languages. The sources for these texts are:

- digitized books: for example, BookCorpus [3] set of 11 thousand books (5Gb) or Project Gutenberg [4] of about 70 thousand books of various genres.
- Online encyclopedia Wikipedia [5], the feature of which is the presence of similar texts (21Gb) in a large number of languages, which is essential for creating multilingual models.
- Links and texts of the largest social network Reddit: these include the publicly unavailable set WebText [6] and the publicly available OpenWebText [7] (38Gb)), as well as the constantly updated PushShift.io [8] (2Tb) that also has convenient tools for data search, aggregation and analysis.
- and, of course, Internet web-pages collected, for example, in the CommonCrawl database [9] containing about a petabyte of data, although of low quality. Filtered versions of this base can already be used for training language models, such as the C4 multilingual set [10] (800Gb) or REALNEWs [11] (120Gb), etc.

It should be noted that language models for English have the best quality; but for other languages, there are also language models and data sets for their training, for example, the National Corpus of the Russian language [12].

For more specific tasks, own sets can be used; for example, to simulate an artificial programming language, one can use the corresponding GitHub or StackOverflow code storages, on the basis of which the BigQuery data set [13] was created, containing common code parts in various programming languages. And for modeling the “chemical” language (the language of molecular compounds), the ChEMBL database [14] can be used, which allows predicting and synthesizing new chemical compounds [15]. However, we note that the main progress of language models is observed in the field of natural language processing, while the rest of the areas inherit the developments obtained in this field.

1.2. Basic blocks of neural network language models

Any neural network (and language models are no exception) consists of interconnected computational layers.

Three broad classes of neural network architectures are used to process language sequences: these are convolutional neural networks, recurrent networks and attention-based networks, which are often used together.

Below is a brief description of such architectures without details; for selected architectures, simulation results on the WikiText-103 set are provided. Perplexity and (less often) other metrics for assessing the performance of language models are provided. The definition and method of computation of these metrics can be found in detail in [16]. To date, the minimum achieved perplexity on this set is 10.6 units.

1.2.1. Convolutional neural network

A convolutional neural network is based on layers that perform the operation of convolution of the input array with a kernel – an array of trainable parameters. Unlike a dense layer, which assigns its own weight (training parameter) to each input element, in convolutional layers, the weights are reused for different inputs and the kernel has much fewer training parameters. This significantly simplifies the training of convolutional neural networks in comparison with fully connected ones. Traditionally, two-dimensional convolutions are considered for image processing tasks, since the two-dimensional representation is natural for them. But in text or speech processing, one-dimensional convolutions are more often used, since the convolution there is performed along the time axis. Such networks are called Temporal Convolution Networks (TCNs). Technically, they do not differ from any other types of convolutions and, of course, use all the advantages of the latter [17]. In particular, padding techniques [18] are widely used, when input arrays are supplemented with dummy items, so that the output array is of the required size (after all, the convolution operation reduces the size of the array). The stride technique enables skipping some possible outputs of the convolution, thus reducing the computation volume, but not greatly affecting the accuracy of the solution.

An important technique that has become widespread in text and sound processing tasks is dilated convolution, when not all input elements in a row are taken to perform the next convolution step, and gaps can occur, for example, when every second or fourth element is taken. The main advantage of this technique is that now the output element is responsible (depends on) for a longer time window, requiring a significantly smaller computation volume. Distributions with a power of two [19] are especially widely used for speech processing. An important difference between TCN networks and classical convolutions is that they use only casual convolutions, i.e., those, in which the current element of the output sequence is predicted depending only on the current and past elements of the input sequence, but not future ones. This is achieved by shifting the indexes of the elements in the sequences.

In general, convolutional networks turned out to be quite successful in natural language modeling tasks; thus, the GCNN-8 convolutional network [20] with an additional “gated” layer that regulates the amount of attention to each input element showed in 2017 a perplexity slightly higher than 30 units, which competes with the results of recurrent networks.

1.2.2. Recurrent neural network

In a recurrent network, neurons or layers of neurons also accept as inputs the previous outputs of the neural network or its other layers, memorized from past time steps. In recurrent networks obtained from ordinary fully connected layers of neurons, it is difficult to evaluate the gradient of the error function by parameters necessary for network training. The gradient decreases significantly (or vice versa, increases significantly) over time, when moving to subsequent time steps. To deal with this phenomenon, gated recurrent neural networks [21] are used: LSTM (long short-term memory) [22], GRU (Gated recurrent unit) [23] and the like.

In recurrent gated networks, the factor of forgetting information from previous time steps, as well as the factor of adding new information for the current step, is regulated through the use of multiplier gates. The forgetting gate receives input, i.e. past information and multiplies it by a factor from 0 to 1, thereby simulating forgetting. This factor itself is trainable: it is the output of a conventional fully connected neuron with an activation function of the sigmoid type [24]. By changing the weight coefficients of such a neuron during the training process, the forgetting factor is also changed (i.e., is regulated). If the factor turns out to be close to one, then past information is practically not forgotten, and the gradient will be relatively large. Gates for adding new information from the current time step are similarly arranged. Various combinations of gates produce various gate networks; thus, 4 different trained neurons for gates are used in classical LSTM networks, and 3 in GRUs. There are thousands of modifications of such gate networks [25]; however, in practice, variations of LSTMs and GRUs are usually used (including bidirectional ones), when the sequence is processed simultaneously both left-to-right and right-to-left, which improves processing in cases where there is a connection between current samples and subsequent ones (and this is obviously the case for text or speech sound), for example, Bi-LSTM networks [26].

The version of the LSTM network with additional changes in the activation function and learning rules in [27] showed a perplexity of less than 30 units, reducing it by 18 points compared to the unmodified LSTM [28], which was the minimum published value for that period of time (2018), while the convolutional variants (namely TCNs) of the published language models of this period [29] yields a perplexity of about 45 units.

1.2.3. Attention mechanism and transformer type networks

In convolutional networks, the current element of the output sequence is affected by a small number of past elements of the input sequence. This holds true for recurrent networks as well, but a few more past elements of the output sequence are added to this. But in practice this is not always the case. Some samples do not affect others at all (such as silence in sound); some have a strong effect only on adjacent samples, and others on far-distant samples. It would be desirable to model such influence directly. For this, the attention mechanism [30]: was proposed: each sample is evaluated in terms of the strength of its connection with all other samples. This mechanism is trainable, so it is possible to regulate, through training, the influence of samples on each other. Such influence of samples (the so-called “attention”) can be established both between different sequences and for samples of the same sequence (self-attention). A variant of the self-attention mechanism implementation can be briefly described as follows: elements of the input sequence (these are vectors) are converted into three vectors Key, Query and Value through multiplying by trainable matrices. Then the scalar products of the Query from one sample with all the Keys of other (including the current) samples are found; with softmax activation, these numbers are converted into numbers in the range from 0 to 1: these are the degrees of influence of the samples on the current one. The Value vectors of all samples are multiplied by these degrees of influence and added together to obtain the current element of the output sequence. This is repeated for all samples of the input sequence. Such a mechanism transforms the input sequence into an output sequence, where the values of the influence of samples are taken into account, and the transformation matrices are trainable and can be adjusted to the sets of training data. Then these transformed sequences are used to solve recognition or synthesis problems.

Based on the attention mechanism, the architecture of the neural network Transformer [30] was created, which consists of two connected parts: an encoder and a decoder. During the training process, the Encoder receives an input sequence (for example, a text in one language), processes it with a self-attention mechanism, and the resulting sequence enters the Decoder. The Decoder receives the output sequence (for example, a text in another language; it is known during training), processes it with a self-attention mechanism, and processes the resulting sequence with a mutual attention mechanism with the output of the Encoder. For the resulting sequence, the confidence levels of tokens (sample values) are evaluated, which, during training, should have the maximum value for those tokens that actually occurred in the output sequence in the current sample. Once trained, such architecture can predict the most likely token of the output sequence and thereby build it. It showed good performance in the tasks of processing text, sounds, etc.

Transformer-like networks have become so effective and popular that today almost all language models are based on transformers [31], the number of variations of which is huge. A large review of such models can be found in [32].

In 2018, the use of a transformer-like network with additional tricks for encoding words [33] led to a significant jump in quality: perplexity decreased from 30 to 18 units.

1.3. Pre-trained models

The language models are huge and require huge computational costs. It is not possible for each researcher to train such models from scratch; instead, they use the transfer learning approach, when a once trained model is reused to create and retrain their models. Large companies or communities that can afford to train huge models make them available, and all other researchers modify these models in various ways to suit their needs. Among the largest model repositories, we note Hugging Face[1] , which provides a large number of language models pre-trained for various languages and on various datasets and provides tools for creating and training additional models.

1.4. Improving models

High results are achieved not only by changing the architecture of neural network models, but also by increasing the variety of approaches, methods and data.

Below are ideas already implemented in the best architectures that are useful for improving language models.

It is possible to combine supervised and unsupervised training procedures. Thus, for example, in the BERT network [34] based on the Transformer encoder, some of the input tokens are masked and the network learns to recover them. For such a procedure, only the

[1] https://huggingface.co/

texts themselves are needed, without partition into classes. Then, the already partially trained network can be further trained to solve other problems, for example, determining the sentiment of the text; to do this, one only needs to add on several layers to it and retrain them on the already marked-up (partitioned) data set [35].

Models can be trained to solve several problems at the same time, for which a certain common part is allocated in the network, as well as a lot of add-ons for solving selected problems with a composition error function. However, for the task of language modeling, if we represent these tasks as predicting a certain sequence of characters, then the architecture does not need to be changed: it suffices to change the encoding, i.e. interpretation of these characters. For example, the task of translating a text is formulated as the sequence ("Translate into English", "text in English", "text in Russian"), and the task of answering a question for a given text is encoded as the sequence ("answer the question", "text", "question", "answer"). In the context of the model, all this is represented as a sequence of numbers. This is how, for example, the GPT-2 model [36] works, which is able to work with any sequences and showed a perplexity of 17.5 for the WikiText-103 set, although it was trained on a different WebText set. Similarly, one can create multilingual models that are trained to model multiple languages at the same time.

The main area for improving language models is to increase the diversity of data, and, therefore, their volume, which leads to the fact that models must be large and contain billions of trainable parameters. Their implementation requires high-performance computers and special architectures that could take into account the limitations of computers and parallelize the process into many such computers. A similar approach is presented in the Megatron-LM[2]tool [37], with the help of which transformer-like models can be effectively parallelized into many computers and achieve a performance of ~ 15 petaflop/s for large models (BERT and GPT-2), which in turn led to a decrease of perplexity down to 10.8 units.

Such models will only increase in size; today, there are models with a trillion trainable parameters, for example, [38] and [39]; but they are, of course, proprietary and unavailable to ordinary researchers. In this context, research on the creation of computers specially designed for massively parallel operations looks very promising; we note the Cerebras-GPT project [40], in which a GPT-like network with tens of billions of parameters was trained on a special Andromeda AI Supercomputer[3] with a performance of about 120 petaflop/s.

2. Text processing by human

The processing algorithm of texts analysis in the human mind looks as follows. In the process of learning to read, humans consistently learn to recognize characters of the alphabet, words and sentences.

2.1. Structural processing of information in cortical columns

The columns of the cerebral cortex of the human brain perform structural processing of information with the formation of hierarchies of dictionaries of images of events of various degrees of complexity of various modalities [41, 42].

[2]https://github.com/NVIDIA/Megatron-LM

[3] https://www.cerebras.net/andromeda/

2.1.1. Formation of dictionaries

The corpus of texts is subjected to statistical analysis, as a result of which its vocabulary components of various levels are revealed.

Fig. 1 shows multi-level hierarchical structure of event image dictionaries of the same modality, where each level has many parallel sub-dictionaries connected with sub-dictionaries of the next level according to the “each-with-each” pattern. Each level forms a system of subdictionaries $\{\grave{B} i \}jkm$. Here i is a word in the subdictionary, j is the number of the subdictionary at the level, k is the number of the level, and m is the number of the modality.

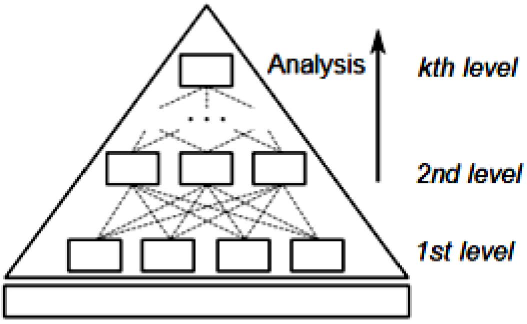

Figure 1. Multi-level hierarchical structure of event image dictionaries of the same modality.

In the process of learning, humans gradually move from recognizing fragments of a separate character of the language alphabet – elements of the first level dictionary $\{Bi \}1$ to its perception as a whole – an element of the second level dictionary $\{Bi \}2$, a dictionary of alphabetic characters is formed in their mind, that is, a dictionary of the graphematic level. Next, humans learn to use the characters of the alphabet in their sequence in the word. In their mind, a dictionary of inflectional structures of the word is formed – elements of the third level dictionary $\{Bi \}3$, a dictionary of the morphemic level. Further, humans learn to use inflectional structures in their sequence in a word along with root stems. In their mind, a dictionary of the root stems of the word $\{Bi \}4$ is formed, that is, a dictionary of the lexeme level. After achieving certain skills in the perception of words, humans begin to perceive words as a single symbol based on the root stem.

After that, two more levels of dictionaries are formed: a dictionary of inflectional structures of syntactic groups - elements of the fifth level dictionary $\{Bi \}5$, which ensures the correct use of the text form, that is, a dictionary of the syntactic level; and then, at the semantic level, a dictionary of acceptable pairwise compatibility of root stems, that is, a dictionary of the semantic level $\{Bi \}6$.

After the process of perceiving words as whole symbols is completed, humans learn to perceive words as symbols in their sequence in a sentence: that is, humans begin to perceive the sentence as a whole.

2.1.2. Formation of a semantic network

The dictionary of pairwise compatibility of the stems of words is actually already a semantic network, since pairs of words are combined by their identical words. In this case, a directed graph is formed, where the chains can have branches. The subsequent re-ranking completes the process of building a semantic network, when we move from the frequency portrait of the text to its semantic portrait (with weighted nodes and connections).

2.2. Formation of situation templates in hippocampal lamellae

In addition to the cerebral cortex, another brain structure that is essential for the formation of the semantic network is the hippocampus. Hippocampal lamellae (sections orthogonal to the long axis of the hippocampus) are responsible for storing information about the relationships of event images stored in cortical columns within the framework of entire situations. The pyramidal neurons of the CA3 field of the p-th lamella of the hippocampus form an artificial Hopfield neural network [43], the weights of the synapses of which store information about the association of images of events stored in the cortical columns, related to a particular situation, within this situation (eq.1).

$$Np = \cup i \; Bi \; . \qquad (1)$$

The caret character [$\hat{.}$]above the corresponding events in (1) is absent because in the hippocampal lamellae it is not the dictionary elements that are combined into a network (fragments of trajectories in a multidimensional space), but their text equivalents, indices.

Hippocampal lamellae receive information from the cortex columns [43]; here the associative principle of addressing information also works. The entire stream of information coming to the hippocampus through some switches from the cortex, comes at the same time to all the lamellae of the hippocampus. But only those lamellae respond to this stream that contain information about events, the images of which are present in the input stream. The response is the greater, the stronger the association and the greater the weight of the images of events in the cortical columns.

At each iteration of the interaction between the cortex and hippocampus, the CA1 field of the hippocampus (as a competitive network) generates the response of only one (or given number of) hippocampal lamella that is the closest to the input situation.

But this is not the end: as a result of the response of the current lamella of the hippocampus in the cortex in the column that initiated the process, additional training occurs (as a result of the so-called long-term potentiation [44]. And at the next iteration, the associative projection of the same situation on the hippocampal lamellae is changed due to this additional training, and the next response of the lamellae changes.

After 15 to 20 iterations, the images of events included in the situation in the cortex column will change due to additional training, which is initiated by the models of situations stored in the lamellae of the hippocampus. Generally speaking, the models of situations in the hippocampal lamellae also change. That is, this iterative process reorders the cortical information about the events of the current situation in accordance with the existing situation models stored in the hippocampus, and these situation models take into account information about the current situation mapped as images of events onto the cortex.

And since the CA3 field of the hippocampus works as an enormous auto-associative recurrent memory across its length and width [43], many individual models of situations Np that are stored in lamellae p, together with the Images of events stored in cortex columns, form a single semantic network N on a multimodal model of the world stored in the cortex columns (eq.2).

$$N = \cup p \; Np. \qquad (2)$$

Here, it is not the detailed representation of the stored images that is important – this is provided by the cortex – but contextual space-time connections of the images within entire situations.

3. Analysis of written text by human

The language is a hierarchy of levels, represented by level-forming elements from graphemes (the level of the alphabet elements) to pairs of root stems (the level of representation of their admissible compatibility, that is, semantics). For simplicity (ignoring the form details – morphology and syntax), this hierarchy can be divided into two levels: at the first level, combinations of alphabetic characters in words (root stems of words) are presented, and at the second level, combinations of pairs of root stems. In the course of processing a specific text (corpus of texts), these dictionaries will be filled: the first one with the root stems of words, and the second one with pairs of root stems. The latter will be necessary to form a homogeneous semantic network. To form dictionaries of level-forming units of the language of these two levels (root stems and pairs of root stems), one needs neural networks based on neuron-like elements with temporal summation of signals (unlike neurons with spatial summation traditionally used in various neural network paradigms, including convolutional networks) to reflect the relationships of lower-level elements in upper-level elements.

3.1. Structural analysis of digital text

In the minimum configuration, the structural analysis of digitized text is reduced to the formation of dictionaries of only two levels instead of dictionaries of six levels: a dictionary (1) of the root stems of words $\{Bi\ \}$4.and a dictionary (2) of pairs of root stems of words $\{Bi\ \}$6.

We take the number of layers of an artificial neural network based on neural-like elements with temporal summation of signals used to analyze texts at the word level does not exceed the length of the longest word [41, 42], for example, equal to 20.

The relationship between words (the root stems of words) of a text is reduced in the simplest case to associativity relations between them. That is, the sentences of the text, and therefore the entire text, are reduced to listing pairs of adjacent words (first-second, second-third, etc.).

The dictionary of the level of root stems is formed relatively simply: by counting the frequency of occurrence of root stems in the text. Therefore, let us consider only the formation of a dictionary of the semantic level in more detail.

3.1.1. Dictionary of the semantic level

The dictionary of the semantic level $\{Bi\ \}$6.is formed as a dictionary of pairwise compatibility of the root stems of words: as a set of asymmetric pairs of events $\{< cicj >\}$, where ci and cj (root stems) are events connected by an associativity relation (joint occurrence in a text sentence). T

hus, the text is reduced to a list of root stems of words and pairs of root stems of words that (virtually) correspond to a homogeneous semantic network. That is, at the upper levels of text analysis, one can proceed to the manipulation of semantic networks.

3.1.2. Semantic network of text

The dictionary of the highest level is the dictionary of pairwise compatibility of root stems in the text (dictionary of restrictions on compatibility), the dictionary of the semantic level that is a virtual homogeneous (associative) directed semantic network. Indeed, if one collects chains from pairs of words (more precisely, the root stems of words), chains with loops and branches will be obtained. However, such a frequency network (where the frequency of occurrence of individual words in the text is known, as well as the pairwise occurrence of words in sentences of the text) is only the initial basis for obtaining the actual semantic network. To recalculate the frequency of occurrence into a semantic weight, the network is re-ranked using an iterative procedure similar to the Hopfield network algorithm [45].

A directed homogeneous semantic network N is a graph, whose nodes correspond to the root stems of the words of the dictionary $\{Bi\ \}$4.of the root stems of the words in the analyzed text (corpus of texts, the language as a whole), and the arcs correspond to the associative relation, that is, pairwise compatibility of the root stems of words in sentences of the text.

Definition 1. A semantic network N is understood to be a set of directed pairs of events $\{< cicj >\}$, where ci and cj are events interconnected by the associativity relation (co-occurrence in a certain situation, eq.3):

$$N \approx \{< cicj >\}. \tag{3}$$

In this case, the association relation is asymmetrical: $< cicj >\neq< cjci >$.

Definition 2. The weight zi of the event image ci in the network is a value of the counter of occurrences of events in the input text.

3.1.2. Re-ranking of the associative network notions

The presentation of the composition of the root stems of words in the texts of the language and their cohesion is a qualitative presentation of the text content. The analysis of the text content should enable a quantitative assessment of both the ranks of the nodes of this network and the characteristics of the links. Thus, when forming a network based on a large corpus of texts, correct weight characteristics of nodes-notions are obtained: the frequency of their occurrence approaches their semantic weight.

3.1.3. Re-ranking of the associative network notions

The presentation of the composition of the root stems of words in the texts of the language and their cohesion is a qualitative presentation of the text content. The analysis of the text content should enable a quantitative assessment of both the ranks of the nodes of this network and the characteristics of the links.

Thus, when forming a network based on a large corpus of texts, correct weight characteristics of nodes-notions are obtained: the frequency of their occurrence approaches their semantic weight. When analyzing small texts, the frequency of occurrence no longer characterizes the importance of the notion. In this case, to identify the ranks of nodes, the weight characteristics of the notions of the associative network are re-ranked using an iterative procedure similar to the Hopfield network algorithm [45], which enables the transition from the frequency portrait of the text to the associative network of key notions of the text (eq.4):

$$wi\ (t+1) = (\sum wi\ (t)\ i,i\neq j\ wij)\sigma(\overline{E}), \tag{4}$$

here wi (0) = zi ; wij = zij/zi ; and $\sigma(\bar{E})$ = 1/(1 + $e^{-k\bar{E}}$) is a function that normalizes to the average value of the energy of all nodes of the network $\bar{E}$, where zi is the frequency of occurrence of the i-th word in the text, and zij is the frequency of joint occurrence of the i-th and j-th words in text fragments.

As a result of such re-ranking, the initial weights of words change. Words that are connected in the network with a large number of words with large weights, including those through intermediate words, increase their weights as a result of this procedure, while weights of other words evenly decrease. The obtained numerical characteristic of words – their semantic weight – characterizes the degree of their importance in the text.

For the analysis of text sequences, artificial neural networks based on non-binary neurons with temporal summation of signals will be used. Non-binary neurons differ from the binary neurons described above with temporal summation of signals (which are convenient to use to explain the mechanism of structural processing in hand-waving terms, but cannot be used to analyze real texts) by the presence of a generalized dendrite, which receives a code sequence consisting of vectors as input.

3.2. Non-binary neuron with temporal summation of signals

Fig. 2 shows a non-binary neuron-like element with temporal summation of signals, which has several (k) generalized dendrites (shift registers of length n), including an adder and a threshold converter. In addition to dendrites, an adder, and a threshold device, the neuron includes a memory area for recording heteroassociative information, and counters that count the number of joint occurrences of the neuron address in the input code sequence and information accompanying the address. As an input code sequence, a sequence of k-valued feature vectors is used, the elements of which are non-negative real numbers. Each of them enters its own shift register.

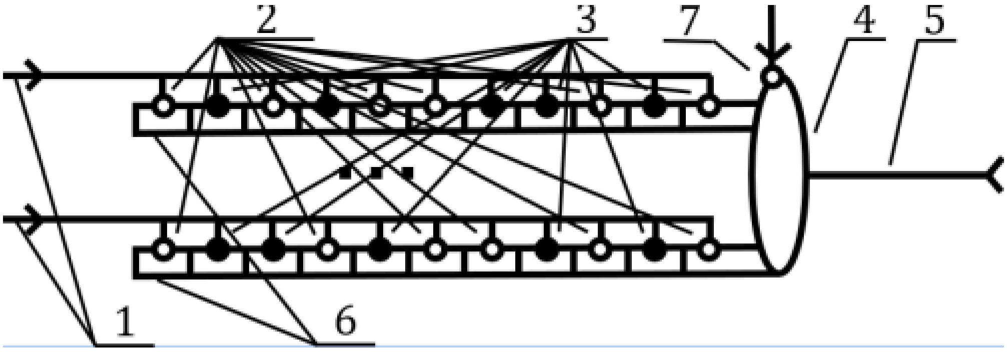

Figure 2 – A non-binary neuron-like element with temporal summation of signals.

In this case, the pyramidal neurons of the third layer of the cortical column are modeled by the so-called dynamic associative memory [41, 42] (hereinafter referred to as DAM), which is a set (see Fig. 3) of neuron-like elements connected in parallel, storing in the corresponding counters the joint occurrence of fragments of length n of the input code sequence consisting of k-bit vectors and related information.

The set of neuron-like elements that make up the DAM is formed according to the input information: the occurrence of meaningful information at the DAM input (n successive k-valued feature vectors) leads to the occurrence in the DAM of a neuron-like element with an address corresponding to the nxk-element matrix, which includes n consecutive k-valued feature vectors. In the DAM, as many neuron-like elements are formed as needed to display the entire input code sequence

Neural-like elements with temporal summation of input signals, which are part of the DAM, model with their addresses the nodes of the (nxk)-dimensional signal space. Consider the formalism of information processing in such a non-binary DAM.

Let we have an (nxk)-dimensional signal space $R^{n\mathrm{x}k}$. For further presentation, we introduce some notations and definitions.

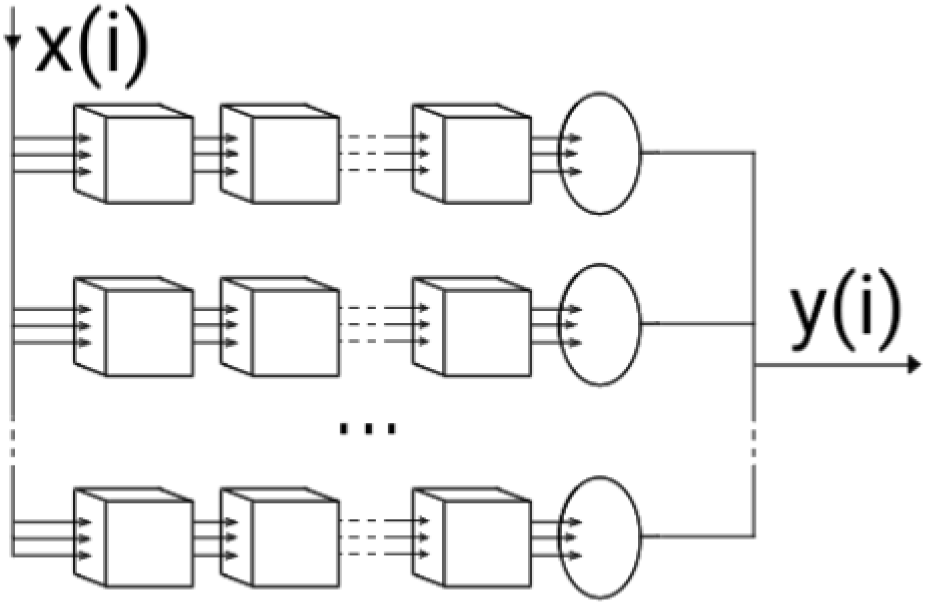

Figure 3 – Dynamic associative memory as a set of initiated neurons.

Denote by $\{\boldsymbol{A}\}$ a set of code sequences formed by the signal periphery of some (for example, speech) analyzer, whose elements are feature vectors that make up the input sequences $\boldsymbol{A} = (\dots\ \boldsymbol{a}_{-1}, \boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_i, \dots)$, where $\boldsymbol{a}_i$ is a k-valued feature vector, whose components are non-negative real numbers (for example, cepstral coefficients in case of a speech analyzer).

Denote by $\{\hat{A}\}$ a set of trajectories of sequences corresponding to the set of input sequences $\{\boldsymbol{A}\}$, whose elements $\hat{a}_i$ are points in the space $R^{n\mathrm{x}k}$, i.e. $\hat{a}_i \in R^{n\mathrm{x}k}$, where $\hat{a}_i = (\boldsymbol{a}_{i-n\mathrm{x}k+1}, \boldsymbol{a}_{i-n\mathrm{x}k+2}, \dots, \boldsymbol{a}_i)$ are successive fragments of the sequence A of length n of k-valued vectors shifted relative to each other by one vector (by one period of time) – coordinates of points of the multidimensional space $R^{n\mathrm{x}k}$.

Definition 3. A **trajectory** is a sequence of points $\hat{a}_i$ of the multidimensional space $R^{n\mathrm{x}k}$, corresponding to the input code sequence **A**.

We introduce a transformation $F^{n\mathrm{x}k}$ (eq. 5):

$$F^{n\mathrm{x}k}: \boldsymbol{A} \rightarrow \hat{A},\ F^{n\mathrm{x}k}(\boldsymbol{A}) = \hat{A},$$

$$\text{where } \boldsymbol{A} = (\dots, \boldsymbol{a}_i, \dots : \boldsymbol{a}_i \in R^{n\mathrm{x}k}),$$

$$\text{and } \hat{A} = (\dots, \hat{a}_{-2}, \hat{a}_{-1}, \dots\ \hat{a}_i, \dots) =$$

$$= (\dots, (\boldsymbol{a}_{-n\mathrm{x}k-1}, \boldsymbol{a}_{-n\mathrm{x}k}, \dots, \boldsymbol{a}_{-2}), (\boldsymbol{a}_{-n\mathrm{x}k}, \boldsymbol{a}_{-n\mathrm{x}k+1}, \dots, \boldsymbol{a}_{-1}), \dots, (\boldsymbol{a}_{i-n\mathrm{x}k+1}, \boldsymbol{a}_{i-n\mathrm{x}k+2}, \dots, \boldsymbol{a}_i), \dots). \tag{5}$$

In a binary case (n=3) trajectory looks like at Fig. 4.

The introduced transformation $F^{n\mathrm{x}k}$, which forms a trajectory in the $n\mathrm{x}k$-dimensional signal space, with the coordinates of the points given by n-term fragments of the original vector sequence A, is the basis for structural information processing (as in the binary case). It has the associativity property of addressing to points of the trajectory $\tilde{A}$ along an n-term fragment of the sequence A: any n vectors of the original sequence A refer us to the corresponding point of the trajectory $\tilde{A}$.

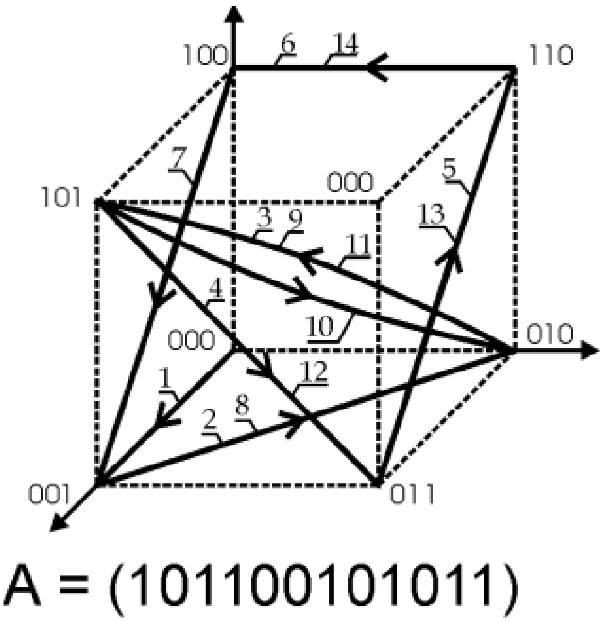

Figure 4. N-dimensional unit hypercube Ge , where n=3. The trajectory in the signal space corresponds to the sequence A.

The associativity of the transformation (5) makes it possible to preserve the structural topology of the information being transformed: identical fragments of the input sequence are transformed into the same fragment of the trajectory, and different fragments into different fragments of the trajectory. Since, in general, the input sequence A may contain repeated n-term fragments, this involves the appearance of self-intersection points of the trajectory (specifically, to the repeated passage of entire fragments of the trajectory).

Let us set a certain sequence J and a trajectory $\tilde{A} \in R\ n\text{x}k$ corresponding to the sequence A. Let us introduce the function M (14 6) that assigns an element of the sequence J to each point of the trajectory $\tilde{A}$ (eq. 6):

$$M(\tilde{a}i\ ,ji+1\ ) = [\tilde{a}i\ ]ji+1. \qquad (6)$$

The resulting trajectory $[\tilde{A}]j$ will be called a trajectory (as in the binary case), conditioned by the sequence J (eg. 7):

$$[\tilde{A}]j = M(Fn\text{x}k\ (\boldsymbol{A}),J). \qquad (7)$$

Definition 4. Thus, the function M records the sequence J at points of the trajectory $\tilde{A}$ (in association with the sequence $\boldsymbol{A}$). Let us call this function a **function of writing to the memory**, the sequence J an **informational** or **conditioning** sequence, the **sequence A** a **carrier sequence**, and this way of writing – a hetero-associative recording.

Definition 5. The restoration of the information sequence J along the trajectory $[\tilde{A}]J$ conditioned by it and the carrier sequence A is implemented with the following function (eq. 8):

$$R\ [\tilde{A}]J = \text{J}, \qquad (8)$$

where R is called a **memory read function**. In this case, an associative mapping onto the multidimensional space of the carrier sequence A leads to the passage of points of the corresponding trajectory $\hat{A}$, which makes it possible to read the characters of the sequence J.

Definition 6. Thus, having the carrier sequence and the trajectory conditioned by the sequence J, the original information sequence can be reconstructed using the function (8). We call this method of reproduction **hetero-associative reproduction.**

Let $\boldsymbol{A}$ be a carrier sequence. If the same sequence A is used as a conditioning sequence, then we have a **self-conditioning** case. Obviously, in this case, the conditional sequence can be obtained as follows (eq. 9):

$$[\hat{A}]A = \mathrm{M}\ (Fn\mathrm{x}k\ (\boldsymbol{A}),\boldsymbol{A}),$$
$$\text{where } \hat{A} = Fn\mathrm{x}k\ (\boldsymbol{A}). \tag{9}$$

Definition 7. In case of self-conditioning, the information sequence can be reconstructed using the function (eq. 10):

$$R\ ([\hat{A}]\ \boldsymbol{A}\ ) = \boldsymbol{A}. \tag{10}$$

Such a recording is called **auto-associative recording,** and such a reproduction is called **auto-associative reproduction**. Thus, the use of functions M and R together with the $Fn\mathrm{x}k$ transformation, which has the feature of associative addressing to information, makes it possible to implement associative memory with the possibility of auto- and hetero-associative recording/reproduction of information in a non-binary case as well.

3.2.1. Accounting for the frequency of occurrence of trajectories

In contrast to the binary case, where the transition frequency at the branch point is taken into account using two counters, in the non-binary case this mechanism cannot be implemented: the number of combinations of possible transitions is infinite, since the trajectory is formed in the entire volume of the signal space. In natural neural networks, this mechanism is implemented at the system level, taking into account the interaction of individual neurons in cognitive networks, which is ensured by the thalamus. This mechanism is beyond the scope of this work, so the problem is solved by simply taking into account the combination of the address fragment and the address of the target point (the address of the next neuron) in each individual case.

In other words, a separate neuron is allocated for each current point of the trajectory and a specific current transition to the next address, so one single counter of this neuron remembers exactly the number of repetitions of this combination. The dilemma of having several neurons with the same address part is resolved by simply comparing the counters of these neurons.

Returning to natural neural networks, it must be said that in real situations of processing specific information, the thalamus selects a subnetwork that relates only to this particular sensory input sequence, and therefore the number of neurons required to store this particular information is relatively small.

Thus, just as in the binary case, the memory mechanism (11) is a counter fixing the number of passages of a given trajectory point in a given direction $C\hat{a}\hat{i}$. The use of these counters makes it possible to determine the value of the most likely transition for a given point. As well as the number of neurons required for memorization, the number of counters is determined by the needs of a specific memorized array of information.

Suppose that a carrier sequence A is defined, as well as a trajectory $\hat{A}$ generated by this sequence. Then the counters $C\hat{a}\hat{i}$ for the i-th point of the trajectory $\hat{A}$ for the t-th moment of time are calculated as follows:

$$M(\hat{a}\hat{i}\ , \boldsymbol{a}i{+}1\ ) = [\hat{a}\hat{i}{+}1\ ] = C\hat{a}\hat{i}\ (t) = C\hat{a}\hat{i}\ (t-1) + 1|\boldsymbol{a}i{+}1 \in V\ s\ , \tag{11}$$

where $V\ s$ is a set of transition vectors $\boldsymbol{a}i{+}1 \in R\ n\mathrm{x}k$ for the neuron with a given address. During reproduction, the states of the counters are analysed, and the current character is formed depending on whether the following condition is met (eq. 12):

$$\boldsymbol{a}i{+}1 = R([\hat{a}\hat{i}\ ]) = R\ (C\hat{a}\hat{i}\ (t))\ |\ \boldsymbol{a}i{+}1 \in V\ k. \tag{12}$$

Such a memory mechanism is sensitive to the number of passages of a given point in a given direction and makes it possible to characterize each point of the trajectory regarding the frequency of occurrence in the input information sequence of any repeating fragment. It is this mechanism that enables formation of dictionaries of repeating fragments in the input information, being the basic mechanism for structural processing.

Let us introduce a threshold transformation H with a threshold h. Then the superposition of the functions $HhRMFn\mathrm{x}k(\boldsymbol{A})$ will enable selection of only those points of the trajectory in the signal space that were passed at least h times.

The use of a training threshold transformation (according to the number of passages of the trajectory) makes it possible to form dictionaries of repeating fragments (level-forming elements of language levels in our case of speech analysis in input code sequences).

3.3. Formation of a semantic network

An artificial neural network in this system is represented by a multilayer structure of a set of parallel-connected neurons of dimension n=2, where, in the first layer, the neurons form an address during training, consisting of the characters of the first two letters of the root stem of each word and they remember their index, and in each subsequent layer the address is formed from the index of the previous combination and the subsequent letter of the root stem of this word.

3.3.1. Formation of a dictionary of root stems

Such an artificial neural network has several layers (for example, 20). In the first layer, a twoterm combination of alphabetic characters is stored, and in subsequent layers, the neuron index of the first layer and the next alphabetic character from the analyzed word are stored. In addition to this information, a particular neuron remembers the frequency of occurrence of this combination in the text.

The resulting state of a particular neuron is reached after the change in its state stops: this means that this neuron is the final one in the combination of neurons containing information on a specific root stem. Its memory stores the frequency of occurrence of this root stem in the text.

Similarly, it is possible to track the presence of established collocations in the text. However, in this case, the final state of the last neuron in the chain is reached with the last combination, including the last character of the alphabet included in this collocation.

3.3.2. Formation of a dictionary of pairs of root stems

For the second level – semantic – one can also build a neural network, but it’s easier to just memorize pairs of indices of root stems, so that later one can build a homogeneous (associative) semantic network from them, that can be manipulated in various ways, comparing texts by meaning, classifying texts, clustering texts into groups, forming an abstract, or a topical abstract of the text.

3.3.3. Formation of a homogeneous semantic network

Pairs of root stems from the dictionary of pairwise compatibility (semantic level) virtually constitute a homogeneous (associative) directed semantic network with branches and loops.

Frequency network. Both nodes and arcs of the primary network are normalized: the former – by the frequency of occurrence of root stems in the text, the latter – by the frequency of pairwise occurrence of root stems in sentences of the text.

To calculate the ranks of the network nodes corresponding to the ranks of notions in the text, it is necessary to perform an iterative reweighing procedure, as a result of which the ranks of the corresponding nodes become dependent on their connections with other network nodes. The reweighing depth (number of iterations) is determined either as a result of the convergence of the iterative process, or is set voluntarily (for example, is set equal to 10 – let’s choose the average number of words in a text sentence as such).

Re-ranking of network nodes. Re-ranking the frequency of occurrence into semantic weight enables some intelligent procedures on texts and text corpora: extract keywords, abstract, compare by meaning, classify, clusterize. n-gram model of the text. Since there is no valid a priori knowledge about the equality of word allocations in various positions of the line, a contextual reference through conditional probabilities is introduced [46]. Therefore, we turn to an n-gram, and more specifically, to a “one-sided” n-gram model, that is, a “right-hand” model adopted for using n-grams, in which the probability of the next word in the line is determined depending on preceding (n − 1) words, which may be written as p(wn|w1 … wn−1).

Topic tree. After building a semantic network from a set of word pairs $\{< wiwj >\}$ (in fact, from asterisks $< wi < wj >>$), after rearranging the nodes of the semantic network (after iterative recalculation of their weights), one can construct a topic tree either for the entire text, or only for some notion presented in the text, to which end the minimal tree subgraph T is extracted from the network.

To extract the minimal tree subgraph from the semantic network, we choose a pair of words (wi , wj), where the main word has the highest weight among all pairs. Then this pair is attached to all other pairs, where the main word is the same as that of the first pair $< wi < wj >>$. To the resulting asterisk asterisks are then added, in which the main words match the secondary words of the first asterisk. Here, two conditions are met: (1) if the secondary word of any pair of the attached asterisk matches the main word of one of the asterisks in the part of the topic tree already formed, the process stops at this point, and this pair of this asterisk is discarded; (2) the weights of the secondary words of the attached pairs are analysed, and if the weight of any secondary word of any of the attached asterisks is less than a predetermined threshold value h, this pair is discarded, and the process in this branch is terminated.

Definition 8. A **topic tree** T is a set of word pairs from the semantic network N obtained using the procedure described above and satisfying conditions (1) and (2). If there is more than one root node, the number of topic trees built matches the number of root nodes.

4. TextAnalyst, a program for automatic semantic analysis of texts

Using such an artificial neural network based on neurons with temporal summation of signals, as well as a reweighing procedure, a software system for automatic semantic analysis of texts “TextAnalyst” was implemented [47]. This technology enabled automatic formation of a description of the semantics (structure) of the subject domain of the text, as well as the functions of organizing the text base into a hypertext structure, automatic summarizing, comparing and classifying texts, as well as the semantic search function.

4.1. Technology software implementation

The system is implemented as a tool for automatic generation of knowledge bases using a set of natural language texts. The system kernel [48] is implemented as a software component (inproc server) that complies with the Microsoft Component Object Model (COM) specification.

The system kernel implements the following functions: normalization of grammatical forms of words; automatic identification of basic notions of the text (words and phrases) and their relationships with the calculation of their relative significance; formation of the representation of the text (text set) semantics in the form of a semantic network.

In addition to the initial processing (preprocessing) unit, the system kernel includes the following units (see Fig. 5): linguistic processor; unit for identification of text notions; unit for semantic network formation; unit for semantic network storage.

4.1.1. Preprocessing unit

This unit is designed to extract text from a file (input data stream) and prepare it for processing in the linguistic processor. Preparation of the text consists of removing characters unknown to the linguistic processor, as well as the correct processing of such text units as abbreviations, initials, titles, addresses, numbers, dates and time pointers.

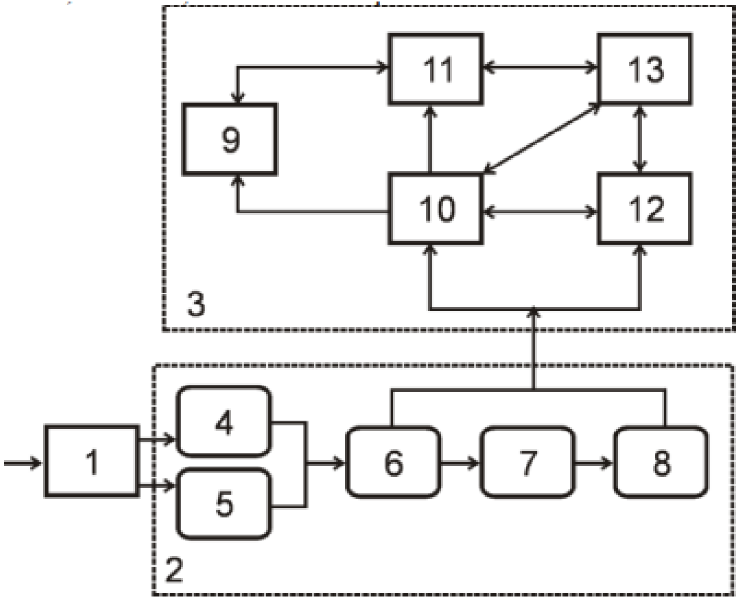

Figure 5 - The text analysis system contains a preprocessing unit (1), linguistic (2) and semantic (3) processors. The linguistic processor includes dictionaries of: (4) delimiting words, (5) empty words, (6) commonly used words, and (7) inflectional and (8) root morphemes. The semantical processor, in turn, contains: (9) a unit of references to the text, (10) a unit for forming a semantic network, (11) a unit for storing this semantic network, (12) a unit for identification of notions, and (13) a control unit

4.1.2. Linguistic processor

The linguistic processor preprocesses the input text (sequences of characters) based on a priori linguistic knowledge common for the selected language (several European languages other than Russian and English are currently supported) and performs the following functions: segmentation of the text sentences based on punctuation marks and special grammatical words; normalization of words and collocations – filtration of inflections (endings) preserving only the root stems; filtration of semantically insignificant, auxiliary words in the text (prepositions, numerals and most commonly used words with a wide meaning are removed); and finally, marking of commonly used words.

Segmentation of sentences makes it possible to break the text into fragments (sentences), which may contain terminological phrases of the subject domain, and to avoid identifying inappropriate phrases at the junctions of such fragments.

As a result of preprocessing, semantically close words and phrases are reduced to the same form (normalized). It is necessary to mark common words in order to exclude their identification as independent terms in further analysis.

The base of general knowledge in the linguistic processor contains dictionaries, one for the implementation of each of the four functions: a dictionary of sentence-delimiting words, a dictionary of auxiliary words, a dictionary of inflections and a dictionary of commonly used words.

4.2. Main functions of the TextAnalyst system

Based on the semantic network obtained as a result of text (corpus of texts) processing, the following functions of text information processing are implemented: 1) function of hypertext (knowledge base) structure formation, 2)) navigation within the knowledge base, 3) formation of a topic tree, 4) text summarization, 5) automatic clustering of multiple texts, 6) text comparison (automatic text classification); and, finally, 7) function of forming an answer to the user’s query, that is, formation of a topic summary.

After the semantic network is formed, the original text combined with hyperlinks to the semantic network, becomes a hypertext structure. In this case, the semantic network becomes a convenient means of navigating through the text. It makes it possible to explore the basic structure of the text, moving from notion to notion through associative links. Using hyperlinks, the user can move from any sentence directly to its context in the text. With the same purpose, the user can use the minimal tree subgraph of the semantic network, a topic tree. This contains hierarchically represented basic and subordinated network notions, where lower-level notions explain the content of higher-level notions. The topic tree can also be used to navigate through the knowledge base, as well as the semantic network, as it resembles the table of contents of the text.

The semantic network with the numerical values of its components (notions and their links) enables calculation of the weight of each sentence in the text. The set of sentences of the text selected in the order of their appearance in the text, the weight of which exceeded a certain threshold level, can be considered a summary of the text. The semantic network of the studied text (or group of texts) can be broken down into subnetworks by removing weak links from it. Each such subnetwork is grouped around a certain notion with the maximum weight in this subnetwork. This notion refers to the topic of a part of the text or individual texts that are grouped in this subnetwork. This automatic clustering makes it possible to split a set of texts into headings.

Using the numerical values of the semantic network, one can compare the networks of two texts in terms of calculating their intersection (common part). That is, one can compare the degree of coincidence of texts in meaning. If a whole heading is taken as one of the texts, then it is possible to estimate the degree to which the original text belongs to this heading, that is, to automatically classify the texts.

The system for the semantic analysis of texts also implements a semantic search (forms a topic summary). The semantic search function, based on an associative hierarchical representation of the information content in the database and on clustering and classification functions, selects information corresponding to the user's query, and structures it in accordance with the similarity to the query.

This semantic search using associations provides the user with information that is not explicitly specified in the query text but is related to it semantically (in meaning). Using this approach does not lead to an increase in the amount of information provided to the user, but rather to its careful selection and analysis by the main criterion – semantic similarity to the query.

5. Applications

The TextAnalyst technology [47, 48] has been used in a number of applications to solve practical problems of analyzing texts and quasi-texts (meaningful sequences of images of various modalities), including: assessing the significance of specific notions in a text (corpus of texts) – for example, ranking individual parameters when assessing human capital assets [42]; assessments of the significance of texts (corpora of texts) within a whole subject domain, for example, assessments of the productivity of individual specialists and entire teams [42]; extraction of implicit information from author's texts [42]; automatic creation of electronic books with associative navigation [49]; analysis of quasi-textual information, for example, the analysis of genetic links [42].

6. Next steps. Heterogeneous semantic network

The mechanisms described above can be extended by representing the text by a heterogeneous semantic network instead of a homogeneous semantic network. At the moment, the published literature does not provide mechanisms for the automatic formation of heterogeneous semantic networks [50]. Nevertheless, there are tools (depending on the analyzed language) [51], which allow revealing the extended predicate structure of individual sentences of the text (up to 85% of the text volume). The TextAnalyst technology and the tools discussed above make it possible to approach the creation of applications for the formation of heterogeneous semantic networks.

The algorithm for automatic formation of a heterogeneous semantic network in this case looks something like this. Based on a given text (corpus of texts), a homogeneous semantic network is built. Then, for each pair of notions of the constructed homogeneous semantic network, the links between these notions in the sentences of the text are revealed: as many as many sentences contain a particular pair of notions. If the analysis of several sentences containing a particular pair of notions results in the identification of one type of links, this is taken into account in the formation of the weight of this pair (the weight of this type of relationship in the heterogeneous semantic network). This is how associative links are replaced by other types of links for all pairs of notions of the homogeneous semantic network. Since existing applications do not allow restoring the extended predicate structure for all sentences of the text, some links remain unchanged (associative).

The formation of a heterogeneous semantic network is a necessary condition for solving a number of problems in text analysis. Therefore, the automation of this process is a progress in this direction. It should be noted that in the process of replacing a homogeneous semantic network with its heterogeneous version, the robust characteristics of the approach deteriorate: the network is stratified, and the power of links decreases, that is, the interpretive properties of the network representation are degraded.

7. Conclusion

The paper considered the issue of using the deep learning approach to solving problems of automatic analysis of textual information. The presented approach is based on understanding the processes of information processing in the human mind, including structural information processing in the columns of the cerebral cortex, which are modeled by artificial neural networks based on neural-like elements with temporal summation of signals (on the example of language information processing), as well as re-ranking the weight characteristics of the notions in the semantic network in the hippo-campal lamellae. The result of structural processing is a hierarchy of dictionaries of event images of various modalities, the top-level dictionary of which (the dictionary of the semantic level, that is, the dictionary of the admissible pairwise compatibility of event images) is used to build a homogeneous semantic network, where the weights of the nodes (conceptual notions) are re-ranked using an algorithm similar to the algorithm of an artificial Hopfield neural network. The paper presents in detail the architecture of a neural network based on neuron-like elements with temporal summation of signals, configured to process specific (textual) information. The architecture of a software system is presented that is designed for processing textual information, including a subsystem for identifying key concepts of a text, as well as a subsystem for forming a semantic network of text based on pairs of key notions identified in text sentences. Algorithms for the implementation of a system for automatic semantic analysis of textual information are proposed on the example of their implementation by the Moscow company MICROSYSTEMS, using the TextAnalyst technology. This technology implements the following functions: formation of a semantic network, comparison of texts by structure (by meaning), classification of texts and automatic summarization of texts. Examples of the use of this technology in a number of subject domains are presented, including those for: 1) information and analytical expert evaluation of authors’ texts; 2) ranking individual characteristics of some entity and their combinations presented in texts (for example, parameters of human capital assets); 3) revealing implicit information in text perception (on the example of the analysis of texts by V. Nabokov and J. Brodsky); 4) analysis of quasi-texts (for example, classification of the results of the analysis of genetic quasi-texts, that is, signalling networks); 5) creation of e-books. Finally, considerations are presented on the possibility of implementing automatic construction of heterogeneous semantic networks.

Conflicts of Interest: The authors declare no conflict of interest

References

1. Lena Voita. Language Modeling. // Available online: https://lena-voita.github.io/nlp_course/language_modeling (accessed on 20 June 2023).
2. Zhao, Wayne Xin, et al. "A survey of large language models." arXiv preprint arXiv:2303.18223 (2023).
3. Y. Zhu, R. Kiros, R. S. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, “Aligning books and movies: Towards story-like visual explanations by watching movies and reading books,” in 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015. IEEE Computer Society, 2015, pp. 19–27
4. “Project Gutenberg.” [Online]. Available: https://www.gutenberg.org/
5. “Wikipedia.” [Online]. Available: https://en.wikipedia.org/wiki/Main Page
6. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al., “Language models are unsupervised multitask learners,” OpenAI blog, p. 9, 2019
7. A. Gokaslan, V. C. E. Pavlick, and S. Tellex, “Openwebtext corpus,” http://Skylion007.github.io/OpenWebTextCorpus, 2019.
8. J. Baumgartner, S. Zannettou, B. Keegan, M. Squire, and J. Blackburn, “The pushshift reddit dataset,” in Proceedings of the Fourteenth International AAAI Conference on Web and Social Media, ICWSM 2020, Held Virtually, Original Venue: Atlanta, Georgia, USA, June 8-11, 2020. AAAI Press, 2020, pp. 830–839.
9. “Common crawl.” [Online]. Available: https://commoncrawl.org/
10. L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mt5: A massively multilingual pre-trained text-to-text transformer,” in Proceedings of the 2021 Conference of the

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, 2021, pp.483–498.

11. R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, “Defending against neural fake news,” in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alch´e Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 9051–9062.
12. The Russian National Corpus (ruscorpora.ru). 2003—2023.
13. “Bigquery dataset.” [Online]. Available: https://cloud.google.com/bigquery?hl=zh-cn
14 Gaulton, A. et al. The ChEMBL database in 2017. Nucleic Acids Res. 45, D945–D954 (2017)
15. Merk, D., Friedrich, L., Grisoni, F. & Schneider, G. De novo design of bioactive small molecules by artificial intelligence. Mol. Inf. 37, 1700153 (2018).
16. Chip Huyen. Evaluation Metrics for Language Modeling \\ The Gradient. 2019 [Online]. Available: https://thegradient.pub/understanding-evaluation-metrics-for-language-models/
17. Gu, Jiuxiang, et al. "Recent advances in convolutional neural networks." Pattern recognition 77 (2018): 354-377.
18. Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." arXiv preprint arXiv:1603.07285 (2016).
19. Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." arXiv preprint arXiv:1609.03499 (2016).
20. Dauphin, Yann N., et al. "Language modeling with gated convolutional networks." International conference on machine learning. PMLR, 2017.
21. Staudemeyer, Ralf C., and Eric Rothstein Morris. "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks." arXiv preprint arXiv:1909.09586 (2019).
22. Sepp; Hochreiter and J¨urgen Schmidhuber. Long Short-Term Memory. Neural computation, 9(8):1735–1780, 1997.
23. Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In arXiv, pages 1–9, dec 2014.
24. Sharma, Sagar, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks." Towards Data Sci 6.12 (2017): 310-316.
25. Jozefowicz, R., Zaremba, W., & Sutskever, I. (2015, June). An empirical exploration of recurrent network architectures. In International conference on machine learning (pp. 2342-2350). PMLR.
26. Alex Graves and Jurgen Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In Proc. of the Int. Joint Conf. on Neural Networks, volume 18, pages 2047–2052, Oxford, UK, UK, jun 2005. Elsevier Science Ltd.
27. Rae, Jack, et al. "Fast parametric learning with activation memorization." International Conference on Machine Learning. PMLR, 2018.
28. Grave, E., Joulin, A., and Usunier, N. Improving neural language models with a continuous
cache. arXiv preprint arXiv:1612.04426, 2016b.
29. Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018).
30. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
31. Lin, Tianyang, et al. "A survey of transformers." AI Open (2022).
32. Kalyan, Katikapalli Subramanyam, Ajit Rajasekharan, and Sivanesan Sangeetha. "Ammus: A survey of transformer-based pretrained models in natural language processing." arXiv preprint
arXiv:2108.05542 (2021).
33. Baevski, Alexei, and Michael Auli. "Adaptive input representations for neural language modeling." arXiv preprint arXiv:1809.10853 (2018).
34. Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv:1810.04805v2
35. Luo, Linkai, and Yue Wang. "Emotionx-hsu: Adopting pre-trained bert for emotion classification." arXiv preprint arXiv:1907.09669 (2019).
36. Radford, Alec, et al. "Language models are unsupervised multitask learners." OpenAI blog 1.8 (2019): 9.
37. Shoeybi, Mohammad, et al. "Megatron-lm: Training multi-billion parameter language models using model parallelism." arXiv preprint arXiv:1909.08053 (2019).

38. Ren, Xiaozhe; Zhou, Pingyi; Meng, Xinfan; Huang, Xinjing; Wang, Yadao; Wang, Weichao; Li, Pengfei; Zhang, Xiaoda; Podolskiy, Alexander; Arshinov, Grigory; Bout, Andrey; Piontkovskaya, Irina; Wei, Jiansheng; Jiang, Xin; Su, Teng; Liu, Qun; Yao, Jun (March 19, 2023). "PanGu-Σ: Towards Trillion Parameter Language Model with Sparse Heterogeneous Computing". arXiv:2303.10845
39. Dai, Andrew M; Du, Nan (December 9, 2021). "More Efficient In-Context Learning with GLaM". ai.googleblog.com. Retrieved 2023-03-09.
40. Dey, Nolan, et al. "Cerebras-GPT: Open compute-optimal language models trained on the Cerebras wafer-scale cluster." arXiv preprint arXiv:2304.03208 (2023).
41. Kharlamov A. A. Assotsiativnaya pamyat' – sreda dlya formirovaniya prostranstva znanij. Ot biologii k prilozheniyam. [Associative memory as an environment for the formation of a space of knowledge. From biology to applications]. Dusseldorf, Germany: Palmarium Academic Publishing - 109 p. (in Russian) Available online: Ассоциативная память — среда для формирования пространства знаний: От биологии к приложениям (Russian Edition): Харламов, Александр: 9783639645491: Amazon.com: Books (accessed on 12 April 2017).
42. Neuroinformatics and Semantic Representations. Theory and Applications. Alexander Kharlamov & Maria Pilgun eds. 317 P. Cambridge Scholars Publishing. 2020. Available online: Neuroinformatics and Semantic Representations: Theory and Applications - Cambridge Scholars Publishing (accessed on 2020).
43. Rolls, E.T. Theoretical and Neurophysiological Analysis of the Functions of the Primate Hippocampus in Memory. In: Cold Spring Harbor Symposia on Quantitative Biology, Vol. LV, 1990, Cold Spring Harbor Laboratory Press. Pp. 995 – 1006. Available online: Dendritic organization in the neurons of the visual and motor cortices of the cat - PMC (nih.gov) (accessed on 1953).
44. Vinogradova O.S. Gippokamp i pamyat'. [Hippocampus and memory] Moscow: Nauka, 1975. - 336 p. (in Russian) Available online: Vinogradova, Okga Sergeevna – Gippokamp i pamyat' [Текст] - Search RSL (accessed on 1975).
45. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. 79, 1982. Pp. 2554 – 2558. Available online: Neural networks and physical systems with emergent collective computational abilities. - PMC (nih.gov) (accessed on 1982).
46. Buzikashvili N.E., Samoylov D.V., Krylova G.A. N-grammy v lingvistike. [N-grams in linguistics]// In Collection of papers: Methods and means of document management. Moscow: Editorial URRS. 2000. Pp. 91-130. (in Russian) Available online: Metody I sredstva raboty s dokumentami | | Editorial URSS | Knigi po reklame, marketingu, PR I dizayinu | Advertology.Ru (accessed on 2000).
47. Kharlamov A.A. Svidetelstvo o registratsii programmy "Programma dlya avtomaticheskogo smyslovogo analiza tekstov na osnove neyronnykh setey "TextAnalyst"" [Certificate of registration of the program "Program for automatic semantic text processing based on neural networks "TextAnalyst""]. Available online: ww.fips.ru/vse-servisy.php (accessed on 31 October 1997).
48. Kharlamov A.A. Sposob avtomatizirovannoj semanticheskoj indeksatsii teksta na estestvennom yazyke. [A method for automated semantic indexing of natural language text] Patent for invention No. 2518946, priority dated November 27, 2012. Registered April 11, 2014 (in Russian) Available online: A METHOD OF AUTOMATED SEMANTIC INDEXING OF TEXT IN NATURAL LANGUAGE. Patent No. RU 2518946 IPC G06F40/20 | Patent Exchange - Moscow Innovation Cluster (i.moscow) (accessed on 2014).
49. R-sistema. Vvedenie v ekonomicheskij shpionazh. Praktikum po ekonomicheskoj razvedke v sovremennom rossijskom predprinimatel'stve. [I-system. Introduction to economic espionage. Workshop on economic intelligence in modern Russian business] In 2 volumes. Moscow, Russia: "Hamtek Publisher", 1997. (in Russian) Available online: Sergey Khich / R-system: introduction to economic espionage. Practicum on economic intelligence in modern Russian entrepreneurship. In 2 volumes. | Arbatkniga (arbatkniga.ru) (accessed on 1997).
50. Golenkov V.V., Gulyakina N.A. Printsipy postroeniya massovoj semanticheskoj tekhnologii komponentnogo proektirovaniya intellektualnykh sistem. [Principles of building a mass semantic technology of component design of intelligent systems] Proc. of the Conference "Open Semantic Technologies for Intelligent Systems" (OSTIS 2012). 2012. Pp. 23-24. (in Russian) Available online Golenkov_Printsipy.PDF (bsuir.by) (accessed on 2012).
51. Ivan Smirnov, Maksim Stankevich, Yulia Kuznetsova, Margarita Suvorova, Daniil Larionov, Elena Nikitina, Mikhail Savelov, and Oleg Grigoriev TITANIS: A Tool for Intelligent Text Analysis in Social Media. Springer Nature Switzerland AG 2021 S. M. Kovalev et al. (Eds.): RCAI 2021, LNAI 12948, pp. 232–247. 2021. Available online https://doi.org/10.1007/978-3-030-86855-0_16 (accessed on 2021)

RESUME

Kharlamov Alexander, Samaev Eugeny, Kuznetsov Dmitry, Pantiukhin Dmitry
Semantic Text Analysis Using Artificial Neural Networks Based on Neural-Like Elements with Temporal Signal Summation

Text as an image is analyzed in the human visual analyzer. In this case, the image is scanned along the points of the greatest informativity, which are the inflections of the contours of the equitextural areas, into which the image is roughly divided. In the case of text analysis, individual characters of the alphabet are analyzed in this way; but as reading skills are mastered, their groups (words) are perceived as whole objects, and then groups of words and phrases are perceived. Next, the text is analyzed as repetitive language elements of varying complexity. Dictionaries of level-forming elements of varying complexity are formed, the top of which is the level of acceptable com-patibility of the root stems of words (names) in sentences of the text, that is, the semantic level. In the case of digitized text, the analysis is greatly simplified due to the absence of necessity to recognize alphabetic characters. The level of semantics represented by pairs of root stems is virtually a homogeneous directed semantic network. Re-ranking the weights of the network vertices corresponding to the root stems of individual names, as occurs in the hippocampus, makes it possible to move from the frequency characteristics of the network to their semantic weights: vertices associated with many other vertices that have large weights increase their weight to the detriment of other vertices. Such networks can be used to analyze texts that represent them: one can compare them with each other, classify and use to identify the most significant parts of texts (generate abstracts of texts), etc. Based on this approach, software technology TextAnalyst was implemented for the semantic analysis of texts. The frequency of occurrence of the root stems of words and pairs of root stems in sentences of the text is analyzed with an artificial neural network based on neurons with temporal summation of signals. The weights of the network vertices are re-ranked using a Hopfield-like iterative algorithm. The resulting technology was used for informational and analytical expert evaluation of texts, ranking of human capital assets parameters, extracting implicit information from texts (using the analysis of texts by V. Nabokov and J. Brodsky as an example), and classifying the results of genetic analysis (based on the analysis of signal genetic networks).

РЕЗЮМЕ

Харламов Александр, Самаев Евгений, Кузнецов Дмитрий,
Пантюхин Дмитрий
Семантический анализ текста с использованием искусственных нейронных сетей на основе нейроподобных элементов с временным суммированием сигналов

Текст как изображение анализируется в зрительном анализаторе человека. При этом изображение сканируется по точкам наибольшей информативности, которые представляют собой перегибы контуров эквитекстуальных областей, на которые условно делится изображение. В случае анализа текста таким образом анализируются отдельные знаки алфавита, но по мере овладения навыками чтения их группы (слова) воспринимаются как целые объекты, а затем группы слов и фраз. Далее текст анализируется как повторяющиеся языковые элементы различной сложности. Формируются словари уровнеобразующих элементов различной сложности, вершиной которых является уровень допустимой сочетаемости корневых основ слов (имен) в предложениях текста, то есть семантический уровень. В случае оцифрованного текста анализ значительно упрощается в связи с отсутствием необходимости распознавания буквенных символов. Уровень семантики, представленный парами корневых стеблей, практически представляет собой однородную направленную семантическую сеть. Перераспределение весов вершин сети, соответствующих корневым основам отдельных имен, как это происходит в гиппокампе, позволяет перейти от частотных характеристик сети к их семантическим весам: вершины, связанные со многими другими вершинами, имеющими большие веса, увеличивают свой вес в ущерб другим вершинам. Такие сети могут быть использованы для анализа представляющих их текстов: их можно сравнивать между собой, классифицировать, использовать для выделения наиболее значимых частей текстов (генерировать аннотации текстов) и т.д. На основе этого подхода была реализована программная технология TextAnalyst для семантического анализа текстов. Частота встречаемости корневых основ слов и пар корневых основ в предложениях текста анализируется с помощью искусственной нейронной сети на основе нейронов с временным суммированием сигналов. Веса вершин сети перестраиваются с помощью итерационного алгоритма, подобного алгоритму Хопфилда. Полученная технология была использована для информационно-аналитической экспертизы текстов, ранжирования параметров человеческого капитала, извлечения неявной информации из текстов (на примере анализа текстов В. Набокова и Йосифа Бродского), классификации результатов генетического анализа (на основе анализа сигнальных генетических сетей).

Харламов Александр Александрович – доктор технических наук, старший научный сотрудник, Институт высшей нервной деятельности и нейрофизиологии РАН; МГЛУ, ВШЭ; Московский физико-технического институт, г. Москва; старший научный сотрудник, профессор кафедры прикладной и экспериментальной лингвистики; профессор департамента программной инженерии; профессор кафедры интеллектуальных информационных систем и технологии. *Область научных интересов:* нейроинформатика, семантические представления, автоматическая обработка текстов, интегральные роботы, физиологий сенсорных систем, *kharlamov@analyst.ru*

Самаев Евгений Сергеевич – руководитель группы, НПП "ГАРАНТ-СЕРВИС-УНИВЕРСИТЕТ", Москва, Ленинские горы, д. 1, стр. 77, *samaev@garant.ru*

Кузнецов Дмитрий – Руководитель группы, Акционерное общество «Позитив Текнолоджиз», Москва, Преображенская пл., д. 8, *dmkuznetsov@ptsecurity.com*

Пантюхин Дмитрий Валерьевич – старший преподаватель кафедры программной инженерии НИУ ВШЭ и кафедры «КБ-4 – Интеллектуальные системы защиты информации» МИРЭА. *Область интересов:* нейронная сеть, нейрокомпьютер, нейроморфные устройства, мемристор, информационная безопасность, система нейросетевого управления, компьютерное зрение, обработка естественного языка, *dim_beavis@mail.ru*

Статья поступила в редакцию 18.05.2023.