UDC 004.58 + 004.94
DOI: 10.15587/1729-4061.2025.324957

DEVISING AN APPROACH TO PREVENTING INFORMATION CHAOS IN CHAT BOTS USING GENERATIVE ARTIFICIAL INTELLIGENCE

Olha Kryazhych
Corresponding author
PhD, Senior Researcher, Associate Professor
Department of Natural Resources*
E-mail: economconsult@gmail.com

Ivan Ivanov
PhD Student
Department of Natural Resources*

Kateryna Iushchenko
PhD, Junior Researcher
Department of Information and Communication Technologies*

Oleksii Kuprin
PhD, Junior Researcher
Department of Information and Communication Technologies*

Oleksandr Vasenko
Associate Professor, Head of Department**

Viacheslav Riznyk
Doctor of Pedagogical Sciences,
Associate Professor, Professor**

Oleksandr Ryzhkov
Doctor of Technical Sciences, Associate Professor**

*Institute of Telecommunications and Global Information Space
of the National Academy of Sciences of Ukraine
Chokolivskyi blvd., 13, Kyiv, Ukraine, 03186
**Department of Digital Methods of Teaching
Hryhorii Skovoroda University in Pereiaslav
Sukhomlynskoho str., 30, Pereiaslav, Ukraine, 08401

The object of this study is the linguistic constructions of queries to chatbots with Large Language Models (LLMs). The area of research is the emergence of information chaos during communication between the user and the chatbot, which leads to errors in forming a response to the query. It is assumed that the user and the chatbot are separate complex systems, the events and actions of which are difficult to predict for a long period. Behavior models of complex systems are subject to the influence of chaos theory. To demonstrate this, the work used one of the simple mathematical problems with a logical component. The Copilot and ChatGPT 4o mini chatbots that were studied gave erroneous results in response to a query for the task. The error occurred when generating a query due to the introduction of a logical component. A similar process was represented by a system of differential equations solving which establishes clear rules for obtaining an accurate answer to the query.

To submit a request from the user, an approach has been proposed that makes it possible to break down the information block of the request by constructing piecewise linear attractors. That is, paired semantic expressions are formed with the formation of a request cleared of information noise. The problem is solved by presenting a methodology for controlling the selection of substitute words, based on the operations of generating the next substitution and calculating the number of the given substitution.

According to the devised approach, the best options for a request to the Copilot chatbot were obtained in 182 characters or 48 words, numbers, and special characters. For the ChatGPT 4o mini chatbot, such a request consisted of 219 characters.

The proposal could be used in practical activities to improve chatbot technologies and form key data sets in artificial intelligence systems, which would further make it possible to avoid errors when solving problems with a logical component

Keywords: Large Language Model, LLM, Copilot, ChatGPT, cyclic class element, attractor

Received 10.01.2025
Received in revised form 20.02.2025
Accepted date 13.03.2025
Published date 22.04.2025

How to Cite: *Kryazhych, O., Ivanov, I., Iushchenko, K., Kuprin, O., Vasenko, O., Riznyk, V., Ryzhkov, O. (2025). Devising an approach to preventing information chaos in chat bots using generative artificial intelligence. Eastern-European Journal of Enterprise Technologies, 2 (2 (134)), 84–95. https://doi.org/10.15587/1729-4061.2025.324957*

1. Introduction

Information chaos can be defined as the state of a system in the presence of an excessive amount of poorly structured, unsystematized information that is contradictory and unverified [1]. The problem of information chaos increased along with the increase in the number of Internet users who generated chaotic content. This issue was explained by using the problem of big data [2]. It was expected that the active development of artificial intelligence (AI) would simplify the task of processing large amounts of information [3]. However, it turned out that with the increase in the capabilities and access of users to services with artificial intelligence, expectations were not justified [4].

Operating in the concepts of chaos theory [5], we can consider the interaction of an Internet user and a service that works using AI tools as dynamic complex systems that are sensitive to initial conditions. In this case, even minor changes at the beginning can lead to significant differences in obtaining the result of the query. Such systems are deterministic because behavior at a certain point in time can be described by equations while long-term behavior is unpredictable.

In the field of AI, chaos can arise, especially when it comes to complex systems and algorithms. Recently, publications have been spreading on the Internet (for example, Oracle AI & Data Science Blog: Unpredictability of Artificial

Intelligence?) about the unpredictability of results obtained using AI. This is explained by the complexity of models and algorithms for processing big data, as well as directly by errors in the data or the lack of their systematization.

The improvement of language models of artificial intelligence [4], processes of user-chatbot interaction based on generative AI [6], and the development of algorithmization of reasoning processes [7] require AI tools not only to understand natural language and context but also to make inferences with subsequent application to real-world scenarios. This proves the relevance of the research topic, which requires study and analysis in order to devise approaches to prevent the emergence of information chaos during the interaction of a user with a chatbot or other tool using AI. The construction of appropriate models and methods will make it possible not only to improve the quality of knowledge bases of tools with generative artificial intelligence but also to directly improve machine learning algorithms. This will also contribute to the further development of Large Language Models (LLMs) of artificial intelligence with improved interaction in the user's natural language. This will make it possible to receive more accurate and meaningful answers to the user's request from chatbots with generative artificial intelligence.

2. Literature review and problem statement

Analysis of errors that occur in tools with generative artificial intelligence, in particular, in chatbots (Copilot, ChatGPT, and others) using chaos theory is a new approach to understanding processes in such complex systems. As noted in [8], the integration of traditional methods of chaos theory and machine learning models makes it possible to complement each other. In particular, owing to this approach, it is possible to analyze chaotic data sets in order to predict errors in the answers obtained using AI in the future. However, such conclusions are theoretical since there is not enough data for practical research on the issue. In addition, chaos itself remains a key problem of the study, because the analysis is carried out only on a complex AI system. A person, as a complex system, falls out of this contour of the problem, which complicates the understanding and possibilities of predicting the interaction of two complex systems. In [9], it is shown that chaos in AI is considered through the dynamics of extreme events in knowledge-based deep learning (KDL) models. It is shown that the study of complex patterns of chaotic systems is possible with simultaneous modeling of individual processes based on differential equations. This is possible even on the basis of small data. Researchers use prior knowledge about extreme meteorological events. The study remains unresolved on the issue of taking into account the human factor when transferring knowledge to the data processing system.

The use of tools for information retrieval and data processing in science, processing big data, and developing experiments leads to concern about the emerging unreliability of data. In [10], scientific achievements based on self-supervised learning are considered. Self-training of models on big data is demonstrated. It is shown that the problem of low data quality and data management remains relevant. The study leaves unresolved the issue of approaches that contribute to understanding the occurrence of errors in data processing using AI.

Papers [11, 12] focus on the errors of Large Language Models in applied applications. Thus, in [11], it is shown that GPT-4, Google, and Meta models can generate speech data on a very large scale. The design of LLM makes it possible to effectively generate natural human language but does not have emotions or understanding of thought processes. Based on this, the authors conclude that LLM is not ready for significant measurements, experiments, and processing the results of practical research. The study leaves unresolved the issues of developing key data sets, improving standards and benchmarks, performing calculations with subsequent analysis.

In [12], it is shown that the LLM model is able to reason when solving mathematical problems. The study uses the GSM-Symbolic test. It is noted that LLM demonstrates noticeable dispersion in different versions of the same task. For example, changes in numerical values, shifting landmarks in the question or increasing the number of question items in one task. In addition, the performance of LLM deteriorates to 65 % when logical reasoning is required. The study leaves unresolved the issue of taking into account the increase in information chaos with the specified changes in the representation of test tasks.

In [13], it is shown that natural language is a complex system, and its transmission is a coded process according to clear rules. It is noted that scaling the model can lead to an improvement in the capacity of the model. Extended language models not only achieve a significant improvement in performance but also demonstrate some special capabilities (for example, in contextual learning). It is emphasized that ChatGPT based on LLM, although it makes it possible to improve the accuracy of the answer, requires an improvement in the process of calculating the linguistic query construction model. The study leaves unresolved the issue of user query constructions, which determine the characteristics where the query is targeted at itself, and the use cases.

In [14], standard algorithms of tools with generative artificial intelligence still face problems of contradiction between accuracy and similarity of the answer to the entered task criteria. For the solution, it is proposed to use an algorithm for predicting connections between users and queries using similarity criteria. However, the accuracy/similarity ratio ensures the correspondence of the answer to the query but does not mean obtaining information according to the criterion of truth. That is, it is the constructions that cause chaos in the query system that are of interest.

In [15], the issue of data sets for training is investigated. The authors summarize the need to improve LLM training. However, the work does not take into account the need to overcome information chaos. The authors propose circumventing this by using additional training and expanding data sets.

In [16], LLM problems that arise due to the occurrence of information noise are clearly defined. This leads to limitations in the received answers, obtaining illogical conclusions, false and irrelevant information. However, the authors only analyze the problem, investigating the issue of the emergence of information noise.

In paper [17], the problems of data annotation in scientific articles are considered. It is noted that there are problems of information distortion and the emergence of information noise caused by small data sets on which LLMs are trained. To solve such problems, the authors propose data augmenters. However, the work presents a narrow understanding of the emergence of information chaos. This is due to the fact that the articles under study are written in only one natural language with strict adherence to the writing style. Changing these parameters will lead to excessive information, which can lead to greater problems with text annotation using AI.

The problems of user interaction with LLMs identified as a result of our review of the literature can be systematized in two directions. The first is the creation of a request to a chatbot using excessive information. In this case, the algorithm sequentially builds answers to each of the specified questions without focusing on the main one. The second is the use of both algebraic and mathematical components when formulating a request. This leads to the emergence of a little-studied process of information chaos, which negatively affects the formation of a response to a user's request.

Based on the above, the main task of our study is to find a solution to prevent the occurrence of information chaos when a user requests a chatbot based on generative AI. That is, the study of the user's request, as a result of which a certain construction is formed in natural language. This construction is interpreted by the chatbot using algorithms to a level understandable to the machine. Excessive information contributes to the formalization of the task in such a way that the answer becomes inaccurate, false, or emphasized on additional conditions.

Solving this problem will make it possible to improve algorithms for formalizing queries posed in natural language. In addition, solutions to prevent digital chaos in AI chatbots will make it possible to improve LLM performance.

3. The aim and objectives of the study

The purpose of our research is to devise an approach to preventing the occurrence of information chaos in chatbots using generative artificial intelligence. This will make it possible to formalize typical errors in answers received to a user's request when solving mathematical problems using chaos theory approaches and to prevent their occurrence in a timely manner.

To achieve the goal, the following tasks were set:

– to propose an information technology algorithm for constructing piecewise linear attractors for breaking down an information block when constructing a user's request to a chatbot in natural language;

– to present a methodology for practical implementation of breaking information into blocks to prevent information chaos when a user requests a chatbot that works on the basis of generative artificial intelligence.

4. The study materials and methods

The object of our study is the linguistic constructions of requests to chatbots with LLMs. A linguistic construction is considered to be a request constructed in the user's natural language with appropriate stylistic connections and the presence/absence of emotional coloring.

The main hypothesis of the study is the following statement: the more concise the linguistic construction according to certain strict rules, the clearer the answer to the request. This is due to the fact that clear rules are used to encode the request from the user's language by the chatbot's language model, since entropy does not arise or is minimal. That is, in this case, the LLM algorithms are not subject to the influence of information chaos.

The work is simplified by using two chatbots based on generative artificial intelligence for research: Copilot and ChatGPT 4o mini. Separate studies of queries formed in the user's natural language were conducted with other chatbots, but no substantive studies were conducted.

We assume that the user and the chatbot are separate complex systems, the events and actions of which are difficult to predict for a long period. In this regard, the behavior models of these complex systems can be described using chaos theory approaches. That is, the human user and the chatbot are sources of chaos with parameters that change over time.

Chaos sources with variable parameters have a wide range of schemes for introducing an information signal chaotically. In fact, the chaotic signal is modulated by the information one. There are a number of methods that allows the source of the information chaotic signal to be represented by a classical system of differential equations:

– Lorentz attractor [18]:

$$\begin{cases} \dot{x}_1 = \sigma\left(x_2 - x_1\right), \\ \dot{x}_2 = -x_1 x_3 + r x_1 - x_2, \\ \dot{x}_3 = x_1 x_2 - b x_3; \end{cases} \tag{1}$$

– Rösler's attractor [19]:

$$\begin{cases} \dot{x}_1 = -x_2 - x_3, \\ \dot{x}_2 = x_1 + a x_2, \\ \dot{x}_3 = c + x_3\left(x_1 - b\right); \end{cases} \tag{2}$$

– Chua's scheme [20]:

$$\begin{cases} \dot{x}_{d_1} = p\left[x_{d_2} - x_{d_1} + f\left(x_{d_1}\right) + s f_1\left(x_{d_1}\right)\right], \\ \dot{x}_{d_2} = x_{d_1} - x_{d_2} + x_{d_3}, \\ \dot{x}_{d_3} = -q x_{d_2}. \end{cases} \tag{3}$$

Systems with chaotic dynamics and described by equations (1) to (3) have a feature. Such systems can synchronize. This means that for a system $\dot{x} = X\left(t, x\right),\ x = \left(x_1, .., x_n\right)$ with chaotic dynamics, it is possible to construct a system with equations $\dot{y} = Y\left(t, y\right),\ y = \left(y_1, .., y_n\right)$. And in this case, under different initial conditions $\dot{x} = \left(x^0{}_1, .., x^0{}_n\right)$ and $\dot{y} = \left(y^0{}_1, .., y^0{}_n\right)$, their state vectors can be very close. That is, $\lim\limits_{t \to \infty}\left(x_i - y_i\right) = 0$, $\forall\, i = 1, .., n$. This circumstance is the basis for building an approach to preventing the occurrence of information chaos during the interaction of the user and the chatbot. And this process of preventing the occurrence of chaos can be implemented directly at the stage of encryption and decryption of the natural language of the user who enters a request to the chatbot.

In the described case, the interaction between the user and the chatbot can be represented through a combination of two complex systems, one of which generates a request, which from the point of view of AI is an encrypted message in the user's natural language. The second system, that is, the chatbot, decrypts the request. And this process can be represented by systems of differential equations (1) to (3), solving which will make it possible to obtain the most accurate possible answer to the query. Using dynamic chaos, the following techniques of such encoding/decoding of information can be used [1]:

a) conversion of the signal fed to the input of the transmitter system and recovery using an inverse dynamic receiver system;

b) parametric modulation;

c) use of masking.

When converting a signal to the dynamic equations of the transmitter system, a usable signal is introduced, thereby changing the original dynamics of the system. The output is transmitted over the communication channel – a fragment of the system state vector, which is reflected by the LLM rules. The receiver system is built in such a way that the difference between the state of the receiver and transmitter systems, which can be expressed in terms of the mismatch, does not depend on the unknown parameters of the request and decreases exponentially. Thus, having determined the conditions under which the mismatch tends to 0, it is possible to construct an answer to the user's request.

Parametric modulation in this case can be considered as using the request itself to obtain an answer to the request. This is a fairly common approach in LLMs [13] for forming a response. Modulation is carried out in the form of changing the system parameters. That is, the parameters take some maximum (when transmitting one) and minimum (when transmitting zero) values. Naturally, the system dynamics change during such "switches". When synchronizing systems, the response vector must be expanded by the parameters involved in the request to the chatbot system itself.

The use of masking when processing a user's natural language request by a chatbot occurs by simultaneously processing a fragment of the state vector and some combination of the usable signal with the components of the state vector. That is, the response vector from the chatbot as a result of synchronization will be close to the user's request vector.

For our study, cyclic piecewise linear attractors according to model (1) were selected, which make it possible to display the request signal to the chatbot and receive a response. A simple mathematical problem adapted for the test according to the approach from [12] was selected as the request. The essence of the problem was as follows: it was required to calculate how many apples the girl collected in three days when it is known how many apples she collected in one day. An insignificant remark in the user's natural language was added to the problem that in one day seven apples were half as many as those collected daily. As a result, the Copilot and ChatGPT chatbots, with the help of which the testing was carried out, reduced the number of apples from the specified daily number.

The computer experiment involved determining the optimal number of words, mathematical signs, and numbers, which together amounted to no more than 1000 characters in natural language, in the request. In this case, the answer to the request should be as accurate as possible. And also determining the maximum number of characters in a user request to a chatbot, which causes information chaos. That is, the user's request is answered incorrectly, inaccurately, and unreliable. In such a calculation, when breaking words into characters, a restriction was set not to violate the logical essence. Abbreviations in requests were assumed, as well as well-known abbreviations or replacing numerals with numbers. Language constructions are analyzed according to paper [6] with the creation of arrays and word samples for building a request.

To implement the experiment, an application was developed in the C++ programming language. The application, based on piecewise linear attractors, broke the user's request, consisting of a maximum of 137 words, into separate questions. These questions contained from 9 to 900 characters each. Nine characters are a simple arithmetic operation according to the conditions of the problem, according to which the testing was carried out (Fig. 1).

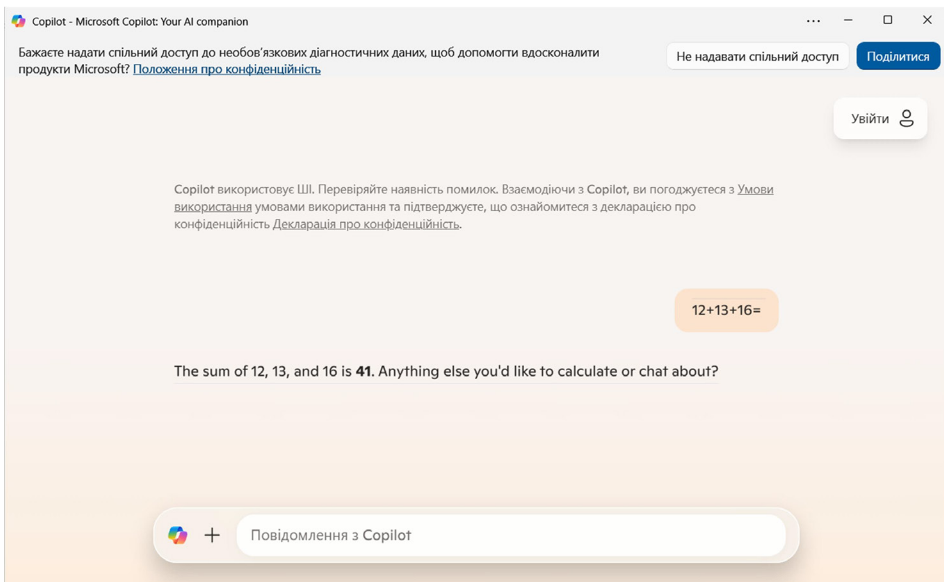


Fig. 1. An example of a simple arithmetic operation based on the conditions of the task for testing a request to a chatbot (screenshot)

The user's initial query consisted of 43 words or 161 characters. The initial query of a simple task was deepened with insignificant details – the number of apples by color, by shape, etc. Similarly, insignificant remarks of the task were discarded. That is, details were added or removed that created additional information noise [21], which artificially overloaded the user's query. For mathematical substantiation, a reference task of lexicographic order was chosen [22].

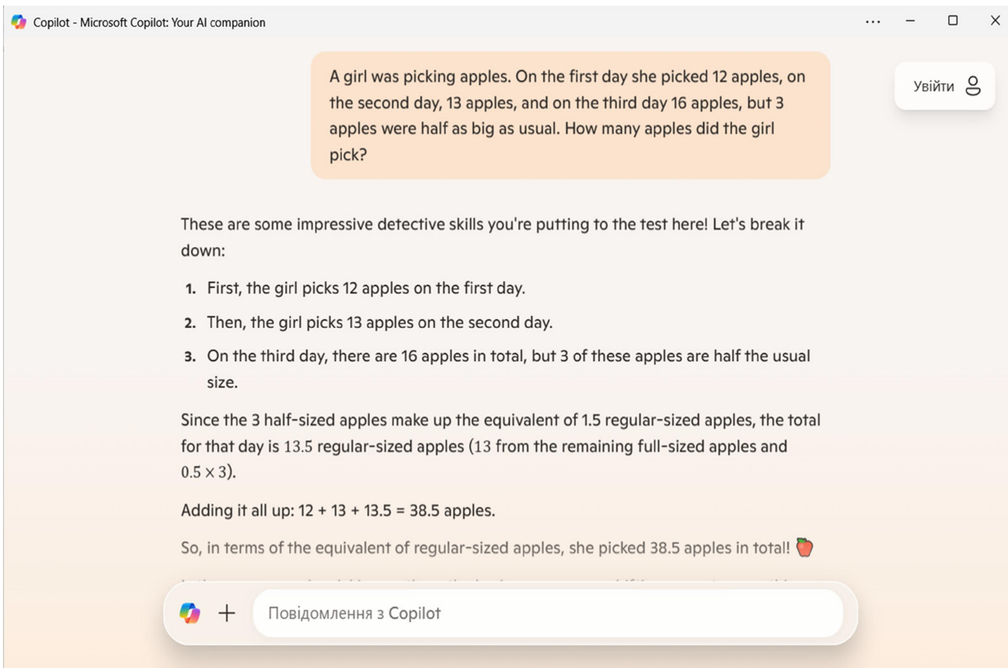
In the process of performing the computer experiment, the time of encryption of the query by the chatbot was studied, that is, the query analysis. After that, the response time was calculated – that is, the time of forming the answer and its display on the monitor screen. Then, the accuracy of the received answer was compared. For each variant of the adapted query, 2.5 thousand tests were performed. The visualization of the experiment results is represented in the form of a map of the occurrence of information chaos when the user forms a query to a chatbot with generative artificial intelligence.

5. Results of research on the development of a methodology for preventing the occurrence of information chaos during user-chatbot interaction

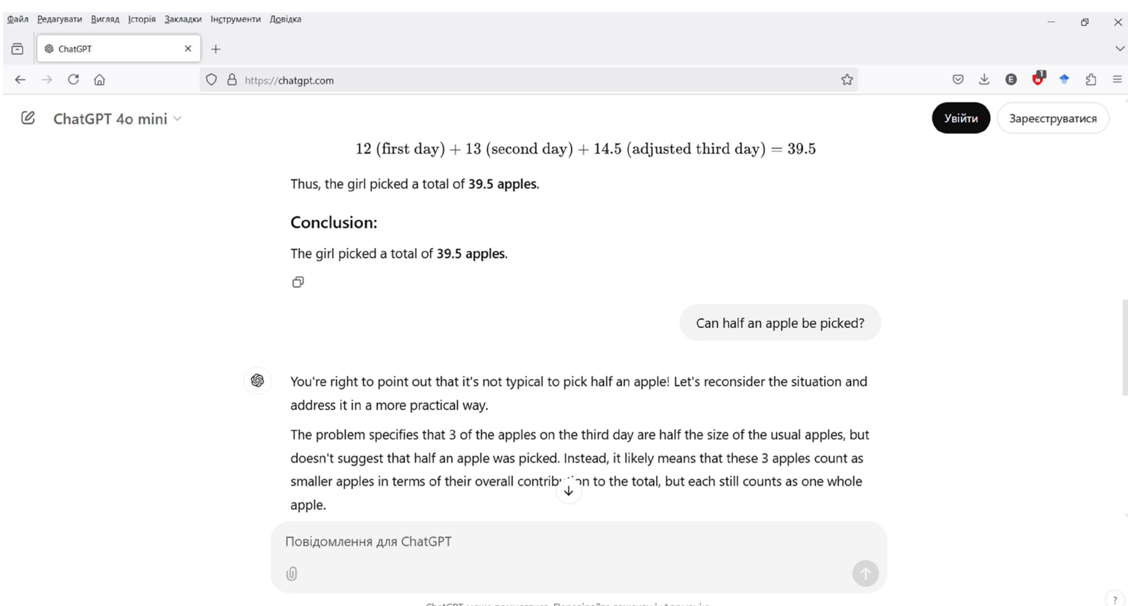
5. 1. Information technology algorithm for constructing piecewise linear attractors for splitting an information block when constructing a query

An example of the answer to the initial statement of the problem in the Copilot and ChatGPT 4o mini chatbots is shown in Fig. 2. From Fig. 2, you can even see the difference in the calculation results for each of the studied chatbots.

In the examples given, we assume that each word is a certain letter of the chatbot dictionary alphabet. Then the query record can be reduced to constructing a set of points in space that reflect the essence of the user's query. Each pair of words that are located next to each other and form a meaningful expression can be considered minimal segments – carriers of information about the essence of the query. In this case, piecewise linear one-dimensional mappings $x_{n+1}=f(x_n)$, $(n \epsilon N)$ can serve as a means for a unified implementation of permutations $a_{i1}\ a_{i2}\ ...a_{in}$ of letters of the alphabet $A=\{a_1, a_2, ..., a_n\}$.



a



b

Fig. 2. Example of an answer to the initial problem statement: *a* — Copilot; *b* — ChatGPT 4o mini (screenshot)

The essence of such permutations is as follows. Using the correspondence $a_j \leftrightarrow b_j=(j-0.5)\cdot n^{-1} \epsilon (0,\ 1)$, $(j=1\ldots,n)$ we make a transition from the letters $a_{i1}\ a_{i2}\ \ldots a_{in}$ belonging to the alphabet A to the permutation of the numbers $b_{i1}\ b_{i2}\ \ldots b_{in}$ belonging to the alphabet $(j=1\ldots,n)$. The permutation $b_{i1}\ b_{i2}\ \ldots b_{in}$ corresponds to the sequence of points $b_j=(b_{ij},\ b_{i(j+1)(n)})$, $(j=1\ldots,n)$ of the plane. The piecewise linear one-dimensional representation $x_{n+1}=f(x_n)$, $(n \epsilon N)$ on the interval $((j-1)\cdot n^{-1} j \cdot n^{-1})$, $(j=1\ldots,n)$ is determined by the line segment passing through the point b_j. This line has a slope s, $(0<s<1)$. If an interval is given that has an initial point b_i, then the representation of the function f can restore the permutation $b_{i1}\ b_{i2}\ \ldots b_{in}$. This is due to the movement along a stable cycle, which is defined by the points $b_{i1}\ b_{i2}\ \ldots b_{in}$. The stability of the cycle ensures that the condition $(0<s<1)$ is met.

It is worth emphasizing that the recovery of the information block $a_{i1}\ a_{i2}\ \ldots a_{in}$, which consists of different letters of the alphabet A, is a representation of the encoded user query. And this encoding is ensured by the one-to-one correspondence between this information block, the query in natural language, and the ordered pair of the starting point and the cycle, along which the search and formation of the response from the chatbot takes place.

The above approach is applicable to encoding any block of information of the user's query. For example, the block of information $C=c_1 \ldots c_l \epsilon A^1,(l \epsilon N)$, i.e., to placement in l-position with repetition in the alphabet A.

We assume that the block code $(C)=(a_1,\ldots,a_n)$, where $a_j\ \epsilon\ (j=1\ldots,\ n)$ is the number of occurrences of the letter $a_j \epsilon A$ in the block C. Thus, $a_j \epsilon Z_+$ and $\sum_{j=1}^{l} \alpha_j = l$. By $a_{j1},\ldots a_{jh}$, $(1 \le j_1 < \ldots < j \le n)$ we denote the letters of the alphabet A, which correspond to the zero components of the vector code C, and by $\pi(C)=\{B_1,\ldots,\ B_h\}$. That is, there is an h-block partition of some set $N_l=\{1,\ldots,\ l\}$, where $|B_r|=a_{jr}$.

Taking into account the above, we can represent the algorithm for the procedure of permutation of query words, denoted by the symbols C, D, N, G, in the form of the following code:

```
Procedure CDNG(C)
     begin
       code(C)=Θ
          do i=1,...,l
             do r=1,...,h
               if cj=ajr
                  then
                      code (C)=code(C) ‖, Bjr=Bj\
               end_if
             end_do
           end_do
       end_begin
```

It is clear that the block code(C) obtained as a result of applying the procedure CDNJ to the block C is a permutation of words in some block N_j of the user's query, made in natural language.

Denote by $S(l)$ a symmetric group, which is defined on the set N_j. Any cycle, which is defined by the sequence $b_1,\ b_2,\ \ldots,\ b_n$ of points of the plane $b_j = \left(b_{i_j}, b_{i_{(j+1)(l)}}\right)$ $(j=1\ldots,n)$, where $b_{i1}\ b_{i2}\ \ldots b_{il}$ is a permutation of words $b_j=(j-0.5)\cdot l^{-1}$, $(0,\ 1)$, $(j=1\ldots,n)$, $(j=1\ldots,n)$. This cycle is characterized by the substitution of other query words $(i_1\ i_2\ i_2\ i_3\ \ldots\ i_{l-1}\ i_l\ i_l\ i_1) \epsilon S(l)$, which belong to the user's query $\{1^0 2^0\ \ldots\ l^1\}$. The established mutual one-to-one correspondence makes it possible to represent the user's query in the form of substitutions l – elements of blocks of information (words, expressions) that were not used by the user when creating the query. But these l belong to the cyclic class $\{1^0 2^0\ \ldots\ l^1\}$ of the symmetric group $S(l)$ of the user's query. That is, owing to such elements of information, the user's query can be cleaned of information noise, made more accurate and concise. The scheme of the algorithm for such query transformation is shown in the form of a diagram in Fig. 3.

The variety of transformations of the user's initial query is determined by the set of mechanisms for controlling the numbering of letters of the alphabet A. That is, the transformation of the initial query is determined by the set of mechanisms for selecting the partition $\pi(C)$. This is also influenced by the set of mechanisms for creating expressions from elements of the cyclic class $\{1^0 2^0\ \ldots\ 1^1\}$. The number of numbering of letters of the alphabet A is equal to $n!$. The selection of the partition $\pi(C)=\{B_1,\ldots,\ B_h\}$ can be carried out directly by $\frac{l!}{\alpha_{j_1}! \cdot \alpha_{j_2} \cdot \ldots \cdot \alpha_{j_r}!}$ techniques. The number of possible effective permutations of words belonging to elements of the cyclic class $\{1^0 2^0\ \ldots\ l^1\}$ is not less than $(l-1)!$.

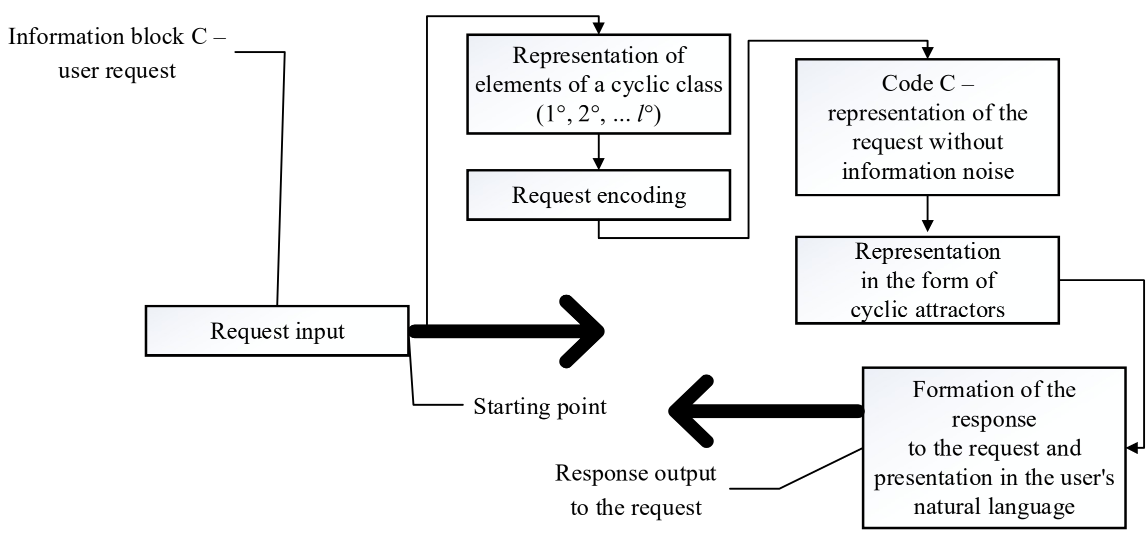


Fig. 3. Schematic of the algorithm for transforming a user query to remove information noise based on cyclic attractors of piecewise linear representations

Therefore, controlling the numbering of letters of the alphabet A in combination with the representation of the information block by elements of the cyclic class $\{1^0 2^0 \ldots l^1\}$ completely eliminates the possibility of frequency analysis. Accordingly, the proposed scheme allows for the systematic construction of computationally stable statements by organizing the management of the independent selection of substitutes from the available sets of the base dictionary.

5. 2. Methodology of information division into blocks to prevent information chaos at the user request

The class of mechanisms for controlling the selection of substitute words for the user request presented to the system allows for the generation of the next replacement for a given substitution $(1v_1\ 2v_2 \ldots kv_k)\epsilon S(k)$. It also makes it possible to calculate the next replacement $(1v_1\ 2v_2 \ldots kv_k)\epsilon S(k)$. For this purpose, we can consider operations with a fixed ordering of elements of the symmetric group $S(k)$. This can be considered on the example of a reference problem of lexicographic order in the following form:

$$\left(1v_1^{(1)}\,2v_2^{(1)}\cdots k\,v_k^{(1)}\right)\prec\left(1v_1^{(2)}\,2v_2^{(2)}\cdots k\,v_k^{(2)}\right)\Leftrightarrow$$
$$\Leftrightarrow\left(\exists i\in\{0,1,\ldots,k-1\}\right)\left(\forall j\in\{1,\ldots,i\}\right)\times$$
$$\times\left(v_j^{(1)}=v_j^{(2)}\ \&\ v_{j+1}^{(1)}<v_{j+1}^{(2)}\right). \qquad (4)$$

When analyzing problem (4), it is assumed that the order is cyclic. To simplify problem (4), the substitution $(1v_1\ 2v_2 \ldots kv_k)\epsilon S(k)$ can be immediately replaced by the permutation $(v_1\ v_2 \ldots v_k)$ of words from the elements of the set N_k.

Taking into account the above, the generation of word substitutions in the user's query in natural language will occur in sequence γ cyclically. The reverse order of such a sequence is γ^{-1}. The generation of query word substitutions is carried out within the memory limits of $O(k)$, and considering that this is generative artificial intelligence, then $(k\rightarrow\infty)$.

Then the procedure for generating NEXT word substitutions is as follows:

```
Procedure NEXT (v1 v2 … vk)
      begin
        if v1 v2 … vk = k(k – 1) …21
          then next (v1 v2 … vk) = 12 …(k – 1)k and HALT
            else i = k – 1 and go to LABEL1
        end_if
        LABEL1: if v1 > v_{i+1}
               then i = i – 1 and go to LABEL1
                       else go to LABEL2
                   end_if
        LABEL2: ζ = v1 …v_{i-1}, j = i + 1 and go to LABEL3
        LABEL3: if v_j > v_i
               then go to LABEL4
                       else j = i – 1 and go to LABEL6
                   end_if
        LABEL4: j = i + 1
                    if j ≤ k
                      then go to LABEL3
                        else go to LABEL5
                    end_if
        LABEL5: β = v_{i+1} …v_k, next (v1 v2 … vk) = ζ|| β^{-1}||v_i
and HALT
        LABEL6: ξ = v_{i+1} …v_{j-1}, β = v_{i+1} …v_k,
                          next(v1 v2 … vk) = ζ||v_j||β^{-1}||v_i||ξ^{-1} and
HALT
      end_begin
```

Based on the above algorithm, represented in C++, it is clear that the time and capacity complexity of the replacement procedure *NEXT* is $O(k)$, $(k\rightarrow\infty)$. This will determine the processing time of the query given in the user's natural language, understanding the decoded essence of the query, and forming the correct response to the query.

The calculation of the next word replacement in the NMBR procedure with the permutation $v_1\ v_2 \ldots v_k$ of the elements of the set N_k is calculated within the same memory capacity:

```
Procedure NMBR (v1 v2 … vk)
      begin
        nmbr = 0, h = k
         LABEL1: if h = 1
                      then nmbr = nmbr + 1 and HALT
                        else go to LABEL2
                 end_if
         LABEL2: do j = 1, …, h – 1
                    if v_{j+1} > v1
                      then μj = v_{j+1} – 1
                        else μj = v_{j+1}
                    end_if
                  end_do
         nmbr = (v1 – 1)·(h – 1)!, h = h – 1
         do j = 1, …, h
           vj = μj
         end_do
         go to LABEL1
      end_begin
```

The permutation $v_1\ v_2 \ldots v_k$, with number *nmbr* is recovered with capacitive computational complexity using the following procedure.

```
Procedure PN (nmbr)
      begin
      pn = 12…(k – 1)k
        do j = 1,…, mnbr – 1
          pn = NEXT(pn)
        end_do
      end_begin
```

The time complexity of the PN procedure is $O(k!)$, $(k\rightarrow\infty)$.

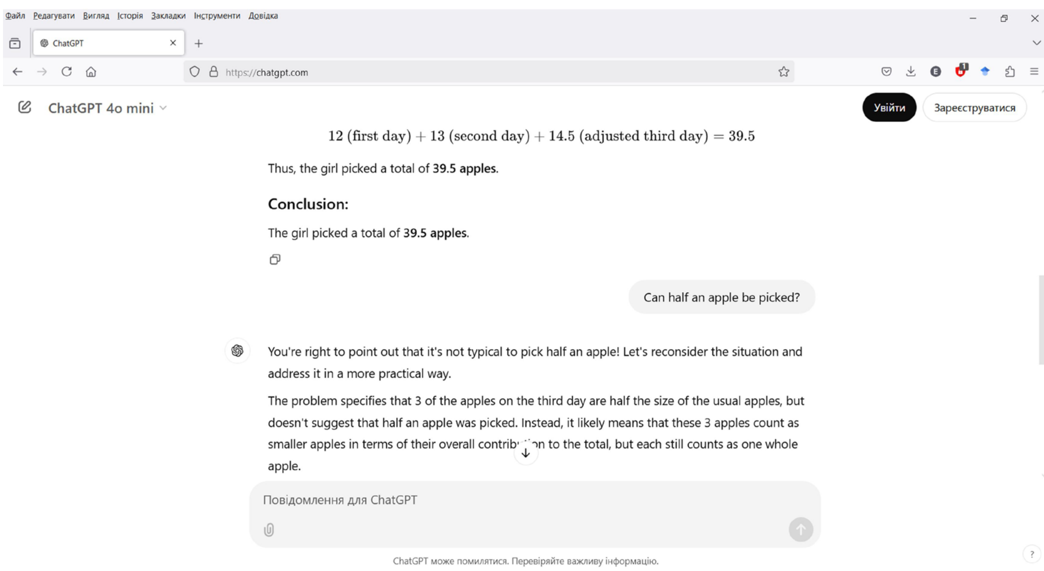
Below are examples of answers from Copilot (Fig. 4, *a*) and ChatGPT 4o mini (Fig. 4, *b*) to the initial problem statement of 43 words or 161 characters. The answer was obtained after additional clarification of the initial condition of the problem – can half of the apple be collected?

Dividing the text into blocks using cyclic attractors of piecewise linear representations with corresponding word substitutions makes it possible to obtain a number of transformations of the initial user query. As a result, we obtained a user query with word and digit substitutions that make it possible to maximally rid the query of information noise and prevent the occurrence of digital chaos (Fig. 5).

The best version of the request to the Copilot chatbot has 182 characters, which is represented by 48 words, numbers, and special characters. The best version of the request to the ChatGPT 4o mini chatbot has 219 characters.

Based on the experiment on the task, maps of the occurrence of information chaos when forming a request by a user to a chatbot with generative artificial intelligence were built (Fig. 6).

a



b

Fig. 4. Examples of answers after additional clarification of the task conditions:
a — Copilot; *b* — ChatGPT 4o mini (screenshot)

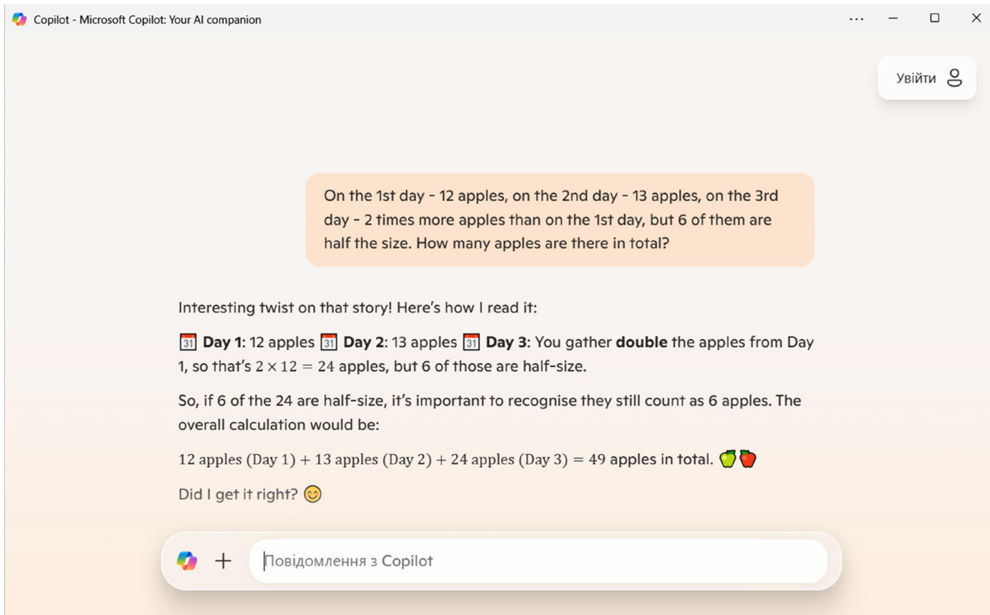
In Fig. 6, the gradation of green (Fig. 6, *a*) and red (Fig. 6, *b*) colors changes in the direction of the arrow with darkening. Muted colors mean the emergence of information noise at first. After that, the gradation changes to black, which indicates the emergence of information chaos. Additional explanations of questions lead to some improvement in the understanding of the person and the chat-bot with AI. However, subsequent attempts to obtain an improved answer through additional expansion and deepening of the question again lead to information noise (Fig. 6, *a*) and information chaos (Fig. 6, *b*).

In order to estimate the processing time of the user's request and the formation of the response to the request, a computational experiment was conducted for various changes in the query variant. The results of the experiment, characterizing the studied trends, are given in Table 1.

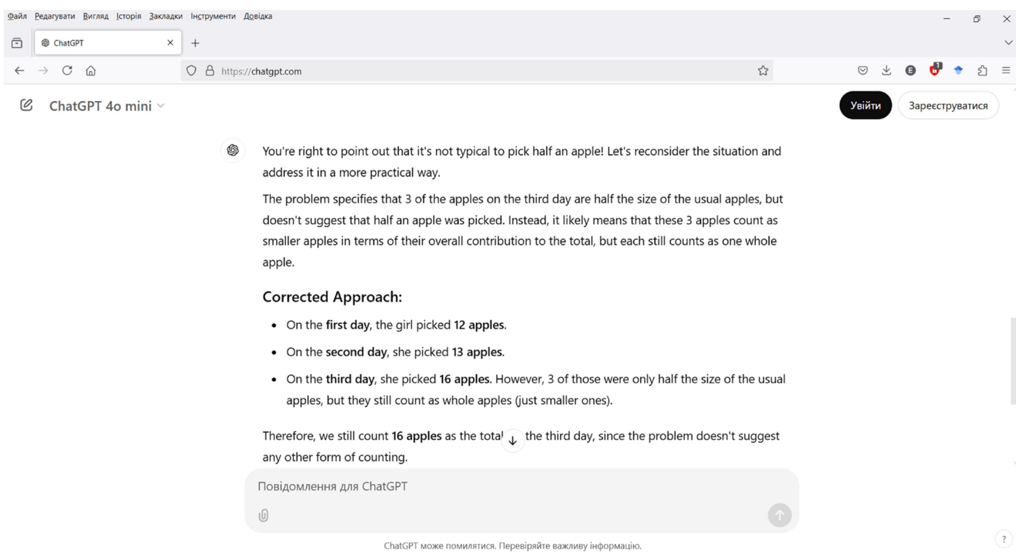
Table 1

Results of the computational experiment

No.	Number of characters in the request	Request processing time	Time to generate a response to the request	The meaning of the answer (true/false)	Percentage of correct answers
1	9	to 1 s	to 2 s	true	100
2	10–18	to 1 s	to 2 s	false	28.7
3	82–90	1 s	to 8 s	true	76.1
4	181–244	1 s	to 12 s	true	92.3
5	893–1,000	2 s	over 12 s	false	22.5

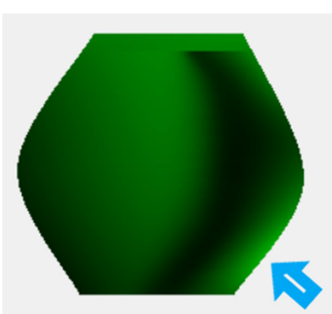


a

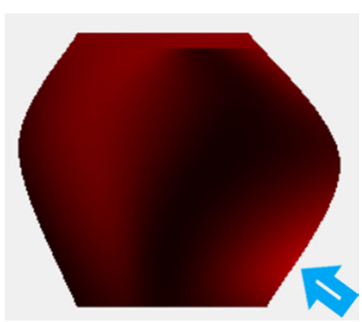


b

Fig. 5. Improved user query based on the following approach: *a* – Copilot; *b* – ChatGPT 4o mini (screenshot)



a



b

Fig. 6. Map of the occurrence of information chaos when a user forms a request to a chatbot with generative artificial intelligence: *a* – Copilot; *b* – ChatGPT 4o mini (the arrow indicates the direction of the increase in the number of characters in the request)

Based on Table 1, the best result for a query to solve the given problem is achieved when using 181–244 characters. These are 42–62 words of a query created in the user's natural language.

6. Discussion of results related to devising an approach to preventing information chaos in chatbots

At the beginning of the experiment, a simple mathematical problem about the number of apples collected by a girl is complicated by questions that are not essential from the user's point of view. However, for a chatbot working under LLM, each question in the problem is essential since it is defined in the conditions. An example of the answer in the chatbots Copilot (Fig. 2, *a*) and ChatGPT 4o mini (Fig. 2, *b*) demonstrates different algorithms for interpreting the problem. But they provide an equally incorrect answer when it is necessary to apply logic in solving the problem, which complements paper [12].

Analysis of the problem specified in [12] through the theory of digital chaos makes it possible to confirm the results reported in [9] on the possibility of modeling the behavior of chaotic systems in order to prevent the occurrence of chaotic phenomena. In particular, the software implementation of the method for preventing the emergence of digital chaos when using chatbots makes it possible to significantly improve, in contrast to [9], the communication process in two stages. At the first stage, it is possible to break the query in natural language into separate fragments without losing the essence. At the second stage, it is necessary to determine the mechanism of substitution and permutation of query fragments for the most accurate understanding of the chatbot by the LLM model.

The scheme of transforming the user query for cleaning from information noise based on cyclic attractors of piecewise linear representations (Fig. 3) allows the user to perform such permutations in the text independently. This will make it possible to change the approach proposed in [17]. However, according to the proposed approach, it is not worth expanding the data sets because such actions can lead to greater information chaos. This is due to the fact that this will resemble the actions of users when creating a search query on the Internet, where one question is replaced by another without losing the original idea. The next step, represented by the NEXT algorithm, makes it possible to avoid this. In this algorithm, an important factor is that for the information block C there is an opportunity to organize a non-stationary choice of partition. That is, $\pi(C)=\{B_1,..., B_h\}$ using an irrational process, which has a time and volume complexity of calculations $O(l)$, $(l\rightarrow\infty)$. This is manifested during the first partition of the information block $C=c_1...c\epsilon A^l$. It is then that the motion vector of the partition $(a_1, ..., a_n)$ appears, where $a_{j1}, ..., a_{jh}$, $(1\leq j_1<...<j_h\leq n)$ are letters of the alphabet A, which correspond to non-zero components of the vector. Then, a step-by-step partitioning occurs, provided that there is an ordered pair that carries a semantic load. This makes it possible to improve the approach proposed in [15] and avoid the occurrence of information noise, as defined in [16].

The calculation of the next word replacement according to the proposed methodology is designed to significantly reduce the formation of information chaos in the query. However, analyzing the algorithm of the next word permutations PN (numb), it can be seen that the volume of calculations increases. This significantly slows down the process of forming a response to the query with an increase in the number of characters l of the information block C, which can be considered a drawback of the methodology.

However, the braking of the specified process can be prevented by fixing the permissible block of information. That is, the length of the block C is fixed by the number of permissible characters. An additional worksheet is also built, in which, with a fixed value of $K\epsilon N$, substitutions with numbers $i\cdot K^{-1}\cdot k!$, $(i=1, 2, ..., K-1)$ are calculated. This explains the limitation of 1000 characters of the query text for conducting the experiment.

Additional clarification of the problem conditions, illustrated in Fig. 4, makes it possible to see the manifestation of discrepancies between the answers to the question that creates information chaos. Whereas in Copilot (Fig. 4, *a*) the process of minimizing the chaotic effects of excessive information is already underway, then ChatGPT 4o mini (Fig. 4, *b*) demonstrates almost the same incorrect result. This confirms the conclusions in [11, 12] regarding the fact that in GPT-4 the LLM construction has difficulty perceiving the need for logical thinking. Therefore, in this case, we can agree with the researchers [11] that the improvement of chatbots working with LLM is possible through the development of key data sets necessary for solving problems with a logical basis.

The correct answer to the query, presented in Fig. 5, has texts of different lengths for Copilot (Fig. 5, *a*) and ChatGPT 4o mini (Fig. 5, *b*). As can be seen from Fig. 5, *a*, for Copilot the correct answer is based on a pair of words located next to each other from the text block "only apples". For ChatGPT 4o mini (Fig. 5, *b*), the use of this paired expression turned out to be insufficient. The correct answer is obtained only after using pairs of words that give a specific indication of ignoring additional conditions.

Fig. 6 makes it possible to consider maps of the occurrence of information chaos when a user forms a query to a chatbot with generative artificial intelligence. In fact, this is a demonstration of the process of self-training chatbot models on big data, as noted in [10]. Dark spots in both figures indicate those queries that caused information chaos when formulating the answer. Again, in the Copilot chatbot (Fig. 6, *a*), reducing the number of characters in the query and representing more concise text blocks led to a better result than the same queries when addressing ChatGPT 4o mini (Fig. 6, *b*). The latter confirms the results reported in [13] that the transmission of the user's natural language is a coded process according to clear rules. Therefore, to organize better interaction between the user and the chatbot, it is necessary to take into account that they are complex systems.

Analyzing Table 1, it is worth noting that the table presents a sample of the best results, which characterize the methodology for preventing the occurrence of information chaos. The omitted samples of the used symbols acquire average values, characteristic of the user level of queries. That is, when creating queries for a certain number of characters/words, the user receives a moderately correct result that satisfies him/her.

Our results can be explained by the fact that an insufficiently formed query condition (Table 1, line 2) or a query overloaded with additional conditions (Table 1, line 5) creates information chaos. However, in order to prevent such chaos, some conditions should be taken into account in each case. Therefore, we can agree with the authors of [8] that the integration of chaos theory and machine learning models makes it possible to complement each other, improving the interaction between the user and machine intelligence.

The advantages of the proposed solutions compared to similar results are the shift in emphasis from the need to create additional data sets for training chatbots working on LLMs to adding new rules. These rules will contribute to understanding in the human-machine interaction system and simplify the algorithmization of the LLM learning process.

Our solutions resolve the problem by dividing it into attractors – blocks of text of small length with a given meaning. Analysis of the results of the computational experiment suggests that the decryption time increases exponentially with the number of characters in the source text. This situation can be eliminated using the following two approaches:

1) permutations are tabulated explicitly in advance (rather than generated by software) and fast block splitting algorithms are used;

2) the source text is divided into small blocks when creating a query. When implementing and testing one approach,

it was found that the processing and response times are approximately the same. For example, as in Table 1, shown in line 1, where the usual mathematical operation of addition was used to describe the condition of the problem. However, if the number of characters increases, the processing time of the query and the formation of the response increases. That is why in normal user practice it is worth using a combination of query formation in different variants, including the use of small text blocks. This prevents the occurrence of information chaos during the interaction of a person and a chatbot with a generative AI. In this case, when formulating a query, the user creates a construction in natural language that remains understandable to the machine. This is achieved by dividing the language construction into separate attractors that hold the semantic vector. In other words, our stated research problem has been solved.

The limitations of our study are the use of two modes of user interaction with an AI-based chatbot. The proposed scheme for transforming a user query to clean it from information noise based on cyclic attractors of piecewise-linear representations is implemented in C++. That is, this is an application that simulates the process of constructing a user query with transmission according to the rules of the user's natural language. The user chooses one of two modes, namely, splitting or permutation. After that, the user can specify paths to three files intended for recording the original information, for recording the recoded expression, and for recording the converted information. The main factors affecting the execution time of query transformation processes are the number of characters in the text, the block length, recursive algorithms for generating permutations and their numbers.

The disadvantages of the study are the use of one logical factor with many concomitant query factors that create information noise. As the logical branches of the question increase when forming a query to a chatbot, the minimum number of characters will change. Query variability is a variable value. This will force the use of adaptive algorithms in practical work.

This research may be advanced by improving the class of mechanisms for controlling the selection of substitute words according to rules. This can be based on the operations of generating the next substitution and calculating the number of a given substitution, showing that the most complex in terms of time are fragments associated with the application of the 2nd operation. This follows from model (4). The reason for this is the internal complexity of the connections between the lexicographic order and the numbering consistent with it. It is worth noting that some (albeit insignificant) reduction in the volume of calculations can be ensured by using additional information (parity/oddity of the permutation, the number of inversions, etc.).

The conditions for applying our research results are the introduction of additional rules to LLMs regarding the restriction of the query by the number of characters. This applies to the statement of problems with a logical component – the LLM model must "understand" it and separate it when formulating the answer.

The devised approach could be used to improve the technology of chatbots with generative artificial intelligence as a self-learning system. It will also contribute to the development of key data sets for further solving various problems, including those with a logical component. A similar approach could be used to improve reference and search systems on the Internet, create interactive textbooks and manuals, as well as improve online courses that use chatbots as tutors.

7. Conclusions

1. An information technology algorithm for constructing piecewise linear attractors for splitting an information block when constructing a user request to a chatbot in the user's natural language has been proposed. The features of the developed algorithm are the splitting of the user's initial request to a chatbot into paired semantic expressions according to the text of the request. This makes it possible to clean the information message from information noise, which ultimately leads to chaos. The task has been implemented using cyclic attractors of piecewise linear representations.

2. A methodology for practical implementation of information division into blocks has been devised to prevent information chaos when a user requests a chatbot. A feature of the methodology is the mechanism for controlling the selection of substitute words, based on the operations of generating the next substitution and calculating the number of the given substitution. This makes it possible to build a query in the form of a coded process in natural language according to clear rules. The best version of the query to the Copilot chatbot has 182 characters, which is represented by 48 words, numbers, and special characters. The best version of the query to the ChatGPT 4o mini chatbot has 219 characters. The proposed methodology could be used in practical activities to improve chatbot technologies that work on the basis of generative artificial intelligence, to form key data sets of artificial intelligence systems for solving problems with a logical component. It is also possible to use the approach to improve reference and search systems in educational activities.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

All data are available, either in numerical or graphical form, in the main text of the manuscript.

Use of artificial intelligence

The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

Acknowledgments

The authors express their gratitude to employees at the Educational and Methodological Laboratory of Digital Education and Artificial Intelligence, the Grigoriy Skovoroda University in Pereyaslav, for their assistance in conducting the computational experiment and testing the devised approach.

References

1. Errichiello, O., Wernke, M. (2022). Order in Chaos - Cybernetics of Brand Management. Springer. https://doi.org/10.1007/978-3-662-65958-8
2. Cappa, F., Oriani, R., Peruffo, E., McCarthy, I. (2020). Big Data for Creating and Capturing Value in the Digitalized Environment: Unpacking the Effects of Volume, Variety, and Veracity on Firm Performance. Journal of Product Innovation Management, 38 (1), 49–67. https://doi.org/10.1111/jpim.12545
3. Saadia, D. (2021). Integration of Cloud Computing, Big Data, Artificial Intelligence, and Internet of Things. International Journal of Web-Based Learning and Teaching Technologies, 16 (1), 10–17. https://doi.org/10.4018/ijwltt.2021010102
4. Junaid, M., Ali, S., Siddiqui, I. F., Nam, C., Qureshi, N. M. F., Kim, J., Shin, D. R. (2022). Performance Evaluation of Data-driven Intelligent Algorithms for Big data Ecosystem. Wireless Personal Communications, 126 (3), 2403–2423. https://doi.org/10.1007/s11277-021-09362-7
5. Fuchs, A. (2013). Nonlinear Dynamics in Complex Systems. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-33552-5
6. Kryazhych, O., Vasenko, O., Isak, L., Havrylov, I., Gren, Y. (2024). Construction of a model for matching user's linguistic structures to a chat-bot language model. Eastern-European Journal of Enterprise Technologies, 3 (2 (129)), 34–41. https://doi.org/10.15587/1729-4061.2024.304048
7. Gravett, W. (2021). Sentenced by an algorithm – Bias and lack of accuracy in risk-assessment software in the United States criminal justice system. South African Journal of Criminal Justice, 34 (1), 31–54. https://doi.org/10.47348/sacj/v34/i1a2
8. Taghia, J. (2024). Exploring the Synergy of Chaos Theory and AI: Predictive Modeling and Understanding of Complex Systems Through Machine Learning and Deep Neural Networks Review. COJ Robotics & Artificial Intelligence, 3 (4). Available at: https://crimsonpublishers.com/cojra/fulltext/COJRA.000566.php
9. Elabid, Z., Chakraborty, T., Hadid, A. (2022). Knowledge-based Deep Learning for Modeling Chaotic Systems. 2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA), 1203–1209. https://doi.org/10.1109/icmla55696.2022.00194
10. Wang, H., Fu, T., Du, Y., Gao, W., Huang, K., Liu, Z. et al. (2023). Scientific discovery in the age of artificial intelligence. Nature, 620 (7972), 47–60. https://doi.org/10.1038/s41586-023-06221-2
11. Demszky, D., Yang, D., Yeager, D. S., Bryan, C. J., Clapper, M., Chandhok, S. et al. (2023). Using large language models in psychology. Nature Reviews Psychology. https://doi.org/10.1038/s44159-023-00241-5
12. Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., Farajtabar, M. (2024). GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. arXiv. https://doi.org/10.48550/arXiv.2410.05229
13. Zhao, W., Zhou, K., Junyi, L., Tianyi, T., Wang, X., Hou, Y. et al. (2023). A Survey of Large Language Models. arXiv. https://doi.org/10.48550/arXiv.2303.18223
14. Ai, J., Cai, Y., Su, Z., Zhang, K., Peng, D., Chen, Q. (2022). Predicting user-item links in recommender systems based on similarity-network resource allocation. Chaos, Solitons & Fractals, 158, 112032. https://doi.org/10.1016/j.chaos.2022.112032
15. Du, F., Ma, X.-J., Yang, J.-R., Liu, Y., Luo, C.-R., Wang, X.-B. et al. (2024). A Survey of LLM Datasets: From Autoregressive Model to AI Chatbot. Journal of Computer Science and Technology, 39 (3), 542–566. https://doi.org/10.1007/s11390-024-3767-3
16. Matsumoto, N., Moran, J., Choi, H., Hernandez, M. E., Venkatesan, M., Wang, P., Moore, J. H. (2024). KRAGEN: a knowledge graph-enhanced RAG framework for biomedical problem solving using large language models. Bioinformatics, 40 (6). https://doi.org/10.1093/bioinformatics/btae353
17. Zhang, Y., Zhang, C. (2024). Extracting problem and method sentence from scientific papers: a context-enhanced transformer using formulaic expression desensitization. Scientometrics, 129 (6), 3433–3468. https://doi.org/10.1007/s11192-024-05048-6
18. Shen, B.-W., Pielke, R. A., Zeng, X., Baik, J.-J., Faghih-Naini, S., Cui, J. et al. (2021). Is Weather Chaotic? Coexisting Chaotic and Non-chaotic Attractors Within Lorenz Models. 13th Chaotic Modeling and Simulation International Conference, 805–825. https://doi.org/10.1007/978-3-030-70795-8_57
19. Martines-Arano, H., García-Pérez, B. E., Vidales-Hurtado, M. A., Trejo-Valdez, M., Hernández-Gómez, L. H., Torres-Torres, C. (2019). Chaotic Signatures Exhibited by Plasmonic Effects in Au Nanoparticles with Cells. Sensors, 19 (21), 4728. https://doi.org/10.3390/s19214728
20. Stankevich, N. V., Kuznetsov, N. V., Leonov, G. A., Chua, L. O. (2017). Scenario of the Birth of Hidden Attractors in the Chua Circuit. International Journal of Bifurcation and Chaos, 27 (12), 1730038. https://doi.org/10.1142/s0218127417300385
21. Voronka, H. V. (2014). Informatsiynyi shum u dovidkovykh vydanniakh. Obriyi drukarstva, 1, 89–95. Available at: https://horizons.vpi.kpi.ua/article/view/95451
22. Anh, L. Q. (2021). Various Types of Well-Posedness for Vector Equilibrium Problems with Respect to the Lexicographic Order. Vietnam Journal of Mathematics, 51 (2), 397–414. https://doi.org/10.1007/s10013-021-00530-7