

Оригинальная статья / Original article

УДК 004.942

<https://doi.org/10.21869/2223-1560-2025-29-3-113-123>

Когнитивный алгоритм и программа конспектирования текста

Л. А. Лисицин¹✉, А.Л. Лисицин², Д.Н. Блоха¹

¹ Юго-Западный государственный университет
ул. 50 лет Октября, д. 94, г. Курск 305040, Российская Федерация

² Курский государственный университет
ул. Радищева, д. 33, г. Курск 305000, Российская Федерация

✉ e-mail: leo_263@mail.ru

Резюме

Цель исследования. Исследование направлено на изучение возможностей применения когнитивных технологий (КТ) для эффективного решения плохо формализуемых задач в сфере анализа больших объемов текстовых данных. Особое внимание уделяется задаче автоматического конспектирования текстов — одной из важнейших проблем современной науки и техники, стимулирующей активное развитие методов искусственного интеллекта и машинного обучения.

Методы. Для достижения поставленной цели применялся извлекательный метод автоматического составления конспекта. Этот подход предполагает выбор наиболее значимых фрагментов исходного текста путем выделения отдельных предложений или фраз на основании определенных критерии. Использовались следующие критерии отбора: частота встречаемости слов; семантическая важность слов и выражений; позиция предложений внутри документа. Эти показатели позволяют выделить основные мысли текста и создать компактное резюме, сохраняющее смысл исходного материала.

Результаты. В ходе экспериментов была разработана программа на языке программирования Python, реализующая извлеченный метод конспектирования. Алгоритм основан на анализе частоты появления ключевых слов в тексте, что обеспечивает эффективную оценку значимости каждого предложения. Программа обладает рядом преимуществ: простота реализации и эксплуатации; открытый исходный код, позволяющий легко адаптировать решение под конкретные нужды пользователей; высокая эффективность при обработке значительных объемов текстовой информации. Разработанный инструмент демонстрирует способность эффективно создавать краткое изложение текста, сохраняя основную смысловую нагрузку и структуру оригинального содержания.

Заключение. Использование когнитивного алгоритма позволило значительно повысить производительность процесса анализа и обработки текстовой информации. Предложенная методика способна автоматизировать рутинные операции по составлению конспектов, помогая специалистам быстро получать общее представление о содержании крупных документов и публикаций. Это особенно актуально в условиях современного информационного общества, характеризующегося постоянно растущими потоками данных, которые требуют быстрого и качественного осмысливания. Таким образом, исследование показало перспективность внедрения когнитивных технологий в сферу автоматизации интеллектуального труда и предложило практическое решение актуальной проблемы конспектирования больших объемов информации, которое может стать важным помощником для специалистов и исследователей.

Ключевые слова: когнитивное моделирование; извлекательное конспектирование; частотный анализ; программирование; обработка естественного языка (NLP).

Конфликт интересов: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Для цитирования: Лисицин Л. А., Лисицин А.Л., Блоха Д. Н. Когнитивный алгоритм и программа конспектирования текста // Известия Юго-Западного государственного университета. 2025; 29(3): 113-123. <https://doi.org/10.21869/-2223-1560-2025-29-3-113-123>.

Поступила в редакцию 17.04.2025

Подписана в печать 24.07.2025

Опубликована 30.09.2025

A cognitive algorithm and a text annotation program

Leonid A. Lisitsin ¹✉, Alexander L. Lisitsin ², Dmitry N. Blokha ¹

¹ Southwest State University
50 Let Oktyabrya str. 94, Kursk 305040, Russian Federation

² Kursk State University
33, Radishcheva str., Kursk 305000, Russian Federation

✉ e-mail: leo_263@mail.ru

Abstract

Purpose of research. The research is aimed at exploring the possibilities of using cognitive technologies (CT) to effectively solve poorly formalized tasks in the field of analyzing large amounts of textual data. Special attention is paid to the task of automatic text annotation, which is one of the most important problems of modern science and technology, stimulating the active development of artificial intelligence and machine learning methods.

Methods. To achieve this goal, an extractive method of automatic summary compilation was used. This approach involves selecting the most significant fragments of the source text by highlighting individual sentences or phrases based on certain criteria. The following selection criteria were used: frequency of occurrence of words; semantic importance of words and expressions; position of sentences within the document. These indicators allow you to highlight the main thoughts of the text and create a compact summary that preserves the meaning of the source material.

Results. During the experiments, a program was developed in the Python programming language that implements the extracted method of taking notes. The algorithm is based on an analysis of the frequency of occurrence of keywords in the text, which ensures an effective assessment of the significance of each sentence. The program has a number of advantages: simplicity of implementation and operation; open source code, which makes it easy to adapt the solution to the specific needs of users; high efficiency in processing significant amounts of textual information. The developed tool demonstrates the ability to effectively create a summary of the text, while maintaining the basic meaning and structure of the original content.

Conclusion. The use of a cognitive algorithm has significantly improved the productivity of the text information analysis and processing process. The proposed methodology is capable of automating routine operations for making notes, helping specialists quickly get a general idea of the contents of large documents and publications. This is especially important in the context of the modern information society, characterized by ever-growing data flows that require rapid and high-quality comprehension. Thus, the study showed the prospects of introducing cognitive technologies into the field of automation of intellectual work and offered a practical solution to the urgent problem of taking notes on large amounts of information, which can become an important assistant for specialists and researchers.

Keywords: cognitive modeling; extractive annotation; frequency analysis; programming; natural language processing (NLP).

Conflict of interest. The Authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

For citation: Lisitsin L. A., A Lisitsin. L., Blokha D. N. A cognitive algorithm and a text annotation program. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2025; 29(3): 113-123. (In Russ.). <https://doi.org/10.21869/2223-1560-2025-29-3-113-123>.

Received 17.04.2025

Accepted 24.07.2025

Published 30.09.2025

Введение

Конспектирование текста – это процесс сокращения исходного текста с сохранением его основной смысловой нагрузки и ключевых идей. В эпоху информационного взрыва [1] умение быстро и точно извлекать важную информацию является критически важным навыком как для студентов, так и для профессионалов [2,3].

Актуальность статьи о конспектировании текста обусловлена несколькими факторами бурного роста информации, когда возникает необходимость в эффективных методах извлечения ключевых данных в условиях ограниченного времени. А также при выделении основных идей и ключевых моментов из текста, что способствует лучшему усвоению материала студентами и успешной подготовке к экзаменам.

Конспектирование текста представляет собой важную задачу, которая стимулирует развитие методов искусственного интеллекта.

Обзор программ и инструментов для автоматического конспектирования текста [4,5,6] включает в себя как коммерческие, так и открытые решения, которые используют различные методы обработки естественного языка (NLP) [7,8] и машинного обучения [9,10]. Ниже

представлены некоторые из наиболее известных и популярных программ и библиотек для конспектирования текста.

SMMRY – это бесплатный онлайн-инструмент, который позволяет пользователям вставлять текст или URL для получения краткого резюме. Использует простые алгоритмы извлекательного конспектирования, основанные на частоте слов и значимости предложений.

Преимущества – легкий в использовании, не требует регистрации, быстро обрабатывает текст.

Gensim [11,12] – это библиотека для обработки естественного языка на Python, которая включает в себя инструменты для извлекательного конспектирования. Использует алгоритм TextRank для извлечения ключевых предложений из текста.

Преимущества – открытый исходный код, поддерживает работу с большими объемами данных.

Сравнение с существующими программами и методами.

Метод извлекательного конспектирования, основанный на частотном анализе, не является уникальным. Рассмотрим основные аналоги.

TextRank. Алгоритм основан на графе, где предложения рассматриваются как вершины, а их связи – как ребра, с учетом частоты слов и их взаимосвязей. Преимущество TextRank – учитывает

структурные связи между предложениями, что может улучшить связность конспекта.

TF-IDF. Метод, использующий частоту слов с учетом их редкости в тексте. В статье метод ограничен частотным анализом внутри одного текста, что менее эффективно для выделения уникальных терминов по сравнению с TF-IDF.

LSA (Latent Semantic Analysis) и LDA (Latent Dirichlet Allocation). Эти методы анализируют семантическую структуру текста, что делает их более сложными, но потенциально более точными в выделении ключевых идей¹ [13].

SMMRY и SummarizeBot. Используют схожие извлекательные подходы [14], основанные на частоте слов и значимости предложений, но могут дополняться абстрактивными методами.

Для сравнения *систем* ужатия текста можно использовать несколько метрик, которые описаны ниже.

Пример таблицы с данными обработки одного размера текста для сравнения, показан в табл. 1.

Эти данные могут варьироваться в зависимости от конкретных условий использования и характеристик текстов, которые обрабатываются.

Для наглядного сравнения *методов* программного конспектирования текста проведем конспектирование текста одно-

го объема. В табл. 2 указаны данные с числовыми оценками по всем методам.

Предлагаемая программа и метод имеют свои достоинства, которые указаны в заключении.

Материалы и методы

Существуют два основных подхода к автоматическому конспектированию: извлекательный и абстрактивный.

Технологические решения для автоматического конспектирования включают классические алгоритмы обработки естественного языка, TF-IDF, методы на основе графов (TextRank), а также современные модели глубокого обучения (BERT, GPT и др.) [15, 16].

В работе предлагается программа на Python для простого извлекательного конспектирования. Данная программа использует частотный анализ слов для оценки и выбора ключевых предложений из текста.

Суть метода. Метод извлекательного конспектирования текста заключается в автоматическом выделении ключевых фраз из исходного текста с целью создания его краткой и информативной версии. Этот метод основывается на анализе содержания текста и использовании различных алгоритмов для оценки значимости его элементов.

Основные этапы метода

Разделение текста на предложения, что позволяет анализировать его структуру и содержание.

¹ Python. Сборник упражнений / пер. с англ. А. Ю. Гинько. М.: ДМК Пресс, 2021. 238 с.

Таблица 1. Работа с данными обработки одного размера текста**Table 1.** Working with single-size text processing data

Характеристика / Characteristics	Предлагаемая программа / Proposed program	Gensim	Sumy	TextRank
Коэффициент сжатия (%)	30%	25%	20%	25%
Точность извлечения (%)	75%	85%	80%	80%
Время обработки (сек)	2	1.5	1.8	2
Размер модели (МБ)	50	20	15	175

Таблица 2. Данные с числовыми оценками по всем методам программного конспектирования**Table 2.** Data with numerical estimates for all software monitoring methods

Метод / Method	Время обработки (сек) / Processing time (sec)	Коэффициент сжатия (%) / Compression ratio (%)	Точность извлечения (%) / Extraction accuracy (%)	Размер модели (МБ) / Model size (MB)
Извлекательное конспектирование	2	50%	80	0.5
Абстрактное конспектирование	5	100%	90	1.0
Суммирование на основе графов	3	80%	85	0.7
Методы на основе машинного обучения	4	150%	95	2.0
Семантическое конспектирование	5	120%	92	1.5
Методы на основе трансформеров	6	200%	98	3.0
Контекстуальное конспектирование	4	90%	88	1.2

Предобработка текста. Удаляются знаки препинания и приводятся слова к нижнему регистру. Это упрощает анализ и позволяет избежать дублирования при подсчете частоты слов.

Частотный анализ слов. Программа подсчитывает, как часто каждое слово встречается в тексте. Это позволяет определить, какие слова являются наиболее значимыми для понимания содержания

жания. Частотный анализ может включать исключение стоп-слов (не несущих смысловой нагрузки).

Оценка предложений. Каждое предложение получает оценку на основе частоты значимых слов. Чем больше значимых слов в предложении, тем выше его оценка.

Выбор ключевых предложений. На основе оценок выбираются несколько предложений с наивысшими значениями, которые будут включены в конспект.

Формирование итогового конспекта. Выбранные предложения объединяются в итоговый текст, который представляет собой сжатую версию исходного материала.

Преимущества и ограничения метода описаны в заключении.

Математическая модель метода извлекательного конспектирования [17].

Обозначения:

T – исходный текст, представленный как множество предложений

$S = \{s_1, s_2, \dots, s_n\}$, где n – количество предложений.

$W = \{w_1, w_2, \dots, w_m\}$ – множество всех слов в тексте после предобработки, где m – общее количество уникальных слов.

$F_{(wi)}$ – частота слова w_i в тексте (количество его появлений).

$StopWords$ – множество стоп-слов, исключаемых из анализа.

$Score(s_j)$ – оценка значимости предложения s_j .

k – желаемое количество предложений в конспекте (или процент от n).

C – итоговый конспект, состоящий из подмножества S .

Этапы метода

1. *Предобработка текста:*

Разделение текста T на предложения: $T \rightarrow S = \{s_1, s_2, \dots, s_n\}$.

Для каждого предложения s_j :

Удаление знаков препинания:

$s_j = RemovePunctuation(s_j)$.

Приведение к нижнему регистру:

$s_j = LowerCase(s_j)$.

Разбиение на слова:

$s_j \rightarrow \{w_{j1}, w_{j2}, \dots, w_{jl}\}$,

где l – количество слов в предложении.

2. *Частотный анализ слов:*

Для каждого слова $w_i \in W$:

$$F(w_i) = \sum_{j=1}^n \text{Count}(w_i, s_j),$$

где $\text{Count}(w_i, s_j)$ – количество появлений слова w_i в предложении s_j .

Исключение стоп-слов:

$W' = W \setminus StopWords$.

Оценка предложений:

Для каждого предложения s_j :

$Score(s_j) = {}^xF(w_i)$.

$w_i \in s_j, w_i \in W'$

Выбор ключевых предложений:

Сортировка предложений по убыванию $Score(s_j)$.

Выбор k предложений с наивысшими оценками: $C = \{s_{i1}, s_{i2}, \dots, s_{ik}\}$, где $k = \lfloor p \cdot n \rfloor$, а p – заданный процент сжатия.

Формирование конспекта:

Объединение выбранных предложений в порядке их появления в исходном тексте: $C = \text{Join}(s_{i1}, s_{i2}, \dots, s_{ik})$.

Формализация задачи

Цель метода – минимизировать объем текста при сохранении ключевой информации:

$$C = \arg \max_{C \subseteq S, |C|=k} \sum_{s_j \in C} Score(s_j).$$

Ограничение: $|C|=k$, где k определяется пользователем или как $k = \lfloor p \cdot n \rfloor$.

Метрики для оценки метода

Для оценки эффективности метода предлагаются следующие метрики:

1. Коэффициент сжатия (CR):

$$CR = \frac{|T| - |C|}{|T|} \cdot 100\%,$$

где $|T|$ – длина исходного текста (в символах или словах), $|C|$ – длина конспекта.

2. Точность извлечения (P, R, F1):

$$P = TP / (TP + FP),$$

где TP – истинно положительные предложения, FP – ложноположительные, FN – ложноотрицательные, P – точность (Precision) доли правильно извлеченных предложений среди всех извлеченных.

Время обработки (T_p): время, затраченное на обработку текста и создание конспекта (в секундах).

Качество связности: оценивается субъективно (экспертами) или с использованием косинусного сходства между векторами предложений.

Размер модели (M_s): Размер программы в мегабайтах.

Когнитивный алгоритм работы программы [18]:

1. Ввод информации.

2. Предварительная обработка текста: разбивка текст на отдельные пред-

ложения (это позволяет анализировать его структуру и содержание) и удаление знаков препинания с преобразованием слова к нижнему регистру (чтобы избежать дублирования при анализе).

3. Анализ содержания: подсчёт, как часто каждое слово встречается в тексте для определения наиболее значимых слов в тексте и исключение распространенных слов (стоп-слов), которые не несут смысловой нагрузки (предлоги, союзы).

4. Оценка предложений: для каждого предложения вычисляется его "оценка" на основе частоты встречаемых в нем значимых слов (чем больше слов, тем выше оценка) и сравниваются оценки всех предложений (чтобы определить, какие из них наиболее информативны).

5. Выбор ключевых предложений: выбирается несколько предложений с наивысшими оценками для сохранения основных идей текста.

6. Формирование итогового конспекта: выбранные предложения объединяются в итоговый текст, который представляет собой конспект. Дополнительно проверяется, насколько логично составлен итоговый текст.

7. Программа выводит итоговый конспект.

Результаты и их обсуждение

Метод не является уникальным, частотный анализ слов – это стандартный подход в извлекательном конспектировании, используемый в таких инструментах, как SMMRY и Gensim. Однако его ценность заключается в простоте, от-

крытом исходном коде и возможности локальной работы. Для повышения уникальности можно интегрировать элементы TextRank или TFIDF, чтобы улучшить учет контекста и связности.

Математическая модель формализована с учетом этапов предобработки, частотного анализа, оценки предложений и выбора ключевых предложений.

Метрики предложены коэффициент сжатия, точность извлечения (F1), ROUGE, время обработки, качество связности и размер модели, что позволяет всесторонне оценить эффективность метода.

Выводы

Использование когнитивных методов позволяет достичь нужных результатов при конспектировании текстов, а именно времени и вычислительной мощности компьютера [19].

Реализованный алгоритм с разработанной программой обладает следующими достоинствами:

1. Простота использования благодаря простому и понятному интерфейсу.

2. Эффективность – благодаря частотному анализу слов для оценки значимости предложений быстро и точно выделяются ключевые идеи из текста.

3. Легкость в настройке.

4. Открытый исходный код для модификации и улучшения.

5. Отсутствие зависимости от внешних API - программа работает локально, что исключает необходимость в интернет-соединении и повышает безопасность данных.

6. Метод не является уникальным, но для улучшения можно интегрировать в него элементы TextRank или TFIDF, для повышения учета контекста и связности.

Недостатки программы

1. Ограниченнная точность (в сложных текстах).

2. Отсутствие абстрактивного конспектирования.

3. Зависимость от качества исходного текста: эффективность программы может снижаться при работе с плохо структурированными текстами.

Список литературы

1. Строверова Н. А., Шустрова М. Л. Искусственный интеллект в профессиональной сфере. Казань: ООО "Ред.-издательский центр "Школа", 2022. Ч. 1. 84 с. EDN PTSPTI.
2. Бова В.В., Кравченко Ю.А., Родзин С.И. Методы и алгоритмы кластеризации текстовых данных (ОБЗОР) // Известия ЮФУ. Технические науки. 2022. № 4 (228). С. 122-143.
3. Григорян Э.Г., Паршин М.Н. Методы NLP для предобработки текстовых данных и выделения признаков // Бизнес и общество. 2021. № 3 (31).
4. Тюрина Д.А., Пальмов С.В. Применение нейронных сетей в обработке естественного языка // Журнал прикладных исследований. 2023. № 7. С. 158-162.

5. Irada Seyidova, Osman Şərifli I.S.O.Ş., Murad Abbasov M.A. Реализация методов искусственного интеллекта в обработке естественного языка с использованием технологий больших данных // Equipment, Technologies, Materials. 2024. Т. 23, № 5. С. 104-113.
6. Кравченко Д.Ю. Модель онтологии знаний для интеллектуальных систем обработки и анализа текстов // Известия ЮФУ. Технические науки. 2024. № 2 (238). С. 38-50.
7. Захарова М.В., Интеллектуальные помощники для научного исследования в университетах // Мир науки. Педагогика и психология. 2024. Т. 12. № 4.
8. Ковтан А.И., Якубович Д.А., Использование систем искусственного интеллекта в обработке текстовой информации // Сборник материалов научно-практических конференций. Владимир, 2024. С. 1785-1796.
9. Лисицин Л. А., Халин Ю. А., Лисицин А. Л. Системы поддержки принятия управлеченческих решений в условиях неполной информации // Известия Юго-Западного государственного университета. 2012. № 4-2 (43). С. 95-99.
10. Sharma M., Schuermyer C., Benware B. Determination of Dominant-Yield-Loss Mechanism with Volume Diagnosis // Proc. of IEEE Design & Test of Computers. 2010. Vol.27, no.3. P.54-61.
11. Сурова Н. Ю., Косов М. Е. Искусственный интеллект. М.: Издательство "Юни-ти-Дана", 2021. 408 с. EDN QZEMDW.
12. Остроух А. В. Введение в искусственный интеллект. Красноярск: Научно-инновационный центр, 2020. 250 с. DOI 10.12731/978-5-907208-26-1. EDN KTJGZM.
13. Когнитивное моделирование информационного обеспечения игрового автоматизированного обучения / Халин Ю.А., Катыхин А.И., Зинкин С.А., Шилин А.А. // Известия Юго-Западного государственного университета. 2022. Т. 26, № 4. С. 117-131. <https://doi.org/10.21869/2223-1560-2022-26-4-117-131>
14. Исследование методов обработки текстовой информации и обзор этапов создания модели искусственного интеллекта при создании чат-ботов / А.В. Иванова, А.А. Кузьменко, Р.А. Филиппов [и др.] // Автоматизация и моделирование в проектировании и управлении. 2021. № 2 (12). С. 19-23.
15. Наместников А. М. Онтологический подход к структурированию знаний проектной организации // Радиотехника. 2016. N 9. С. 77-83.
16. Цифровая трансформация и искусственный интеллект в разработке биологически активных веществ и лекарственных форм / Л. В. Погребняк, Е. А. Кульгав, Е. В. Ковтун, А. В. Погребняк. М.: Мир науки, 2022. 201 с. EDN LWVFJC.
17. Lal M. Neo4j Graph Data Modeling. Packt Publishing, 2015. 119 p.
18. Никитин Ю.В., Хорошилов А.А. Интеллектуальный текстовый процессор // Искусственный интеллект. Теория и практика. 2023. № 1 (1). С. 56-75.
19. Пань В., Цзы Е. Роль искусственного интеллекта (AGI) в цифровой журналистике: от производства новостей до взаимодействия со зрителем // Успехи гуманитарных наук. 2024. № 8. С. 36-40.

References

1. Staroverova N. A., Shustrova M. L. Artificial intelligence in the professional sphere. Kazan: Red.-izdatel'skii tsentr "Shkola"; 2022. Part 1. 84 p. (In Russ.). EDN PTSPTI.
2. Bova V.V., Kravchenko Yu.A., Rodzin S.I. Methods and algorithms for clustering text data (review). *Izvestiya YuFU. Tekhnicheskie nauki = Izvestiya SFU. Technical sciences.* 2022; (4): 122-143. (In Russ.).
3. Grigoryan E.G., Parshin M.N. NLP methods for text data preprocessing and feature extraction. *Biznes i obshchestvo = Business and Society.* 2021; (3). (In Russ.).
4. Tyurina D.A., Palmov S.V., Application of neural networks in natural language processing. *Zhurnal prikladnykh issledovanii = Journal of Applied Research.* 2023; (7): 158-162. (In Russ.).
5. Irada Seyidova, Osman Şərifli I.S.O.Ş., Murad Abbasov M.A., Implementation of artificial intelligence methods in natural language processing using big data technologies. *Equipment, Technologies, Materials.* 2024; 23(5): 104-113. (In Russ.).
6. Kravchenko D.Y., Knowledge ontology model for intelligent text processing and analysis systems. *Izvestiya YuFU. Tekhnicheskie nauki = Izvestiya SFU. Technical sciences.* 2024; (2): 38-50. (In Russ.).
7. Zakharova M.V. Intellectual assistants for scientific research at universities. *Mir nauki. Pedagogika i psichologiya = World of science. Pedagogy and psychology.* 2024; 12(4). (In Russ.).
8. Kovtan A.I., Yakubovich D.A., The use of artificial intelligence systems in text information processing. In: *Sbornik materialov nauchno-prakticheskikh konferentsii = Collection of materials of scientific and practical conferences.* Vladimir; 2024. P. 1785-1796. (In Russ.).
9. Lisitsin L. A., Khalin Yu. A., Lisitsin A. L. Management decision support systems in conditions of incomplete information. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University.* 2012; (4-2): 95-99. (In Russ.).
10. Sharma M., Schuermyer C., Benware B., Determination of Dominant-Yield-Loss Mechanism with Volume Diagnosis. *Proc. of IEEE Design & Test of Computers.* 2010; 27(3): 54-61. (In Russ.).
11. Surova N. Y., Kosov M. E. Artificial intelligence. Moscow: Unity-Dana Publishing House; 2021. 408 p. (In Russ.). EDN QZEMDW.
12. Ostroukh A.V. Introduction to artificial intelligence. Krasnoyarsk: Nauchno-innovatsionnyi tsentr; 2020. 250 p. (In Russ.). DOI 10.12731/978-5-907208-26-1. EDN KTJGZM.
13. Khalin Y. A., Katykhin A. I., Zinkin S. A., Shilin A. A. Cognitive Modeling of Information Support for Game-Based Automated Learning. *Izvestiya Yugo-Zapadnogo gosu-*

darstvennogo universiteta = Proceedings of the Southwest State University. 2022; 26(4): 117-131 (In Russ.). <https://doi.org/10.21869/2223-1560-2022-26-4-117-131>

14. Ivanova A.V., Kuzmenko A.A., Filippov R.A., et al. Research of text information processing methods and an overview of the stages of creating an artificial intelligence model when creating chatbots. *Avtomatizatsiya i modelirovaniye v proektirovaniye i upravlenii = Automation and modeling in design and management.* 2021; (2): 19-23. (In Russ.).

15. Namestnikov A.M. An ontological approach to structuring knowledge of a project organization. *Radiotekhnika = Radio Engineering.* 2016; (9): 77-83. (In Russ.).

16. Pogrebnyak L. V., Kulgav E. A., Kovtun E. V., Pogrebnyak A.V. Digital transformation and artificial intelligence in the development of biologically active substances and dosage forms. Moscow: Mir nauki; 2022. 201 p. (In Russ.). EDN LWVFJC.

17. Lal M. Neo4j Graph Data Modeling. Packt Publishing, 2015. 119 p.

18. Nikitin Yu.V., Khoroshilov A.A. Intelligent word processor. *Iskusstvennyi intellekt. Teoriya i praktika = Artificial intelligence. Theory and practice.* 2023; (1): 56-75. (In Russ.).

19. Pan V., Zi E., The role of artificial intelligence (AGI) in digital journalism: from news production to interaction with the viewer. *Uspekhi gumanitarnykh nauk = Successes of the Humanities.* 2024; (8): 36-40 (In Russ.).

Информация об авторах / Information about the Authors

Лисицин Леонид Александрович, кандидат технических наук, доцент, доцент кафедры программной инженерии, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: leo_263@mail.ru

Лисицин Александр Леонидович, старший преподаватель кафедры информационной безопасности, Курский государственный университет, г. Курск, Российская Федерация, e-mail: vip.lisicin@mail.ru

Блоха Дмитрий Николаевич, студент кафедры информационной безопасности, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: alex25lisitsyn@gmail.com

Leonid A. Lisitsin, Cand. of Sci. (Engineering), Associate Professor, Associate Professor of the Software Engineering Department, Southwest State University, Kursk, Russian Federation, e-mail: leo_263@mail.ru

Alexander L. Lisitsin, Senior Lecturer, Information Security Department, Kursk State University, Kursk, Russian Federation, e-mail: vip.lisicin@mail.ru

Dmitry N. Blokh, Student of the Information Security Department, Southwest State University, Kursk, Russian Federation, e-mail: alex25lisitsyn@gmail.com