



<http://dx.doi.org/10.35596/1729-7648-2025-23-3-54-61>

УДК 681.3

МЕТОД АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ ВОПРОСОВ И ОТВЕТОВ ДЛЯ СИСТЕМ ТЕСТИРОВАНИЯ ЗНАНИЙ

С. А. МИГАЛЕВИЧ, Ю. О. ГЕРМАН, О. В. ГЕРМАН

*Белорусский государственный университет информатики и радиоэлектроники
(Минск, Республика Беларусь)*

Аннотация. Рассмотрена техника автоматической генерации вопросов и ответов для тестирования знаний. Предложенный подход построен на разработанном методе определения и использования кластеров ключевых слов для генерации по ним вопросов и ответов на базе глобальной или локальной языковой модели, что определяет теоретико-прикладную новизну. Кластеры строятся на основе корпуса текстовых документов, относящихся к изучаемой области (учебная литература, методические пособия, электронные ресурсы). Представленное в статье техническое решение является логически законченным и может служить основой для практических разработок.

Ключевые слова: информационные технологии, обучение, тестирование знаний, кластерный анализ.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Мигалевич, С. А. Метод автоматической генерации вопросов и ответов для систем тестирования знаний / С. А. Мигалевич, Ю. О. Герман, О. В. Герман // Доклады БГУИР. 2025. Т. 23, № 3. С. 54–61. <http://dx.doi.org/10.35596/1729-7648-2025-23-3-54-61>.

METHOD OF AUTOMATIC GENERATION OF QUESTIONS AND ANSWERS FOR KNOWLEDGE TESTING SYSTEMS

SERGEI A. MIGALEVICH, JULIA O. GERMAN. OLEG V. GERMAN

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Abstract. The technique of automatic generation of questions and answers for knowledge testing is considered. The proposed approach is based on the developed method of defining and using clusters of keywords to generate questions and answers based on a global or local language model, which constitutes theoretical and applied novelty. Clusters are built on the basis of a corpus of text documents related to the studied area (textbooks, methodological manuals, electronic resources). The technical solution presented in the article is logically complete and can serve as a basis for practical developments.

Keywords: information technologies, education, knowledge testing, cluster analysis.

Conflict of interest. The authors declare no conflict of interests.

For citation. Migalevich S. A., German Ju. O., German O. V. (2025) Method of Automatic Generation of Questions and Answers for Knowledge Testing Systems // *Doklady BGUIR*. 23 (3), 54–61. <http://dx.doi.org/10.35596/1729-7648-2025-23-3-54-61> (in Russian).

Введение

Известные подходы к решению задачи генерации вопросов в автоматизированной системе тестирования знаний обучаемых [1, 2] используют генеративные нейросети, интеллектуальные интерактивные текстовые аналитические системы типа глобальной языковой модели ChatGPT, системы генерации вопросов на основе подготовленных шаблонов (templates), динамический отбор вопросов из базы данных аналогичных систем, механизмы грамматического разбора и др.

Проблемным местом является генерация вопросов к конкретному текстовому материалу, например электронному ресурсу, учебнику, конспекту лекций, авторской монографии, книге и т. п. Вопросы генерируют либо к конкретным предложениям, параграфам, выдержкам, темам, либо ко всему документу.

В статье рассмотрена новая техника генерации вопросов в автоматизированной системе тестирования знаний обучаемых, а именно – генерация вопросов ко всему документу. Подход основан на выделении ключевых слов в документе и разбиении их на кластеры, причем допускается пересечение кластеров. Изложены общий подход и метод построения кластеров ключевых слов, используемых для формирования вопросов и получения ответов на них, приемлемые с точки зрения вычислительной сложности для практических приложений. Для каждого кластера можно построить несколько вопросов, используя локальную (кратко рассмотренную в статье) или глобальную языковую модель типа ChatGPT, и получить ответы на базе этой модели. Глобальные языковые модели представляют мощнейший семантический аппарат, но не свободный от недостатков. Так, на вопрос к ChatGPT: «Объясни способ построения матрицы покрытия для системы дизъюнктов задачи выполнимости по книге О. В. Германа. Введение в теорию экспертных систем и обработку знаний» [3] система выдала неполный (по существу, неверный) ответ. Проиллюстрируем это на следующем примере. Предположим, что имеются дизъюнкты:

$$\begin{aligned} D1: & A \vee \neg B \\ D2: & \neg A \vee C \\ D3: & B \vee \neg C \\ D4: & \neg A \vee \neg B \vee C. \end{aligned}$$

Матрица покрытия по ChatGPT выглядит согласно рис. 1.

Литералы	A	B	C	$\neg A$	$\neg B$	$\neg C$
D1	1	1	0	0	1	0
D2	1	0	1	1	0	0
D3	0	1	0	0	0	1
D4	1	1	1	1	1	0

Рис. 1. Матрица покрытия по ChatGPT
Fig. 1. ChatGPT Coverage Matrix

Правильный ответ с учетом данных, приведенных в [3, с. 108], должен быть таким, как изображено на рис. 2.

Литералы	A	B	C	$\neg A$	$\neg B$	$\neg C$
D1	1	1	0	0	1	0
D2	1	0	1	1	0	0
D3	0	1	0	0	0	1
D4	1	1	1	1	1	0
D5	1	0	0	1	0	0
D6	0	1	0	0	1	0
D7	0	0	1	0	0	1

Рис. 2. Матрица покрытия по [3]
Fig. 2. Coverage matrix according to [3]

Дополнительные строки D5–D7 соответствуют тавтологическим дизъюнктам, добавление которых определяет размер минимального покрытия при выполнимости (невыполнимости) системы дизъюнктов, что необходимо для исключения одновременного появления контрольной пары литер в минимальном покрытии. Нарушение этого требования может давать неверный ответ.

Использование кластеров ключевых слов для генерации вопросов

Ключевые слова помогают сравнительно просто выделить основные идеи и темы, содержащиеся в тексте, и составляют основу многих хорошо зарекомендовавших себя методов. Выделе-

ние ключевых слов – это задача экспертов, либо ее можно решить с помощью средств TextMining, например, реализованных в Python NLTK [4] (т. е. быть полностью автоматизированной). Каждый тестовый вопрос строится на основе нескольких (например, 3–5) выбранных взаимосвязанных ключевых слов. Взаимная связь данных (в том числе ключевых слов) передается обычно через коэффициенты корреляции/ковариации (другие техники могут базироваться на методе типа ANOVA [5], на регрессионном анализе [6] и т. д.).

Для решения задачи отбора ключевых слов в список ключевых слов вопроса строили матрицу информационно-смысловой близости слов (МИСБС). Теоретически числа в ячейках МИСБС могут представлять коэффициенты корреляции между словами, которые можно получить по некоторому корпусу текстов по проблематике экзамена (дисциплины, теста). Это обеспечило бы объективность оценки информационно-смысловой связности слов. Вместе с тем есть проблема, когда коэффициенты корреляции сравнительно одинаковы по всей матрице корреляции, в силу чего определение кластерной структуры сопряжено с известной неопределенностью. Поэтому автоматический расчет коэффициентов корреляции можно заменить заполнением МИСБС экспертом с использованием оценочной шкалы Харрингтона [7]. Шкала Харрингтона – это дискретная оценочная шкала, состоящая из пяти интервалов единичного отрезка со следующими оценочными интервалами для степени близости слов: очень высокая (0,8–1,0), высокая (0,63–0,8), средняя (0,37–0,63), низкая (0,37–0). Имея такую оценочную МИСБС, можно построить кластеры с ограничением на число слов в каждом кластере и число кластеров. Кроме того, с помощью оценочной МИСБС можно построить кластеры с ограничением на число слов в каждом из них. Задача имеет две особенности:

M1: размер кластера ограничен тремя-пятью ключевыми словами, при этом каждый кластер соответствует одному или более тестовому вопросу, который будет сгенерирован для этого кластера;

M2: разные кластеры могут содержать общие ключевые слова (но не совпадать).

Достаточно полное описание известных методов кластеризации дано в [8, с. 20], где кластеры объектов формируются таким образом, чтобы объекты внутри кластера имели высокое сходство и были похожи друг на друга, но весьма не похожи на объекты в других кластерах. В рассматриваемом в статье случае это определение требует существенного уточнения, связанного с более общей трактовкой кластера:

1) «сходство» объектов (ключевых слов) внутри кластера определяется высокими значениями коэффициентов парной корреляции;

2) допустимо, что объекты, попадающие в разные кластеры, также могут иметь высокие коэффициенты попарной корреляции в силу замечания M2.

Важно, чтобы в кластер входили связанные ключевые слова, и при этом одно и то же ключевое слово может попасть более чем в один кластер. Под кластером в данной статье понимается общепринятое понятие объединения однородных элементов, связанных единством темы, рассматриваемых как самостоятельная единица.

Теоретически многие задачи разбиения объектов на кластеры относятся к категории NP-трудных [3, 8]. Задача является эффективно решаемой, поскольку число всех кластеров, содержащих $t = 3-5$ ключевых слов в множестве с $N \geq t$ слов, определяется формулой для числа сочетаний из N по t : $N(N-1) \cdot \dots \cdot (N-t+1)/t!$ Для фиксированного t это полином $O(N^t)$. Однако при больших N перебор приводит к существенным потерям времени, а при $t \rightarrow N/2$ временные затраты становятся экспоненциально сложными.

В связи с этим предлагается приближенный алгоритм отыскания кластеров ключевых слов с практически подтвержденной эффективностью, учитывающий M2. Идея алгоритма основана на последовательном исключении из МИСБС слов с минимальной (суммарной/средней) связностью с остальными словами. Подобный подход аналогичен использованному в [9] методу для отыскания максимальной по размеру нулевой подматрицы 0,1-матрицы, т. е. в описанной реализации нулевые элементы соответствовали бы сильно коррелированным парам ключевых слов, а единичные – некоррелированным или слабо коррелированным, причем 0,1-матрица связей получается из МИСБС заменой каждого числа в ячейке на 0, если это число больше среднего по МИСБС, и на 1 – если не больше.

Алгоритм построения кластеров

В качестве исходных данных задаются множество размером N всех ключевых слов и МИСБС размером $N \times N$. На выходе получаются кластеры Cl_1, Cl_2, \dots, Cl_z , где z определяется устанавливаемым пользователем числом генерируемых вопросов по теме документа. Реальное число может оказаться меньше, поскольку одинаковые кластеры k исключены. Построение каждого очередного кластера начинается с выбора его центроида a_i . Для слова «центроид a_i » выполняется следующий алгоритм:

шаг 1: удаляются строка и столбец, соответствующие выбранному слову из МИСБС;

шаг 2: подсчитываются суммы элементов в невычеркнутых строках. Выбирается строка (и одноименный столбец) для вычеркивания с вероятностью, обратно пропорциональной сумме элементов в строке (столбце);

шаг 3: если число оставшихся невычеркнутыми строк (столбцов) больше заданного числа слов в кластере, то происходит возврат на шаг 1. Иначе оставшиеся невычеркнутыми строки МИСБС определяют ключевые слова, образующие найденный кластер. Построение очередного кластера завершено.

Рассмотрим пример по теме объектно-ориентированного программирования. Пусть некоторый вариант МИСБС имеет вид, представленный в табл. 1, где a_1 – объект, a_2 – класс, a_3 – метод, a_4 – тип, a_5 – оператор, a_6 – параметр.

Таблица 1. Пример матрицы информационно-смысловой близости слов
Table 1. An example of a matrix of information-semantic proximity of words

Слова	a_1	a_2	a_3	a_4	a_5	a_6
a_1	1	0,8	0,3	0,4	0,7	0,3
a_2	0,8	1	0,6	0	0,5	0,2
a_3	0,3	0,6	1	0,1	0,6	0,6
a_4	0,4	0	0,1	1	0,8	0
a_5	0,7	0,5	0,6	0,8	1	0,7
a_6	0,3	0,2	0,6	0	0,7	1

После первого шага выполняем редукцию матрицы информационно-смысловой близости слов (табл. 2), заключающуюся во включении в кластер строки a_1 (в качестве центроида) и в вычеркивании ее (и столбца a_1) из матрицы.

Таблица 2. Редукция матрицы информационно-смысловой близости слов после первого шага
Table 2. Reduction of the matrix of information-semantic proximity of words after the first step

Слова	a_2	a_3	a_4	a_5	a_6
a_2	1	0,6	0	0,5	0,2
a_3	0,6	1	0,1	0,6	0,6
a_4	0	0,1	1	0,8	0
a_5	0,5	0,6	0,8	1	0,7
a_6	0,2	0,6	0	0,7	1

Подсчитаем суммы элементов в строках и получим следующие значения: строка a_2 – 2,3, a_3 – 2,9, a_4 – 1,9, a_5 – 3,6, a_6 – 2,5. Определим вероятности для строк, обратно пропорциональные этим суммам с помощью Python. Затем случайно выбираем строку по этим вероятностям. Скриншот выбора элемента для удаления по вероятности приведен на рис. 3.

```

Administrator: C:\Windows\System32\cmd.exe
Elements: [2.3, 2.9, 1.9, 3.6, 2.5]
Probabilities: ['0.22', '0.17', '0.27', '0.14', '0.20']
Sum of probabilities: 1.00
Chosen element: 1.9
Index of chosen item: 2

c:\Users\asa\AppData\Local\Programs\Python\Python312>
    
```

Рис. 3. Выбор элемента для удаления по вероятности
Fig. 3. Selecting an element to remove by probability

На втором шаге удаляем выбранный элемент 1,9 (a_4) из списка, пересчитываем суммы оставшихся строк в матрице и рассчитываем вероятности (табл. 3).

Таблица 3. Редукция матрицы информационно-смысловой близости слов после второго шага
Table 3. Reduction of the matrix of information-semantic proximity of words after the second step

Слова	a_2	a_3	a_5	a_6
a_2	1	0,6	0,5	0,2
a_3	0,6	1	0,6	0,6
a_5	0,5	0,6	1	0,7
a_6	0,2	0,6	0,7	1

Действуя по аналогии далее, получаем в итоге первый кластер из неудаленных элементов (a_1, a_3, a_5) с центроидом a_1 . Поскольку элементы кластера выбирались по вероятностям, очередной прогон процедуры может дать другой кластер с тем же центроидом a_1 . Поставим задачу получить 2–4 различных кластера с центроидом a_1 . Для того чтобы обеспечить неповторение кластеров с данным центроидом, после формирования первого кластера удалим из МИСБС второй выбранный элемент в кластере (a_1, a_3, a_5), т. е. a_3 . Повторим процедуру без a_3 . Так, при повторном прогоне получили кластер (a_1, a_4, a_2). Снова удалим второй элемент, т. е. a_4 , и отыщем очередной кластер – (a_1, a_2, a_5). Ограничиваемся формированием 2–4 кластеров. Каждый кластер используется для генерации связанного с ним тестового вопроса. Кластер из ключевых слов, сформированный за один прогон программы, показан на рис. 4.

```

Administrator: C:\Windows\System32\cmd.exe
Remaining matrix:
[[1. 0.6 0.5]
 [0.6 1. 0.6]
 [0.5 0.6 1. ]]
Indices of remaining rows: [0, 1, 3]
Words for remaining rows: ['object', 'class', 'type']
c:\Users\asa\AppData\Local\Programs\Python\Python312>
    
```

Рис. 4. Кластер из ключевых слов, сформированный за один прогон программы
Fig. 4. A cluster of keywords formed in one run of the program

При повторном прогоне программы определен новый кластер, включающий слова object, type, operator.

Оценка вычислительной сложности и определение числа ключевых слов в тексте

Для оценки вычислительной сложности описанного метода использовали программу на языке Python. В экспериментах отыскивали кластеры из 3–5 ключевых слов, варьируя общее число ключевых слов N в тексте. Фиксировали время, затрачиваемое компьютером на выполнение программы. Результаты приведены в табл. 4.

Таблица 4. Оценка вычислительной сложности для k , равного 3, 4, 5
Table 4. Estimated computational complexity for k equal to 3, 4, 5

N	Время, с, для		
	$k = 3$	$k = 4$	$k = 5$
10	0,24	0,31	0,31
20	0,62	0,62	0,62
50	2,15	2,19	2,17
200	16,7	17,3	17,4

Выполняли эксперименты для оценки вычислительной сложности порождения различных кластеров, содержащих $k = 3–5$ ключевых слова из N (метод простого перебора), с последующим отбором четырех кластеров с максимальной суммой парных коэффициентов корреляции. Результаты представлены в табл. 5.

Таблица 5. Оценка вычислительной сложности переборного метода для k , равного 3, 4, 5
Table 5. Estimation of computational complexity of the enumeration method for k equal to 3, 4, 5

N	Время, с, для		
	$k = 3$	$k = 4$	$k = 5$
10	0,15	0,15	0,31
20	0,50	1,37	3,7
50	5,8	79,9	1143,7
200	1238,2	Остановлено после 30 мин счета	

Приведенные в табл. 4, 5 экспериментальные данные определяют преимущество предложенного подхода. Задачу определения ключевых слов можно связать с задачей о минимальном покрытии 0,1-матрицы множеством строк. Воспользуемся оценкой [10], согласно которой число строк μ в покрытии π 0,1-матрицы \mathbf{B} с R строками и C столбцами, которое отыскивается «жадным» алгоритмом, оценивается как:

$$\mu \approx 1 - [\ln(Cpq^{0,5})]/\ln q, \quad (1)$$

где p – плотность единичных элементов в матрице \mathbf{B} ; q – плотность нулевых элементов в матрице \mathbf{B} , $q = 1 - p$.

Плотность p вычисляется по формуле

$$p = E/RC, \quad (2)$$

где E – число «1» в матрице \mathbf{B} .

Покрытие π состоит из строк матрицы \mathbf{B} , причем хотя бы одна строка из π содержит «1» в каждом столбце матрицы \mathbf{B} . «Жадный» алгоритм на каждом шаге включает в покрытие строку с максимальным числом «1» и удаляет покрываемые ею столбцы из \mathbf{B} , пока таковые есть.

Воспользуемся следующими оценками. Среднее число слов в предложении английского текста составляет 15–20 (данные ChatGPT). Текст из 10 000 слов будет включать порядка 500 предложений. Каждое предложение соответствует столбцу 0,1-матрицы \mathbf{B} ($C = 500$), каждое слово – строке матрицы \mathbf{B} . В среднем в англоязычном тексте употребляется 1000–3000 различных слов w_k . Возьмем половину этого диапазона – 1500 ($R = 1500$). Итак, при таких «среднепотребительных» данных число единиц в матрице \mathbf{B} составит $E = 10\,000$ (по общему числу слов в тексте). Единица в строке i и столбце j означает, что слово w_i вошло в предложение (столбец) C_j . Размер матрицы \mathbf{B} составит $CR = 500 \cdot 1500 = 750\,000$. Отсюда плотность единиц $p = 10\,000/750\,000 = 1/75$. Соответственно, плотность «0» $q = 1 - p = 74/75$. Из (1) вычислим, что $\mu \approx 141$. Таким образом, при $\mu \approx 141$ каждое предложение будет содержать как минимум одно ключевое слово. Увеличив число ключевых слов, мы, очевидно, повысим качество поиска ответов. Для практических целей можно использовать приближенную оценку числа ключевых слов, выбираемых из текста с общим числом слов Z как $\text{const}Z^{0,5}$, где коэффициент const для текстов в несколько десятков тысяч слов близок к 2–3.

Генерация тестовых вопросов и получение ответов на них

Используя полученные ключевые слова, мы обратились к ChatGPT для генерации вопросов с этими словами и ответов на них. Получили следующие варианты вопросов, выданные ChatGPT:

вопрос a: How do objects, types, and classes relate? (Как соотносятся объекты, типы и классы?). (Explain the relationship between objects, types, and classes in object-oriented programming. How does a class define the type of an object?);

вопрос b: Can you have multiple objects of the same type? (Можете ли вы располагать множеством объектов одного и того же типа?). (Discuss how multiple objects can be created from a single class and how they share the same type. What implications does this have for memory usage and data management?).

На приведенные вопросы в ChatGPT получены следующие ответы:

ответ на вопрос a: In essence, classes define types, and objects are instances of these types, allowing for organized and modular programming (По сути, классы определяют типы, а объекты являются экземплярами этих типов, обеспечивая организованное модульное программирование);

ответ на вопрос b: Yes, you can have multiple objects of the same type in object-oriented programming (У вас может быть несколько объектов одного и того же типа в объектно-ориентированном программировании).

Глобальная языковая модель типа ChatGPT вполне приемлема для генерации вопросов к общим темам программирования, физики, истории и т. п. Но и в этом случае определение ключевых слов для конкретного учебного документа отражает его специфику (например, ограниченное по затрагиваемым темам содержание, для которого нужно сформулировать тестовые вопросы). Для узкоспециализированной тематики более точна локальная модель.

В качестве примера использования локальной языковой модели можно взять [11]. Работа такой модели основана на определении позиций ключевых слов вопроса в оригинальном тексте и локализации смысловых блоков (обычно параграфов) с этими ключевыми словами. Неключевые слова вопроса используются для уточнения оценки степени соответствия каждого из найденных смысловых блоков тексту вопроса. Формулировки вопросов могут строиться на базе общих шаблонов типа «Укажите связь между k_1, k_2, \dots, k_z » (где k_1, k_2, \dots, k_z – ключевые слова); «Дайте определение k_1 », «В каком контексте используют k_1, k_2, \dots, k_z » и др. Более интересен и практически целесообразен способ генерации вопросов непосредственно к смысловым блокам, предварительно определяемым по ключевым словам. При этом можно обратиться к ChatGPT для генерации примеров вопросов к данному текстовому блоку и воспользоваться этим блоком в качестве ответа. Кроме того, ChatGPT дает примеры неправильных ответов, что позволяет использовать тесты для контроля знаний.

Заключение

1. Реализованный подход может быть в значительной степени автоматизирован, допуская участие эксперта для оценки качества и содержательности построенных тестов и ответов на них.
2. Предложен метод определения кластеров ключевых слов для формирования на их основе тестовых вопросов и получения ответов из используемых языковых моделей.
3. Приведены теоретические оценки для числа ключевых слов в тексте и получены результаты практической апробации, свидетельствующие об эффективности всего подхода.

Список литературы

1. A Review on Question Generation from Natural Language Text / Ch. Ruqing [et al.] // ACM Trans. on Information Systems. 2021. Vol. 40, No 1.
2. Brusilowsky, P. Web-Based Testing for Distance Education / P. Brusilowsky, P. Miller // WebNet'99, World Conference of the WWW and Internet, AACE, Honolulu, 1999. P. 149–154.
3. Герман, О. В. Введение в теорию экспертных систем и обработку знаний / О. В. Герман // Минск: Ди-зайнПро, 1995.
4. Bird, S. Natural Language Processing with Python / S. Bird, E. Klein, E. Loper // O'Reilly Media Inc. USA, 2007.
5. Montgomery, D. C. Design and Analysis of Experiments / D. C. Montgomery // John Wiley&Sons Inc. USA, 2013.
6. Neter, J. Applied Linear Regression Models / J. Neter, W. Wasserman, M. H. Kutner // R. D. Irwin Inc. USA, 1983.
7. Blei, D. M. Latent Dirichlet Allocation / D. M. Blei, A. Y. Ng, M. I. Jordan // Journal of Machine Learning Research. 2003. No 3. P. 993–1022.
8. Han, J. Data Mining. Concepts and Techniques / J. Han, M. Kamber, J. Pei // Elsevier. 2012.
9. Герман, О. В. Алгоритм булевой оптимизации на (0,1)-матрицах / О. В. Герман, Е. И. Германович, В. Г. Найдено // Журнал вычислительной математики и математической физики. 1992. Т. 32, № 7. С. 1114–1125.
10. Герман, О. В. Разрешающий принцип для задачи о минимальном покрытии 0,1-матрицы / О. В. Герман // Кибернетика и системный анализ. 1996. № 1. С. 135–145.
11. Герман, Ю. О. Метод извлечения информации из резюме / Ю. О. Герман, О. В. Герман, С. Наср // Труды БГТУ. Серия 3. Физико-математические науки и информатика. 2019. № 1. С. 64–68.

References

1. Ruqing Ch., Jiafeng G., Lu Ch., Xueqi Cheng., Yixing Fan. (2021) A Review on Question Generation from Natural Language Text. *ACM Trans. on Information Systems*. 40 (1).
2. Brusilowsky P., Miller P. (1999) Web-Based Testing for Distance Education. *WebNet '99, World Conference of the WWW and Internet, AACE, Honolulu*. 149–154.
3. German O. V. (1995) *Introduction to Expert Systems Theory and Knowledge Processing*. Minsk, DesignPro Publ. (in Russian).
4. Bird S., Klein E., Loper E. (2007) *Natural Language Processing with Python*. O'Reilly Media Inc. USA.
5. Montgomery D. C. (2013) *Design and Analysis of Experiments*. John Wiley & Sons Inc. USA.
6. Neter J., Wasserman W., Kutner M. H. (1983) *Applied Linear Regression Models*. R.D. Irwin Inc. USA.
7. Blei D. M., Ng A. Y., Jordan M. I. (2003) Latent Dirichlet Allocation. *Journal of Machine Learning Research*. (3), 993–1022.
8. Han J., Kamber M., Pei J. (2012) *Data Mining. Concepts and Techniques*. Elsevier.
9. German O. V., Germanovich E. I., Naidenko V. G. (1992) Boolean Optimization Algorithm on (0,1)-Matrices. *Journal of Computational Mathematics and Mathematical Physics*. 32 (7), 1114–1125 (in Russian).
10. German O. V. (1996) Decision Principle for the Minimum Covering Problem of a 0,1-Matrix. *Cybernetics and Systems Analysis*. (1), 135–145 (in Russian).
11. German Yu. O., German O. V., Nasr S. (2019) Information Extraction Method from a Resume (CV). *Works BSTU. Series 3. Physics and Mathematics Science and Computer Science*. (1), 64–68.

Received: 8 January 2025

Accepted: 22 May 2025

Вклад авторов

Мигалевич С. А. определил способ генерации кластеров ключевых слов и формирования текстов вопросов к кластерам.

Герман Ю. О. осуществила взаимодействие системы генерации вопросов с ChatGPT для формирования вопросов и получения ответов.

Герман О. В. определил общую концепцию работы и основные задачи.

Authors' contribution

Migalevich S. A. defined the method to generate clusters of keywords and creating questions associated to clusters.

German Ju. O. realized the interaction of the proposed system and ChatGPT to form questions and getting answers.

German O. V. defined the general concept of the work and the related basic tasks.

Сведения об авторах

Мигалевич С. А., нач. центра информатизации и инновационных разработок, соискатель каф. программного обеспечения информационных технологий, Белорусский государственный университет информатики и радиоэлектроники (БГУИР)

Герман Ю. О., канд. техн. наук, доц. каф. информационных технологий автоматизированных систем, БГУИР

Герман О. В., канд. техн. наук, доц. каф. информационных технологий автоматизированных систем, БГУИР

Адрес для корреспонденции

220013, Республика Беларусь,
Минск, ул. П. Бровки, 6
Белорусский государственный университет
информатики и радиоэлектроники
Тел.: +375 17 293-23-20
E-mail: migalevich@bsuir.by
Мигалевич Сергей Александрович

Information about the authors

Migalevich S. A., Head of the Center of Informatization and Innovations, Applicant at the Department of Software Provision of Information Technologies, Belarusian State University of Informatics and Radioelectronics (BSUIR)

German Ju. O., Cand. Sci. (Tech.), Associate Professor at the Information Technologies in Automated System Department, BSUIR

German O. V., Cand. Sci. (Tech.), Associate Professor at the Information Technologies in Automated System Department, BSUIR

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovki St., 6
Belarusian State University
of Informatics and Radioelectronics
Tel.: +375 17 293-23-20
E-mail: migalevich@bsuir.by
Migalevich Sergei Alexandrovich