DOI 10.37882/2223-2966.2026.03.19

СИСТЕМА СРАВНЕНИЯ ТЕКСТОВ: ЭТАПЫ ПРЕДОБРАБОТКИ И ОЦЕНКА ОРИГИНАЛЬНОСТИ ДОКУМЕНТОВ

TEXT COMPARISON SYSTEM: PREPROCESSING STAGES AND DOCUMENT ORIGINALITY ASSESSMENT

K. Krez
Y. Shneiderov
V. Golushko

Summary. This article presents a methodology for automated document originality assessment based on the integration of modern natural language processing methods and classical text comparison algorithms. The proposed approach includes a three-stage data processing algorithm: preliminary document cleaning and normalization, their semantic vectorization using the Sentence-BERT model, and subsequent similarity assessment using a hybrid algorithm. During the vectorization stage, the text is converted into 384-dimensional embeddings reflecting its semantic content, enabling efficient semantic searches for potential borrowing sources using approximate nearest neighbor search (ANS) algorithms. To accurately quantify the degree of similarity, the hashing shinnling method is used, ensuring deterministic comparison of text fragments. The developed algorithm automates the process of verifying the uniqueness of academic papers, reducing the complexity of manual verification, and increasing the reliability of the results. The proposed hybrid approach combines the high efficiency of semantic search with the accuracy of classical comparison methods and can be effectively applied when working with large text databases.

Keywords: vectorization of documents, evaluation of the originality of documents, equalization, embedding, shingle.

Крез Карина Сергеевна
Аспирант, УО Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
karinakrez04@gmail.com
Шнейдеров Евгений Николаевич
Кандидат технических наук, доцент, УО Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
shneiderov@bsuir.by
Голушко Вадим Игоревич
УО Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Беларусь
vadimgolushko2004@gmail.com

Аннотация. В статье представлена методика автоматизированной оценки оригинальности текстовых документов, основанная на интеграции современных методов обработки естественного языка и классических алгоритмов сравнения текста. Предложенный подход включает трехэтапный алгоритм обработки данных: предварительную очистку и нормализацию документов, их семантическую векторизацию с использованием модели Sentence-BERT и последующую оценку степени сходства с помощью гибридного алгоритма. На этапе векторизации текст преобразуется в 384-мерные вложения, отражающие его семантическое содержание, что позволяет эффективно проводить семантический поиск потенциальных источников заимствований с использованием алгоритмов приближенного поиска ближайшего соседа (ИНС). Для точной количественной оценки степени совпадения используется метод шинлинга с хешированием, обеспечивающий детерминированное сравнение фрагментов текста. Разработанный алгоритм позволяет автоматизировать процесс проверки уникальности научно-учебных работ, снижая сложность ручной проверки и повышая надежность результатов. Предложенный гибридный подход сочетает в себе высокую эффективность семантического поиска с точностью классических методов сравнения и может эффективно применяться при работе с большими текстовыми базами данных.

Ключевые слова: векторизация документов, оценка оригинальности документов, выравнивание, встраивание, шингл.

Введение

В контексте цифровизации образовательной и научной среды проблема проверки оригинальности текстовых документов приобретает особую актуальность. Ежегодный рост объема научных публикаций, курсовых и выпускных квалификационных работ приводит к усложнению задач по выявлению заимствований и объективной оценке уникальности материалов. Традиционные методы, основанные исключительно на лексическом совпадении фрагментов текста, все чаще оказываются недостаточными, поскольку не учитывают семантическую близость перефразированных или структурно модифицированных заимствований.

Современные достижения в области обработки естественного языка (далее — NLP) открывают новые возможности для построения интеллектуальных антиплагиатных систем, способных анализировать не только форму, но и смысл текста. Использование моделей нейронных сетей семейства BERT и технологий векторного представления (далее — эмбеддинг) позволяет перейти

от поверхностного сравнения строк к семантическому поиску и более точной оценке сходства документов.

Статья посвящена разработке и обоснованию комплексного подхода к автоматизированной проверке подлинности документов, сочетающего методы семантической векторизации и классические алгоритмы шинглирования. Предложенная методология в первую очередь ориентирована на применение в образовательной сфере и направлена на повышение точности, объективности и производительности систем обнаружения займов при работе с большими объемами текстовых данных.

Аналогичные алгоритмы для оценки оригинальности текста

Для повышения эффективности поиска в больших массивах документов используются методы MinHash и Locality-Sensitive Hashing (далле — LSH), которые позволяют аппроксимировать коэффициент Жаккара между наборами шинглов и значительно сократить время сравнения [1].

Теоретически этот подход был обоснован в работах А. Бродера, где в качестве количественных показателей заимствования были введены понятия «сходства» и «включительности» документов [2].

Несмотря на высокую точность обнаружения прямых копий, методы шинглирования имеют ряд ограничений: они чувствительны к перестановке слов, неустойчивы к перефразированию и не учитывают семантический контекст текста.

Значительный шаг вперед был сделан с появлением распределенных векторных представлений слов и документов. Модели Word2Vec и Doc2Vec позволяют отображать слова и тексты в непрерывном векторном пространстве, где близость векторов отражает семантическую близость понятий [4, 5].

Использование этих моделей в задачах по борьбе с плагиатом частично решило проблему синонимии и изменчивости формулировок. Документы сравниваются не по совпадению отдельных слов, а по близости их семантических представлений.

Однако эти методы имеют ограниченную способность учитывать сложный контекст и иерархию значений в длинных текстах. Кроме того, усреднение векторов слов часто приводит к потере структурной информации.

Современный этап развития систем оценки оригинальности связан с внедрением трансформерных нейронных сетей, прежде всего моделей семейства BERT. В отличие от предыдущих подходов, такие модели формируют контекстно-зависимые вложения, где значение слова определяется его окружением [6].

Особую роль в задачах сравнения текстов сыграла архитектура Sentence-BERT, предложенная Н. Реймерсом и И. Гуревичем, которая использует сиамскую сеть для получения компактных векторных представлений предложений и документов [7]. Эти эмбеддинги демонстрируют высокую корреляцию с оценками семантической близости, полученными человеком, и успешно используются в задачах кластеризации, поиска дубликатов и обнаружения займов.

В ряде работ предлагается комбинировать семантические эмбеддинги с алгоритмами приближенного ближайшего соседа (далее — ИНС) для масштабируемого поиска похожих документов в больших базах данных. Наиболее известными реализациями являются HNSW и FAISS, которые обеспечивают логарифмическую сложность поиска с минимальной потерей точности [8, 9].

В данной статье будет описан гибридный подход, сочетающий преимущества семантического анализа и классических детерминированных методов.

Типичная архитектура таких систем включает два уровня:

1. Семантический фильтр — на основе эмбеддингов нейронной сети и поиска ИНС выбирается ограниченное количество потенциально похожих документов.
2. Точный анализ — для выбранных кандидатов используются методы шинглинга, хеширования или другие методы символического сравнения для вычисления количественной меры совпадения.

Основные результаты

Первый этап, «Подготовка данных», включает очистку текста от структур сервиса и его нормализацию. Второй этап, «Векторизация документа», является ключом к преобразованию текстовой информации в машиночитаемый (далее — векторизация) формат. Для этой цели используется архитектура Sentence-BERT, которая генерирует многомерные векторные представления. Заключительный этап, «Расчет оригинальности», основан на поиске ближайших семантических соседей и детальном сравнении фрагментов текста (далее — шинглы) для количественной оценки сходства. Общая схема процесса показана на рисунке 1.

Этап 1. Подготовка данных.

Цель этого этапа — извлечение полезного текстового контента и удаление шумовых данных.

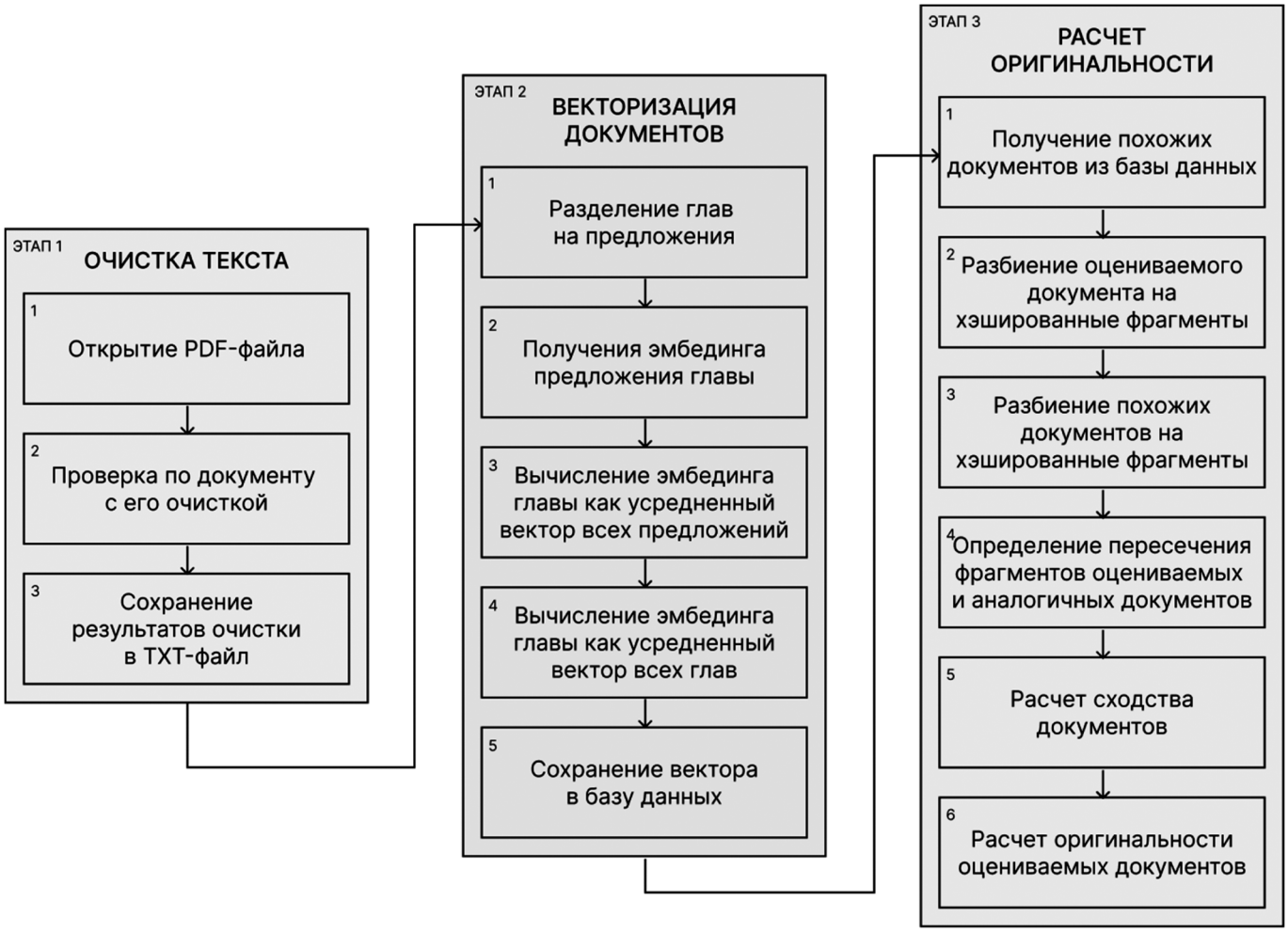

Рис. 1. Общая схема предварительной обработки документа

Этап 1.1. Открытие и чтение файлов *.pdf или *.word. На начальном этапе исходный файл читается постранично в формате *.pdf или *.word. Главная задача этапа — исключение элементов, не несущих семантическую нагрузку для анализа уникальности (так называемые «шумовые» строки). К ним относятся: титульные страницы, аннотации, содержание, ссылки, приложения, а также подписи к графическим объектам (например, строки, содержащие шаблоны «Рисунок…», «Таблица…») [10].

Этап 1.2. Прохождение документа с его очисткой. В процессе итерации по страницам формируется структурированный словарь данных. Ключами словаря являются названия структурных единиц текста (например: разделы, главы, подразделы), а значениями — очищенный текст, соответствующий этим разделам. Этот подход позволяет сохранить логическую структуру документа, что крайне важно для корректной векторизации. При формировании текста разделы разделяются между собой специальными маркерами — двойными переносами строк, что упрощает их последующую программную обработку.

Этап 1.3. Сохранение результата очистки в файл *.txt. Нормализованный текст сохраняется в промежуточный файл *.txt. Это решение обусловлено необходимостью разделения процессов предварительной обработки и анализа:

— при изменении модели векторизации повторная очистка исходного файла не требуется;
— этапы векторизации и вычисления уникальности технически разделены. Поскольку сравнение требует данных не только из текущего документа, но и из многих других документов из базы данных, передача данных исключительно через оперативную память неэффективна.

Этап 2. Векторизация документа. На этом этапе текстовая информация преобразуется в эмбеддинги, отражающие семантическое значение.

Этап 2.1. Разделение глав на предложения.

Текст, загруженный из TXT-файла, десериализуется в структуру данных. Содержимое каждой главы токенизируется на уровне разделения предложений, образуя список единиц для анализа.

Этап 2.2. Получение эмбеддинга предложений заголовка.

Каждое предложение преобразуется в вектор с использованием модели нейронной сети «paraphrase-MiniLM-L6-v2». Эта модель является модификацией архитектуры BERT [11], адаптированной библиотекой sentence-transformers (SBERT) [12] для задач семантического поиска. Используемая модель работает с векторами размерности 384 (в отличие от 768 для базового BERT), что обеспечивает баланс между точностью семантического представления и скоростью вычислений. Модель имеет 6 слоев трансформера, что также способствует повышению производительности. На рисунке 2 представлен график, где по оси X отложено количество измерений, а по оси Y — точность. Линии отражают различные задачи, для решения которых были обучены модели.

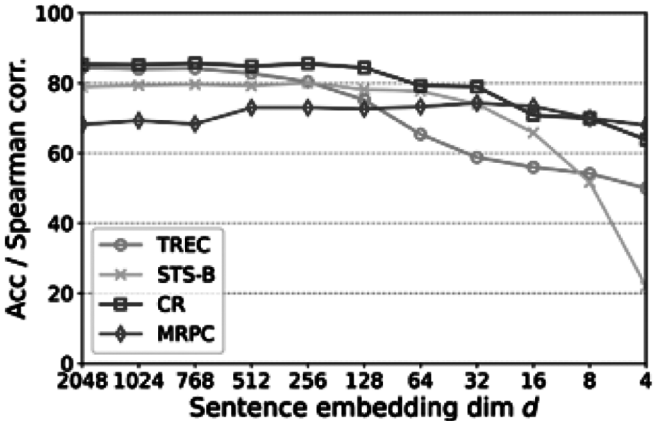

Рис. 2. График точности полученного результата в зависимости от размерности измерений

В данном исследовании по оси Y отложен коэффициент ранговой корреляции Спирмена для задачи оценки семантического сходства (STS-B) и индекс точности для задач классификации (TREC, CR, MRPC).

Используемые наборы данных характеризуются следующими параметрами:

— TREC: классификация вопросов (500 тестовых случаев, 6 классов);
— STS-B: оценка семантической близости предложений (1379 тестовых пар, 5-балльная шкала);
— CR: бинарная классификация тональности (3773 предложения);
— MRPC: определение парафраза (1726 тестовых пар).

Анализ представленных графических данных позволяет сделать вывод, что размерность вектора в 384 измерения является оптимальной для задачи семантического сходства. Дальнейшее увеличение размерности (до 512 и 768) не приводит к значительному увеличению метрик, в то время как уменьшение до 256 измерений демонстрирует сопоставимые результаты, что подтверждает избыточность многомерных векторов для задач такого типа.

Процесс токенизации и векторного кодирования:

1. Токенизатор BERT преобразует текст в последовательность идентификаторов (ID) с использованием статистического метода сегментации слов на подслова. Это позволяет эффективно обрабатывать редкие лексемы и слова с богатой морфологией.
2. Слова, отсутствующие в словаре, разлагаются. Например, токен «illumination» можно разделить на токены ['under', '##light', '##ka'], где префикс ## обозначает подтокены, не являющиеся началом слова.
3. Каждому токену присваивается вектор фиксированной размерности (в данном случае 384). Эти веса оптимизируются на этапе предварительного обучения и накапливают дистрибутивную и семантическую информацию о токене.

Для корректного функционирования модели в структуру входных данных интегрированы специальные управляющие токены:

— [CLS] (ID 101): размещается в начале последовательности; его конечное скрытое состояние используется в качестве агрегированного представления всей фразы для задач классификации;
— [SEP] (ID 102): служит разделителем между контекстными парами (например, в задачах анализа пар предложений) и указывает на конец последовательности.

Вектор всего документа вычисляется аналогичным образом: векторы всех глав усредняются и вычисляются по формуле 1. Полученный вектор представляет собой сжатое семантическое представление всего текста.

$$PE_{(pos,2i)} = \sin\sin\left(\frac{pos}{10000^{\frac{2i}{d\text{model}}}}\right), \quad (1)$$

где *PE* — позиционное встраивание, pos — позиция токена в последовательности (например, 0 для первого токена, 1 для второго и т. д.), *i* — индекс измерения в векторе позиционного встраивания (от 0 до *d_model*), *d_model* — размерность модели (количество измерений в векторе встраивания, 384).

Позиционные встраивания формируются на основе последовательности токенов, включая знаки препинания. Архитектура оригинальной модели BERT имеет 512 фиксированных позиционных индексов, что ограничивает максимальную длину входной последовательности соответствующим количеством токенов.

Встраивания типов токенов используются для решения задачи классификации пар предложений. В со-

временных версиях архитектуры этот механизм сохраняется для обеспечения структурной совместимости, но фактически не используется: всем токенам по умолчанию присваивается идентификатор сегмента «А». Полученный входной вектор формируется путем пошагового суммирования трех типов эмбеддингов (токенов, позиций и сегментов), после чего он передается в блоки кодировщика трансформера.

Отличительной особенностью моделей SBERT является наличие слоя пулинга после выходного слоя трансформера. Пулинг — это операция агрегирования признаков, направленная на получение фиксированного вектора исходных данных из последовательности векторов токенов. Рассматриваемая модель использует метод среднего пулинга (Mean Pooling), в котором окончательное представление предложения вычисляется как арифметическое среднее значений векторов всех его токенов.

Для формирования семантического корпуса документа реализован многоуровневый подход к агрегированию данных:

— вектор главы вычисляется как центроид в N-мерном пространстве путем нахождения арифметического среднего векторов всех предложений, составляющих эту главу;
— окончательный усредненный вектор, представляющий общее семантическое направление документа, сохраняется в базе данных векторов для последующего использования.

Например, возьмем 2 вектора в двумерном пространстве. Вектор 1 = (1, 2) и вектор 2 = (3, 4). Теперь найдем сумму векторов: это будет вектор суммы = (1 + 3, 2 + 4) = (4, 6). Чтобы получить вектор среднего значения, остается только разделить вектор суммы на 2, поскольку в сумме получилось 2 вектора: (4, 6)/2 = (4/2, 6/2) = (2, 3). Вектор суммы отражает общее направление, заданное векторами 1 и 2, и то же самое происходит с векторами предложений. Полученное среднее значение указывает общее направление предложений, а следовательно, и глав. Получив вектор главы, он добавляется к списку векторов всех глав и продолжается по документу до тех пор, пока не будут получены векторы всех глав. После этого выполняется та же операция, уже с векторами глав, и получается вектор документа, который сохраняется в базе данных.

Этап 3. Оценка оригинальности документа.

Этап 3.1. Семантический поиск и обоснование выборки.

Для выявления потенциальных источников заимствований выполняется поиск семантически схожих векторов в базе данных. Алгоритм извлекает пять наиболее релевантных документов (k=5). Этот размер выборки эмпирически обоснован: наибольшая степень сходимости текста наблюдается среди первых результатов. Расширение выборки за пределы пяти кандидатов нецелесообразно, поскольку включение тематически отдаленных документов приводит к неоправданному снижению вычислительной эффективности и риску искажения окончательной оценки путем анализа нерелевантного контента.

Поиск реализуется с использованием алгоритма приблизительного ближайшего соседа (даллее — ANN) [13]. Использование ANN вместо метода полного перебора обусловлено необходимостью высокой вычислительной эффективности при работе с большими объемами данных. Ошибка алгоритма ИНС устраняется на последующих этапах анализа, поскольку векторный поиск служит лишь фильтром для отбора кандидатов, а окончательный расчет основан на детерминированном сравнении текстовых сигнатур.

Этап 3.2. Метод шинглирования и хеширования.

Для проведения детального сравнения документов используется метод N-грамм (даллее — шинглов). В данной статье используется размер шингла n=5. Процесс декомпозиции текста реализуется по принципу «скользящего окна»: при сдвиге окна на одно слово формируется новый перекрывающийся фрагмент.

Например, для фразы «Кот сидит на ковре и смотрит в окно» при n=5 формируется набор: [Кот сидит на ковре и][сидит на ковре и смотрит]... и т. д. Для оптимизации процесса сопоставления каждый шингл преобразуется в уникальный числовой идентификатор — хеш-код. Хэширование гарантирует, что идентичные фрагменты текста будут иметь одинаковые значения. Для каждого документа генерируется множество уникальных хешей, что автоматически исключает дублирование фрагментов в одном и том же тексте.

Этап 3.3. Математическая модель для расчета оригинальности. Для определения степени сходства анализируемого документа с найденным источником.

$$S = \frac{I}{U}, \qquad (2)$$

где: I — количество хешей анализируемого документа, найденных в наборе хешей найденных источников (пересечение); U — общее количество уникальных хешей (длина набора) целевого документа. Показатель S варьируется в диапазоне [0; 1], где 1 соответствует полному совпадению всех текстовых фрагментов, а 0 — полному отсутствию идентичных шинглов.

Итоговый показатель оригинальности документа (в процентах) рассчитывается по формуле 3:

$$O = 100 - (S \cdot 100), \quad (3)$$

где O — оригинальность документа, S — сходство документов.

Поскольку наше сходство определяется в диапазоне от 0 до 1, то для получения процента мы умножаем на 100. Процент оригинальности можно получить, вычитая процент сходства из 100.

Заключение

Авторы статьи разработали и обосновали комплексную методологию автоматизированной проверки уникальности текстовых документов, ориентированную на применение в образовательной сфере. Предложенная система решает проблему проверки содержания с помощью трехэтапного конвейера обработки данных, сочетающего современные методы обработки естественного языка и классические алгоритмы сравнения:

1. Предварительная обработка и нормализация — реализован эффективный механизм очистки документов (форматы *.pdf или *. word) от «шумовых» элементов (титульные страницы, библиографии, подписи к рисункам) с использованием регулярных выражений.
2. Семантическая векторизация — выбрана модель нейронной сети Sentence-BERT. Она преобразует текст в 384-мерные векторы, что, по мнению авторов, является оптимальным балансом между точностью семантического представления и скоростью вычислений. Используется иерархический подход: векторы предложений объединяются в векторы глав, а затем в вектор документа.
3. Гибридный алгоритм поиска и сравнения — для быстрого отбора кандидатов используется приближенный алгоритм поиска ближайшего соседа (ANN), который находит 5 наиболее семантически близких документов в базе данных.

Для точного расчета процента оригинальности используется детерминированный метод сравнения хешированных шинглов (n-грамм длиной 5 слов). Итоговая оценка основана на коэффициенте (пересечении множеств шинглов).

Представленный алгоритм позволяет полностью автоматизировать процесс проверки различных документов, таких как статьи, курсовые работы, диссертации и лабораторные работы, минимизируя ручной труд. Гибридный подход (ANN для скорости и шинглы для точности) обеспечивает высокую производительность при работе с большими базами данных, а использование моделей BERT позволяет учитывать семантический контекст при первоначальном отборе, отфильтровывая нерелевантные документы.

ЛИТЕРАТУРА

1. Leskovec J., Rajaraman A., Ullman J. Mining of Massive Datasets. — Cambridge: Cambridge University Press, 2014.
2. Broder A.Z. On the resemblance and containment of documents // Compression and Complexity of Sequences. — 1997. — P. 21–29.
3. Salton G., McGill M. Introduction to Modern Information Retrieval. — New York: McGraw-Hill, 1983.
4. Mikolov T. et al. Efficient estimation of word representations in vector space // arXiv preprint arXiv:1301.3781, 2013.
5. Le Q., Mikolov T. Distributed representations of sentences and documents // Proceedings of ICML, 2014. — P. 1188–1196.
6. Devlin J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding // NAACL-HLT, 2019. — P. 4171–4186.
7. Reimers N., Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks // EMNLP-IJCNLP, 2019. — P. 3982–3992.
8. Johnson J., Douze M., Jégou H. Billion-scale similarity search with GPUs // IEEE Big Data, 2017.
9. Malkov Y., Yashunin D. Efficient and robust approximate nearest neighbor search using HNSW // IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. — Vol. 42, No. 4. — P. 824–836.
10. Крез К.С. Разработка модуля для предобработки документов обнаружения заимствований / Крез К.С., Ефименко Д.Д., Войнилович Н.Ю. // Информационные технологии и системы 2025: материалы Междунар. науч. конф., Минск, 16–17 нояб. 2023 г. / Белорусский государственный университет информатики и радиоэлектроники ; редкол.: Л.Ю. Шилин [и др.]. — Минск, 2025. — С. 75–76.
11. BERT Documentation // Hugging Face: [website]. — URL: https://huggingface.co/docs/transformers/model_doc/bert (accessed: 04.11.2025).
12. Sentence Transformers Documentation // SBERT.net: [website]. — URL: https://sbert.net/ (accessed: 04.11.2025).
13. [Understanding Approximate Nearest Neighbor (ANN) // Elastic Blog: [website]. — URL: https://www.elastic.co/blog/understanding-ann (accessed: 10.11.2025).

© Крез Карина Сергеевна (karinakrez04@gmail.com); Шнейдеров Евгений Николаевич (shneiderov@bsuir.by);
Голушко Вадим Игоревич (vadimgolushko2004@gmail.com)
Журнал «Современная наука: актуальные проблемы теории и практики»