УДК 004.891.2

РАЗРАБОТКА И ОПТИМИЗАЦИЯ LLM-СЕРВИСА ДЛЯ КОНСУЛЬТИРОВАНИЯ ПОЛЬЗОВАТЕЛЕЙ ПО ЭКСПЛУАТАЦИИ ПЛАТФОРМЫ — СРАВНЕНИЕ ПОДХОДОВ ПРОЕКТИРОВАНИЯ ВОПРОСНО-ОТВЕТНЫХ СИСТЕМ

Хисаметдинова Д.Н.[1] (студент)

Научный руководитель – кандидат технических наук Ходненко И. В. [1]

[1]Университет ИТМО

Аннотация

Статья посвящена разработке и оптимизации LLM-сервиса для консультирования пользователей по эксплуатации платформы, с акцентом на сравнении подходов построения вопросно-ответных систем. Предложена архитектура, сочетающая лексический поиск, семантическое векторное индексирование (Sentence-BERT, FAISS), генерацию синонимов и уточнение запросов. В качестве базы знаний используется разметка документации, агрегированная в пары «вопрос–ответ». Реализован асинхронный пайплайн автоматического извлечения ключевых слов и аугментации синонимичными формулировками, что обеспечило увеличение recall и повышение релевантности поиска. Запросы пользователя уточняются с учетом ключевых терминов, после чего выполняется гибридный поиск: предварительный отбор кандидатов по BM25+, далее ранжирование по косинусному сходству эмбеддингов и итоговая фильтрация на основе soft matching (Jaccard, SequenceMatcher, FuzzyWuzzy). Оптимальные пороги этих метрик определялись алгоритмическим перебором на тестовой выборке для максимизации F1-score, отражающего баланс между точностью и полнотой. Проведена автоматизированная оценка качества на репрезентативной тестовой выборке, а также сравнительный анализ различных конфигураций алгоритма и LLM-моделей. В итоговом сервисе был использован вариант, обеспечивший корректные ответы практически на все вопросы, со значениями cosine similarity >0.9, recall >0.7, F1-score >0.8.

Ключевые слова

Большая языковая модель, вопросно-ответная система, семантический поиск, BM25+, Sentence-BERT, FAISS, аугментация данных, soft-matching, эмбеддинги.

База знаний формировалась из исходной документации, которая разбивалась на небольшие фрагменты (чанки), не превышающие размер окна модели. Для каждого отрывка был создан связанный набор вопросно-ответных пар, привязанных к своему исходному тексту. Такая структура обеспечивает локальную релевантность поиска и позволяет учитывать особенности предметной области при обработке запросов.

Для увеличения разнообразия формулировок вопросов и повышения полноты поиска применена аугментация данных при помощи большой языковой модели (LLM): для каждого исходного вопроса автоматически сгенерировано несколько семантически эквивалентных вопросов с тем же ответом. Генерация выполнена при низком значении параметра температуры (≈0.1), что обеспечивает минимальные отклонения по смыслу и стилистике – по сути, LLM предложила синонимичные формулировки вопросов. Например, для вопроса «Как сбросить настройки устройства X?» модель генерировала варианты: «Как выполнить сброс настроек на устройстве X?» и т.п. Все сгенерированные варианты были добавлены в базу знаний наряду с исходными вопросами. С помощью LLM из текста вопроса извлекались ключевые слова (наиболее значимые термины запроса). Эти keywords сохранились для возможного использования при ранжировании и для быстрого сравнения запросов методом Жаккара.

Оригинальные и сгенерированные вопросы были преобразованы в векторное представление с использованием предобученной модели Sentence-BERT (SBERT) [1]. Она быстро формирует эмбеддинги предложений в пространстве высокой размерности, в котором косинусное расстояние отражает семантическую близость предложений. При тестировании нескольких вариантов SBERT, результат которого отражён на рисунке 1,

модель paraphrase-multilingual-mpnet-base-v2 продемонстрировала наивысшую F1-меру (~0.77) и высокую полноту, поэтому выбрана для системы; более компактная модель all-MiniLM-L6-v2 заметно уступает по качеству.

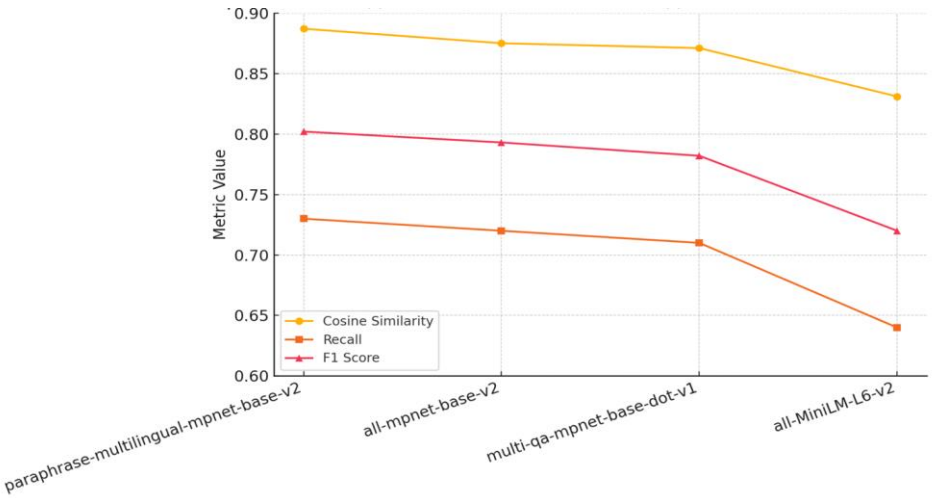

Рис. 1 – Сравнение моделей Sentence-BERT на задаче семантического поиска

Все эмбеддинги вопросов объединяются в матрицу и индексируются с помощью библиотеки FAISS, реализующей эффективный поиск ближайших соседей по косинусному сходству между векторами. Для лексического поиска по базе строится инвертированный индекс на основе токенизации вопросов и расчёта BM25+ — классического метода ранжирования по частотам терминов и обратной частоте документа, что позволяет находить вопросы с точными совпадениями по ключевым словам.

В реализации системы применён модуль Rank-BM25 для вычисления BM25-score для каждого запроса, а для семантического поиска — плоский индекс IndexFlatIP библиотеки FAISS, позволяющий находить ближайшие по косинусной мере векторные представления вопросов. Такой подход обеспечивает точный поиск даже на сравнительно небольших коллекциях данных, жертвуя скоростью в пользу качества совпадений.

При поступлении запроса пользователя выполняется его нормализация, извлечение ключевых слов и преобразование в эмбеддинг через выбранную модель Sentence-BERT. Далее запускается гибридный поиск: BM25+ возвращает кандидатов с максимальным текстовым совпадением, сортируя их по убыванию BM25-score. FAISS находит k ближайших вопросов по косинусной близости эмбеддингов.

Так как численные значения BM25 и косинусного сходства лежат в разных диапазонах, их прямое суммирование невозможно. Поэтому обе метрики нормализуются (BM25-score приводится к диапазону [0, 1], например, делением на максимальный score в выборке). Для ранжирования кандидатов используется итоговая метрика релевантности — взвешенная сумма нормированных значений:

$$S_{hybrid}(q, Q_i) = \alpha \cdot BM25(q, Q_i) + (1 - \alpha) \cdot \cos\left(\mathbf{v_q}, \mathbf{v_{Q_i}}\right), \quad (1)$$

где q – запрос, Q_i – i-й кандидат-вопрос из базы знаний, v_q, v_{Q_i} – их эмбеддинги; $BM25(q, Q_i)$ – нормированный BM25-score, $\cos\left(v_q, v_{Q_i}\right)$ – косинусная мера сходства; α – вклад лексического поиска в итоговый рейтинг (в эксперименте оптимальное значение α = 0.2, то есть 20% веса приходится на BM25).

Кандидаты сортируются по убыванию S_{hybrid}, и топ-1 результат выбирается в качестве наиболее подходящего ответа. Однако, окончательное решение о возврате

ответа основывается не только на позициях в ранжированном списке, но и на применении пороговой семантической фильтрации, описанной далее и показанной на рисунке 2.

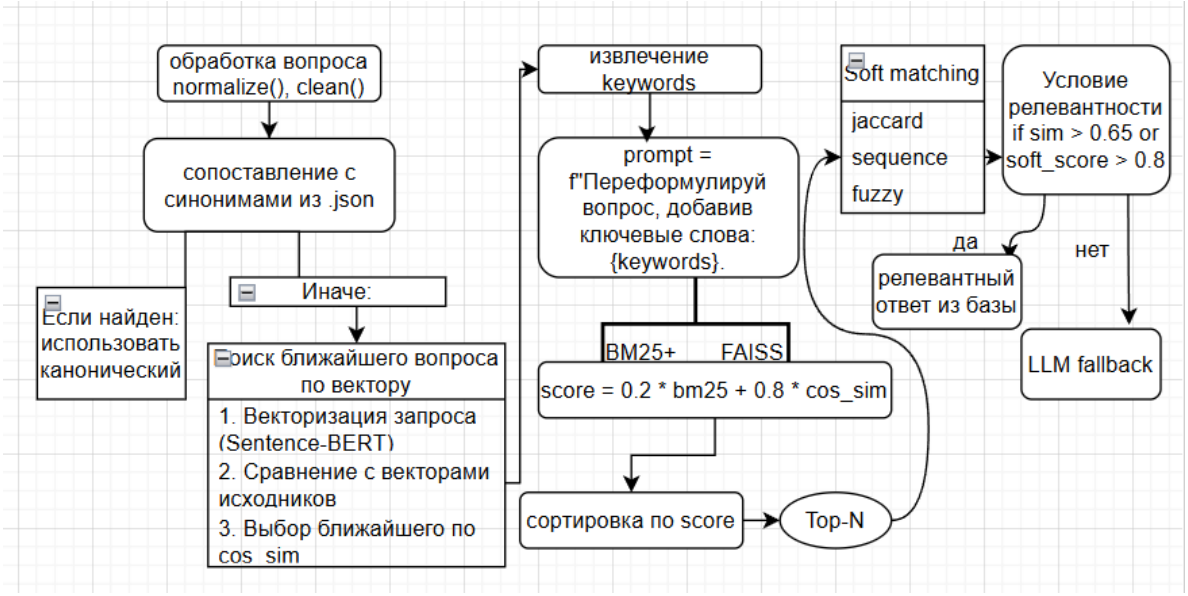

Рис. 2 – Схема работы пайплайна гибридного поиска и семантической фильтрации

Для повышения качества сопоставления вопросов реализован этап soft-matching — проверка текстового сходства с помощью метрик, не требующих полного совпадения. Этот этап применяется, когда семантическое сходство (по эмбеддингам) между вопросом пользователя и кандидатом из базы недостаточно высоко. Цель — уловить переформулированные и синонимичные запросы, избегая случайных совпадений.
В качестве метрик выбраны: коэффициент Жаккара, расстояние Левенштейна (реализовано через SequenceMatcher из difflib) и метрика fuzzy (разработка на основе измерения похожести двух строк с учетом перестановок слов).

Коэффициент Жаккара вычисляется как отношение размера пересечения множеств слов двух вопросов к размеру их объединения. Метрика Левенштейна измеряет минимальное число правок, необходимых для превращения одной строковой формулировки в другую; нормированное значение на основе SequenceMatcher показывает долю совпадения символов с учетом порядка. Fuzzy вычисляет расстояние на основе наиболее похожих подстрок, более лояльно относясь к перестановке слов. Каждая из этих метрик дает значение сходства в диапазоне от 0 до 1. Был взят максимум из трех метрик. Например, два вопроса могут иметь низкий Jaccard (мало общих слов), но высокий Fuzzy Score (те же слова в другом порядке), и тогда soft_score будет высоким.

После основного гибридного поиска (BM25 + FAISS) для каждого кандидата дополнительно вычисляется soft_score. Вопрос считается релевантным, если выполняется хотя бы одно из двух условий:

$$\text{cos\_sim} > T_{\text{sim}} \quad \text{или} \quad \text{soft\_score} > T_{\text{soft}} \tag{2}$$

где cos_sim — косинусное сходство между эмбеддингами запроса и вопроса; soft_score — интегральная метрика текстового сходства (максимум из Жаккара, Левенштейна, Fuzzy); T_{sim} — порог для косинусного сходства; T_{soft} – порог для soft_score.

Soft-matching применяется после основного ранжирования кандидатов и не участвует в формировании начального списка результатов. Другими словами, сначала гибридный поиск отбирает топ-N потенциально подходящих вопросов, а затем для них выполняется проверка эквивалентности с помощью порогов cosine similarity и soft_score. Это предотвращает излишнее рассмотрение заведомо нерелевантных документов и

снижает вычислительную нагрузку (метрики soft-matching считаются не для всей базы, а только для нескольких кандидатов). Если ни один кандидат не удовлетворяет условиям, происходит попытка сгенерировать ответ с помощью LLM на основе общей информации.

Чтобы найти оптимальные пороговые значения (thresholds) был взят список тестовых вопросов реальных пользователей платформы. Для диапазона потенциальных значений порога косинусного сходства от 0.5 до 0.95 с шагом 0.01 проводилось испытание: для каждого threshold выполнялся полный цикл поиска кандидатов и семантической фильтрации при фиксированном T_{soft} = 0.8, после чего сравнивались полученные ответы с эталонными. По результатам вычислялись precision, recall и F1-score для каждого порога. На рисунке 3 показана зависимость F1-score от значения порога cosine similarity:

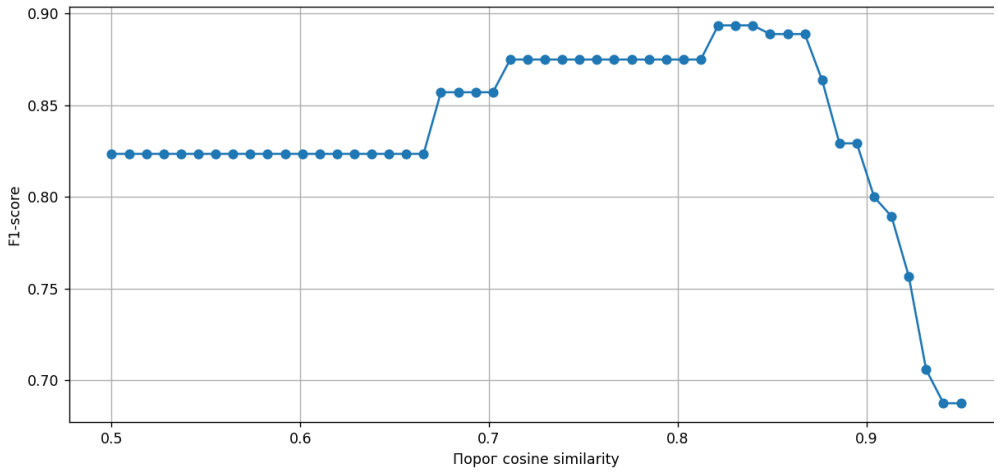

Рис. 3 – Зависимость F1-score от значения порога косинусного сходства

Видно, что при слишком низких порогах (менее 0.6) качество ниже из-за большого числа ложных позитивных срабатываний (высокий recall достигается ценой падения precision). С другой стороны, при слишком высоком пороге (более 0.8) система редко находит соответствия (precision высока, но многие правильные ответы пропущены, recall низкий). Максимум F1 наблюдается в районе T_{sim}≈ 0.65–0.70. При значении 0.65 достигался наилучший баланс – precision и recall оказались примерно равны. Так, при T_{sim}= 0.65 и T_{soft} = 0.8 система правильно сопоставила ~88% вопросов из тестового набора, дав при этом около 12% неверных соответствий. Иными словами, precision ≈ 0.88, recall ≈ 0.88, что дает F1 ≈ 0.88. Для сравнения, при использовании только семантического поиска (без BM25 и без soft-matching) на том же наборе вопросов F1 не превышала ~0.80, а при использовании только BM25 – ~0.75. Таким образом, введение гибридной схемы и дополнительной фильтрации заметно улучшило качество, что согласуется с результатами других исследований [2]. Система улавливает перефразированные вопросы и при этом практически не дает несоответствующих ответов. Подбор порогов на валидации позволил настроить систему под требуемый уровень качества. В практических условиях выбор конкретного значения может зависеть от приоритетов: например, для критически важной системы можно пожертвовать полнотой (увеличив пороги для минимизации ошибок).

На этих же тестовых вопросах и ориентируясь на те же метрики, что и выше, проведено сравнение следующих больших языковых моделей – Saiga LLaMA 8B и Qwen 32B GPTQ. Несмотря на то, что первая дообучалась на данных на русском языке, число параметров оказалось важнее, поэтому Qwen 32B GPTQ превосходит Saiga LLaMA 8B по точности на ~15%, по косинусному сходству — на ~12%, по полноте извлечения — на ~30%, что видно на рисунке 4.

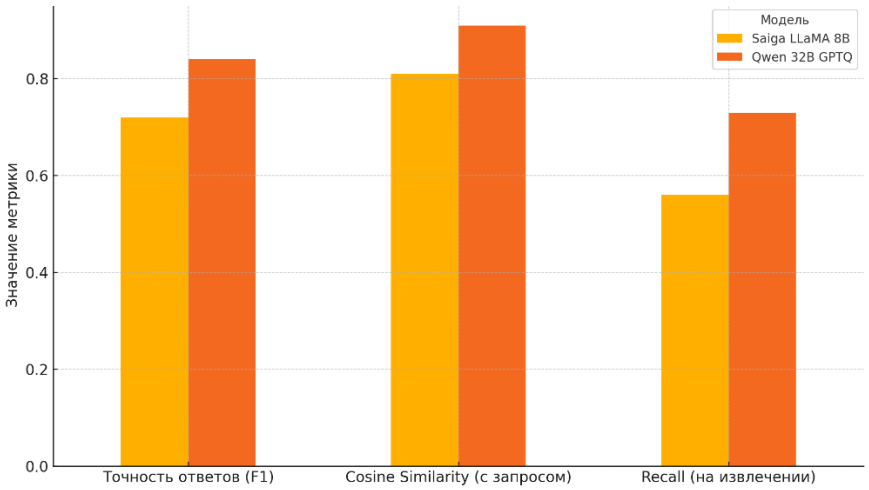

Рис. 4 – Сравнение LLM на тестовых вопросах

Разработанная гибридная система вопросно-ответного поиска продемонстрировала высокую эффективность за счёт комбинирования комплементарных подходов. Лексический поиск (BM25) обеспечил точность на уровнях терминов, а семантический (SBERT + FAISS) — покрытие смысловых вариаций. Однако именно сочетание этих методов, дополненное soft-matching и проверками эквивалентности, позволило достигнуть наилучших результатов, что подтверждено другими исследованиями [3].

На рисунке 5 видно, что отдельное применение BM25 или FAISS даёт умеренные значения: F1 около 0.63 и 0.57 соответственно. Их объединение (BM25 + FAISS) даёт прирост обеих метрик, а включение переранжирования через LLM (BM25 + FAISS + LLM rerank) ещё сильнее улучшает показатели (до F1 ≈ 0.75).

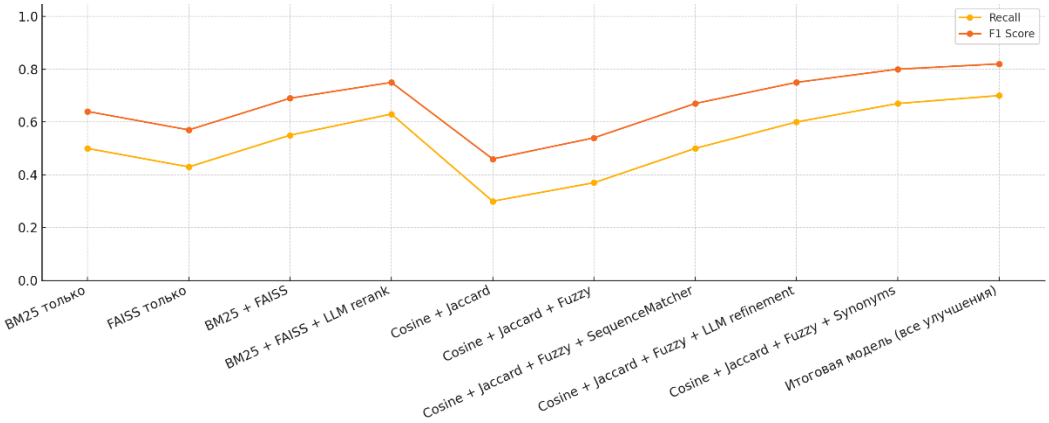

Рис. 5 – Итоговое сравнение всех комбинаций подходов

Однако наиболее заметный рост наблюдается при добавлении soft-модулей (Cosine + Jaccard + Fuzzy + SequenceMatcher). Несмотря на временное падение качества на отдельных конфигурациях (например, Cosine + Jaccard), постепенное включение новых метрик и проверок приводит к устойчивому улучшению результатов.

Финальная модель (с применением всех улучшений, включая synonyms и refinement через LLM) демонстрирует наивысшие значения: F1 ≈ 0.82, Recall ≈ 0.71. Это подтверждает, что многоступенчатая фильтрация и гибридное представление запросов позволяют значительно сократить количество ложных срабатываний и не упустить релевантные ответы.

Для повышения точности семантического сопоставления перспективным направлением является fine-tuning модели Sentence-BERT на данных конкретной предметной области. Это может быть выполнено на парах «вопрос–переформулированный вопрос» и «вопрос–неэквивалентный вопрос» с бинарной

разметкой (1 – эквивалентен, 0 – нет). Для этого подходит задача semantic textual similarity classification с использованием contrastive loss или triplet loss, где положительная пара — перефраз, а отрицательная — тематически близкий, но нерелевантный вопрос. Такая донастройка позволяет модели точнее учитывать смысловые различия и контекстные ограничения (например, отрицания и условия), которые универсальные эмбеддинги SBERT могут игнорировать.

Кроме того, можно использовать подход Multiple Negatives Ranking Loss (MNRL) из оригинальной реализации SBERT — при наличии большого числа положительных и негативных пар он показывает высокую эффективность в задачах поиска и сопоставления. В перспективе возможно объединение fine-tuned SBERT с обучаемым ранжированием (например, LightGBM или RankNet), где эмбеддинговое сходство становится лишь одним из признаков в модели.

По завершении исследовательской части работы, с применением наиболее эффективных подходов, был написан и развёрнут вопросно-ответный LLM сервис с бэкенд частью на FastAPI, фронтендом на Angular, его интерфейс представлен на рисунке 6.

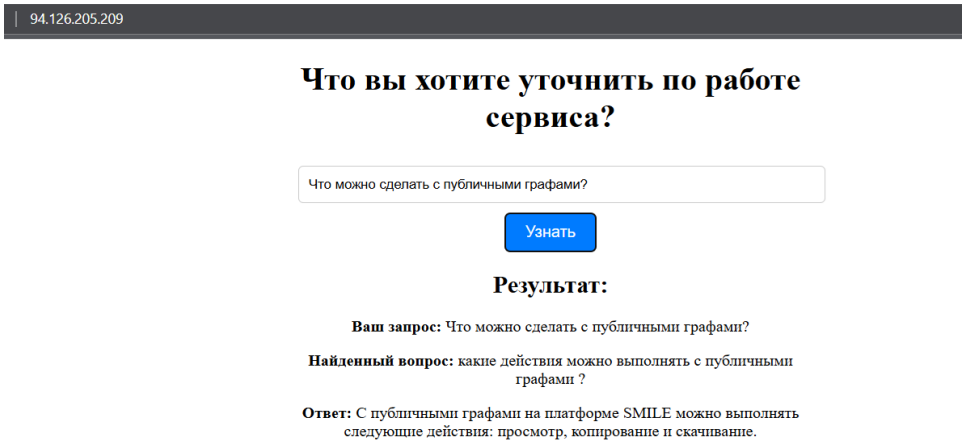

Рис. 6 – Пользовательский интерфейс сервиса

Генеративная модель развёрнута в отдельном изолированном контейнере на базе фреймворка vLLM, архитектура которого обеспечивает высокопроизводительный inference за счёт поддержки параллельной генерации ответов с использованием алгоритма PagedAttention. По сравнению с традиционной реализацией на базе transformers (HuggingFace), использование vLLM позволило сократить задержку ответа при одновременной обработке нескольких запросов, а также снизить потребление видеопамяти без потери качества генерации. В ходе эмпирического тестирования vLLM показал прирост производительности до 3 раз в условиях высокой нагрузки и стабильно обслуживал сервис на модели Qwen2.5-32B-Instruct-GPTQ при использовании одной GPU NVIDIA RTX 6000 Ada.

Таким образом, комплексный подход обеспечивает значимое преимущество по сравнению с отдельными компонентами. Он позволяет системе уверенно справляться с разнообразием формулировок пользовательских запросов, повышая и точность, и полноту.

Литература

1. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084.
2. Dewang Sultania, et al. Domain-specific Question Answering with Hybrid Search // arXiv:2412.03736.
3. Шалагин Н.Д. Обзор алгоритмов семантического поиска по текстовым документам // International Journal of Open Information Technologies. 2024. №7(1). С. 26–34.