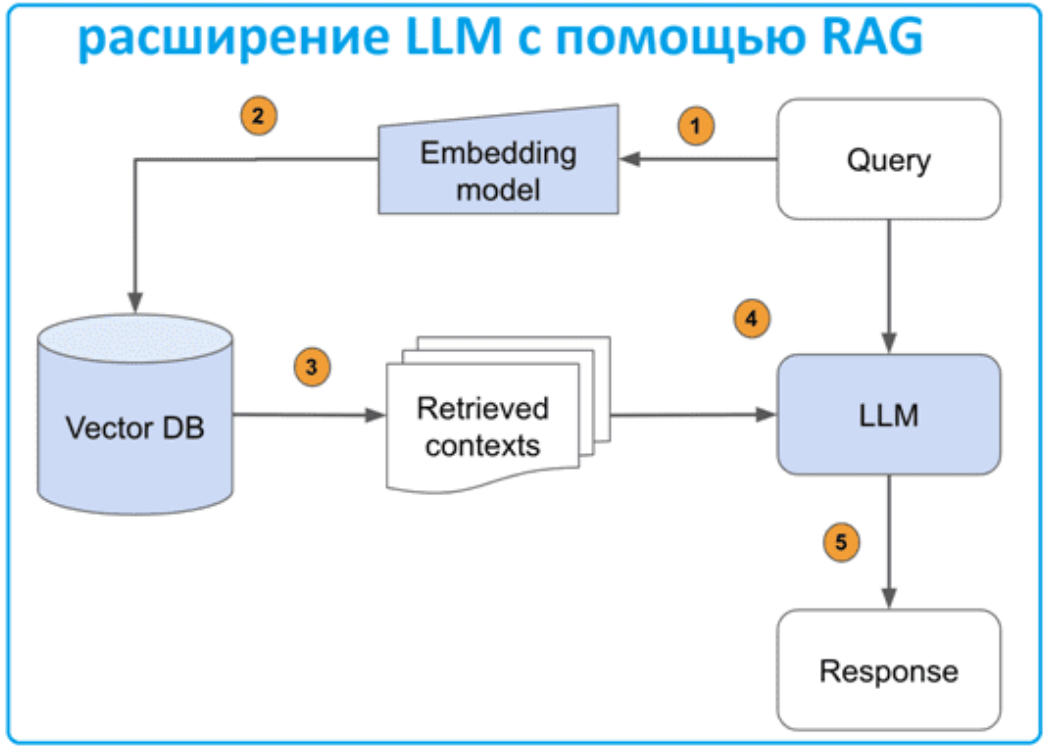
СБЕР УНИВЕРСИТЕТ

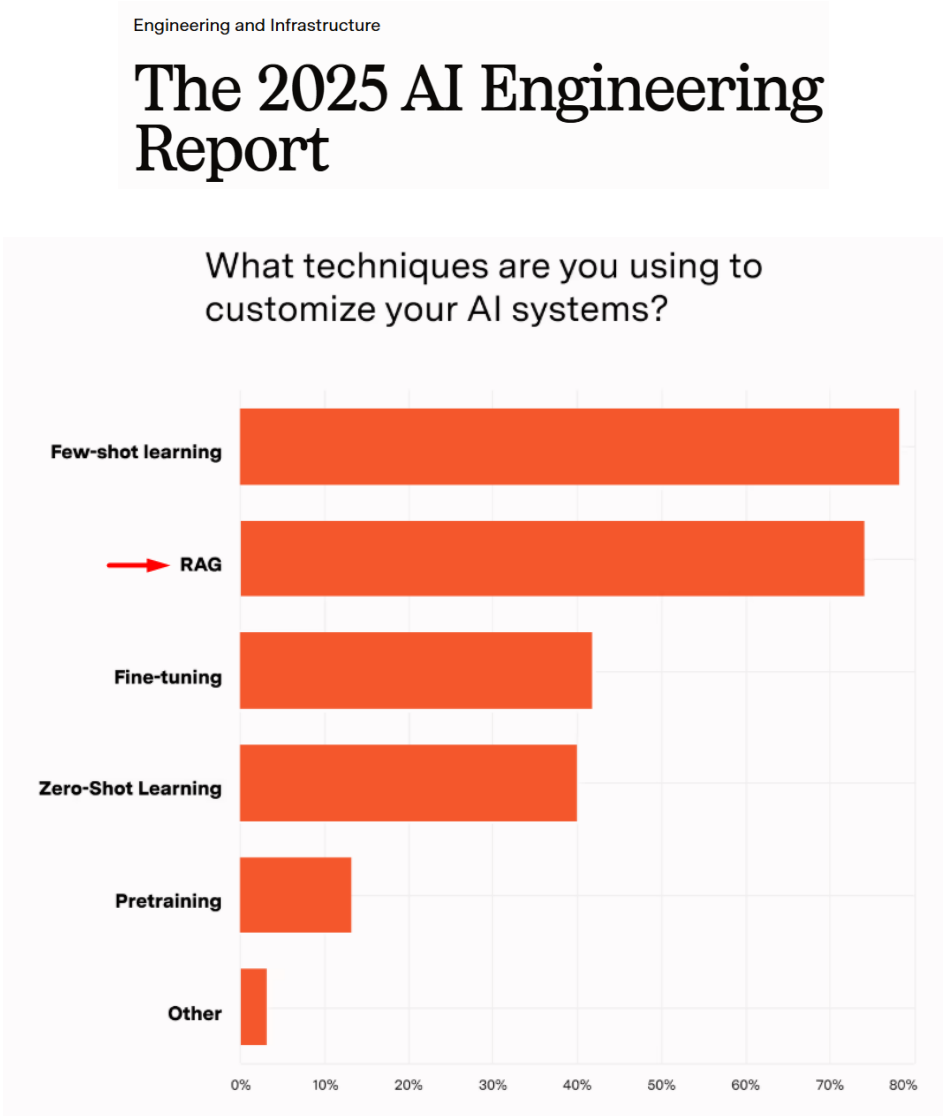
Итоговый проект на тему: «Мультимодальная RAG-система по отчету Сбера»

Акимов Дмитрий Андреевич

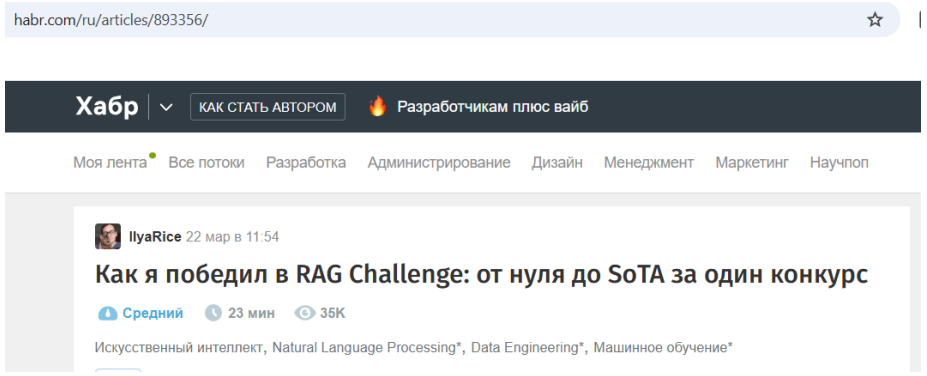
Шумов Александр Владимирович

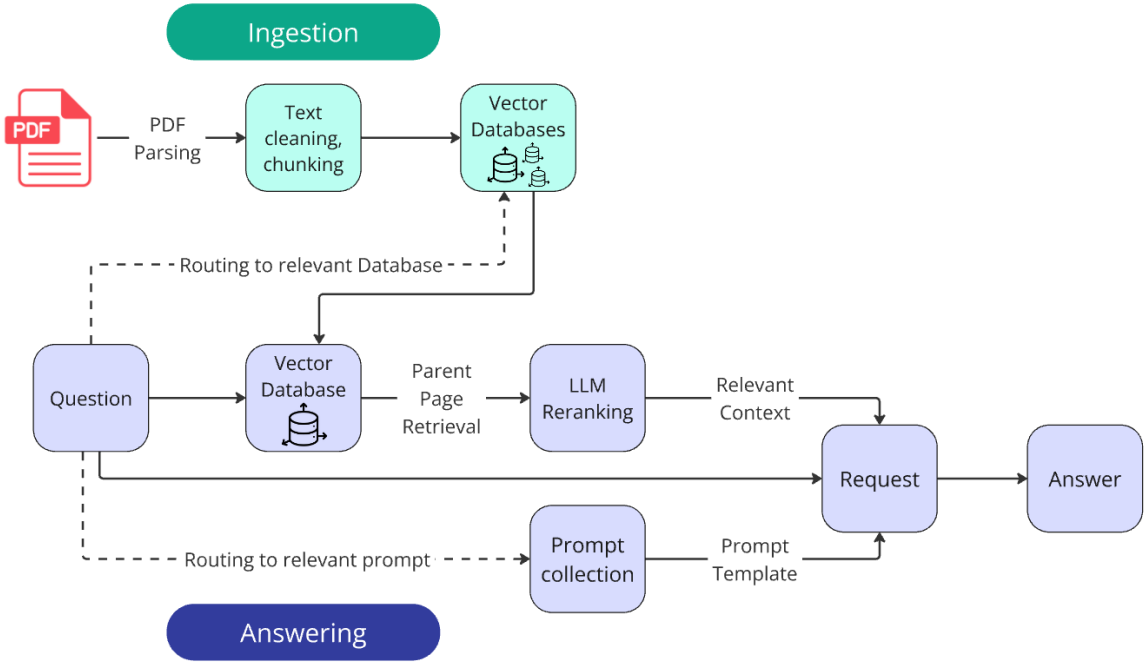
Актуальность темы и ее проблематика





State-of-the-art RAG техники





Конкурс на деньги (как и в случае с Kaggle) – часто лучший способ определить state-of-the-art техники.

В чём суть RAG Challenge?

Нужно создать вопросно-ответную систему на основе годовых отчётов компании. Если коротко, то в день конкурса:

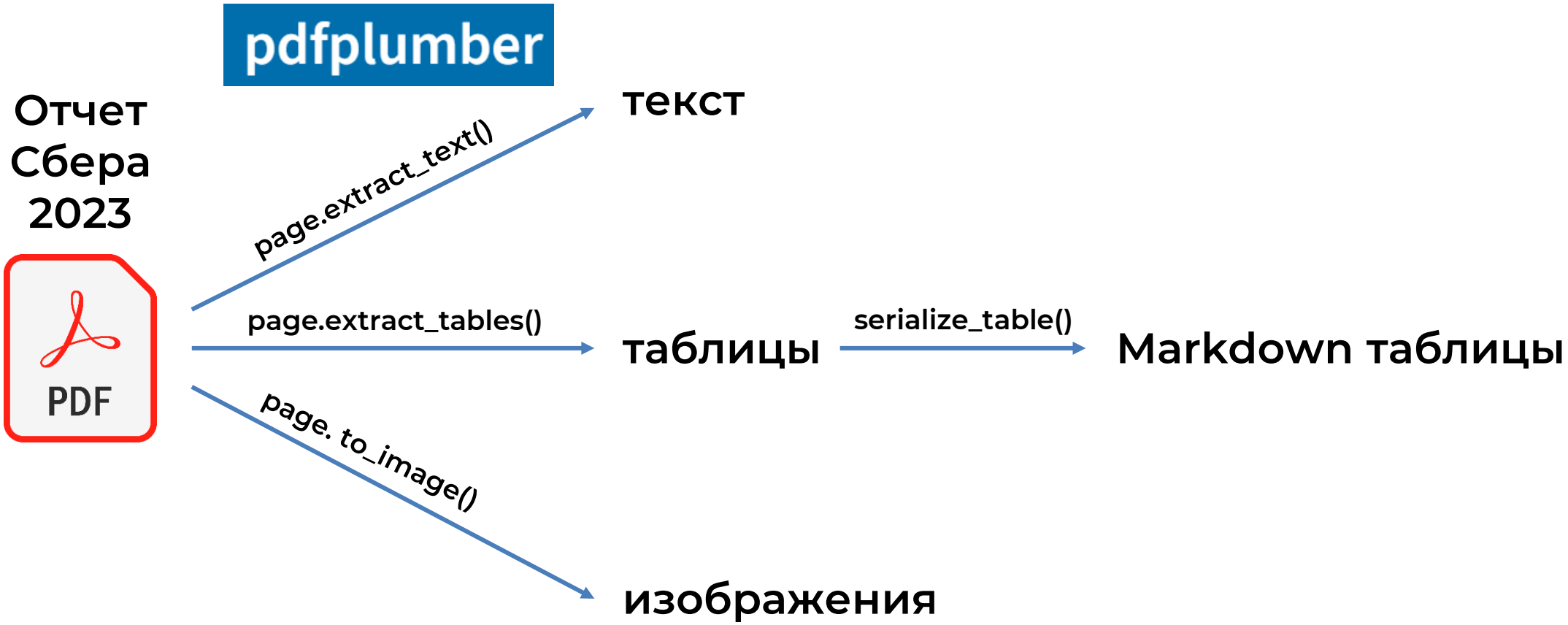
1. Выдаётся 100 годовых отчётов по случайно выбранным компаниям и 2.5 часа на их парсинг и составление базы данных. Отчёты представляют из себя PDF размером до 1000 страниц.
2. После этого генерируется 100 случайных вопросов (по заранее известным шаблонам), на которые система должна как можно быстрее ответить.

Все вопросы должны иметь однозначный ответ:

- Да/Нет;
- название компании (или нескольких компаний);
- названия управляющих позиций, выпущенных продуктов;
- размер той или иной метрики: выручка, количество магазинов и т.д.

Каждый ответ должен сопровождаться ссылками на страницы с ответом в качестве доказательства, что система по честному нашла ответ и не сгаллюцинировала.

Parsing



Победитель Enterprise RAG Challenge использовал Docling, арендовав сервер на Runpod – мы сделали так же, но долго было разбираться с конфликтом зависимостей, и вернулись на pdfplumber... А unstructured работал очень долго.

Ingestion

text-embedding-3-large

Текст → RecursiveCharacterTextSplitter(chunk_size=300, chunk_overlap=80) → Чанки текста по 300 символов с перекрытием по 80

Таблицы → Каждая таблица – один чанк

Изображения → GPT-4o → Описание изображения – один чанк

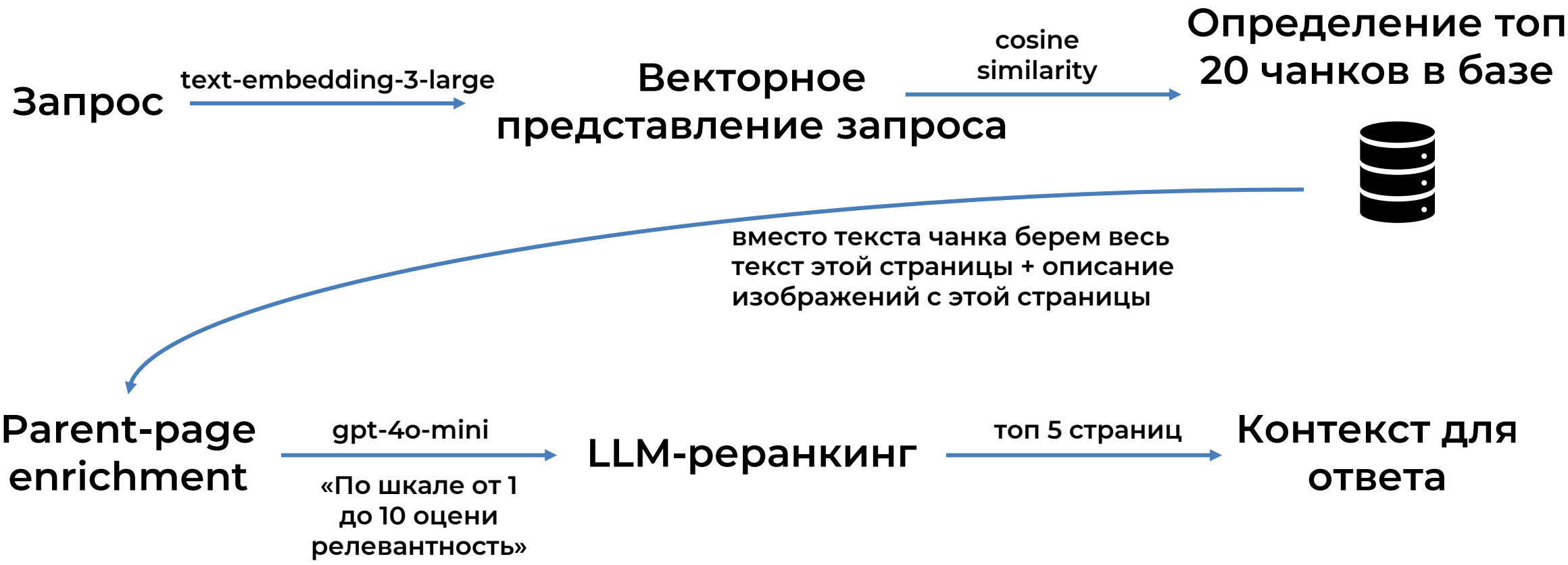
FAISS
(5388 векторов размерностью 3072)

+

metadata.json (что на какой странице + описание изображений)

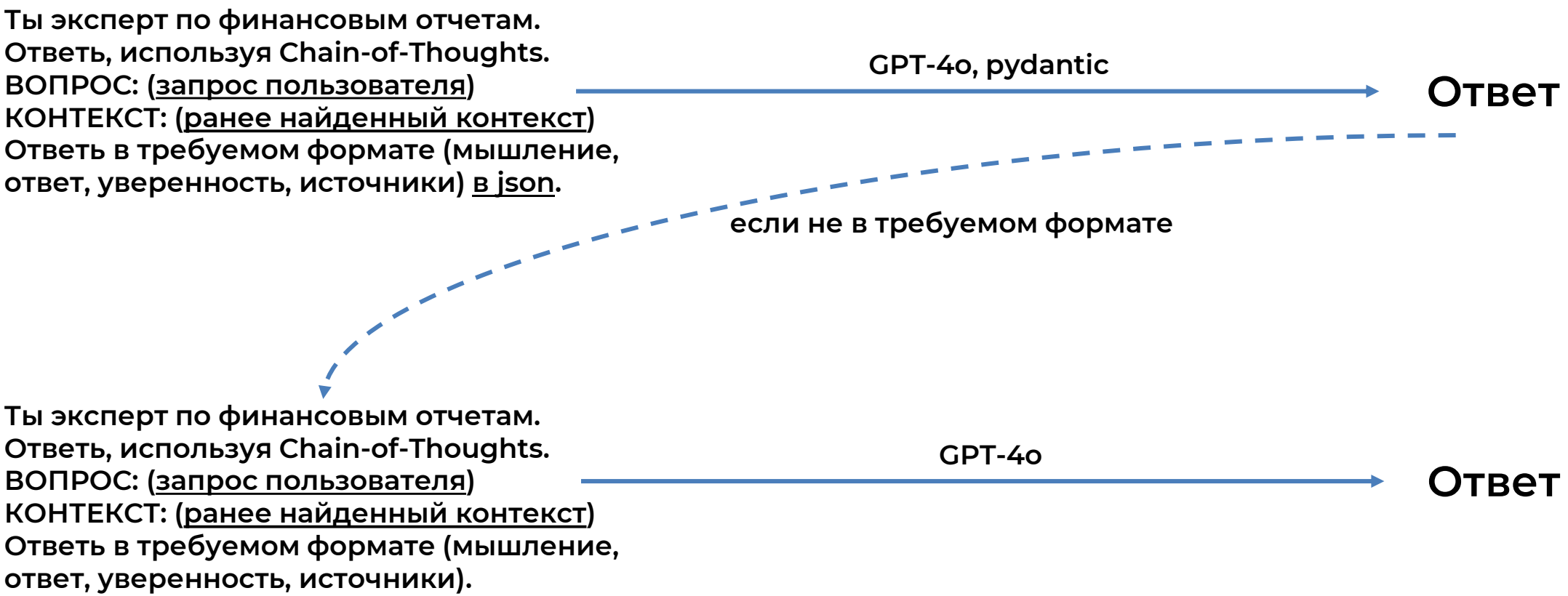
Победитель Enterprise RAG Challenge использовал тот же пайплайн, но делил на чанки текст вперемешку с таблицами и использовал перекрытие 50.

Retrieval

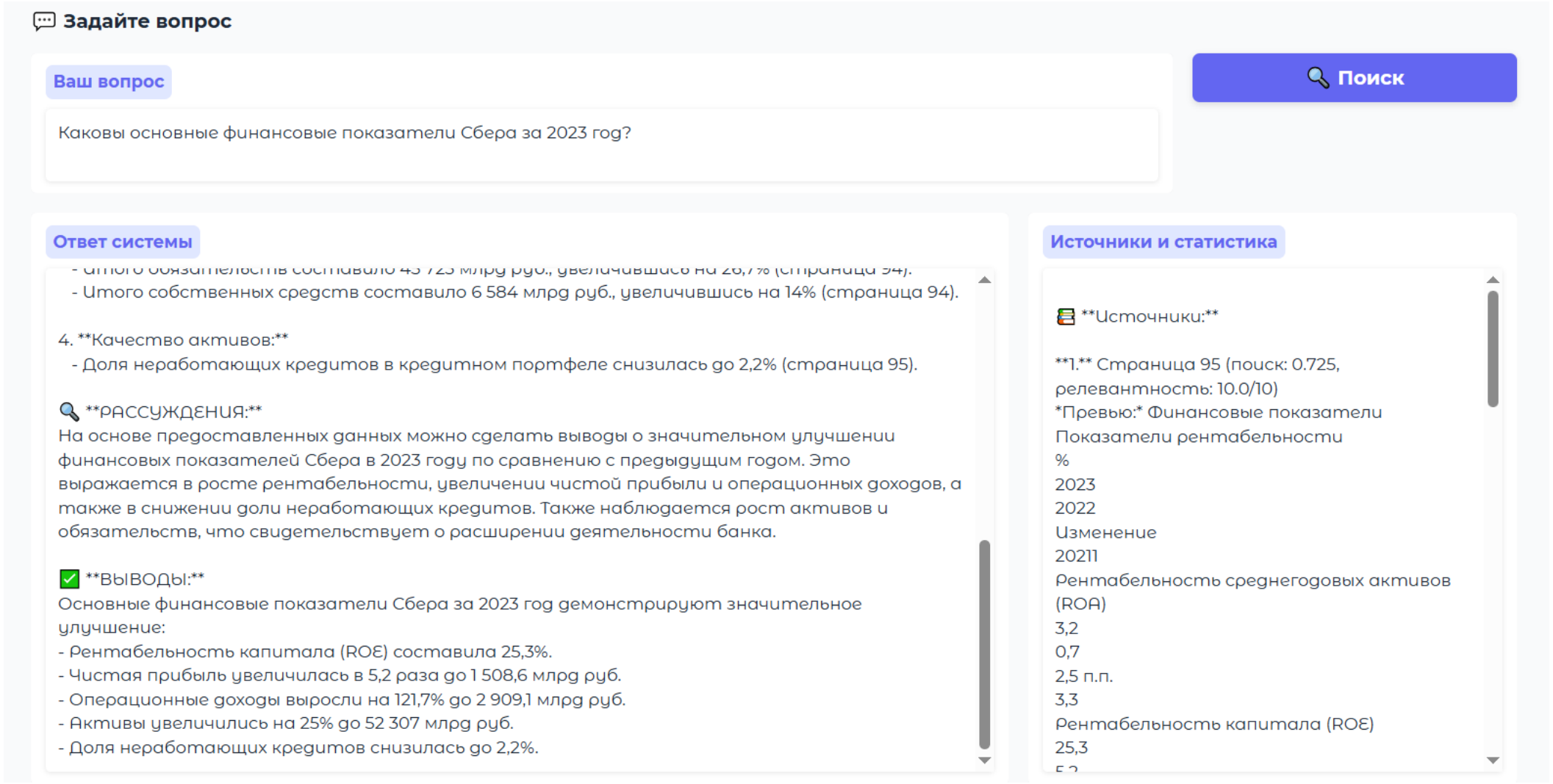


Победитель Enterprise RAG Challenge использовал тот же пайплайн, но брал топ 30 чанков / топ 10 страниц, поскольку у него было больше документов на вход.

Augmentation + Generation



Demo



Скриншот развернутой на HuggingFace мультимодальной RAG-системы

Выводы

- RAG – одно из основных применений LLM на сегодняшний день.
- В работе была реализована мультимодальная RAG-система, вдохновленная победным решением Enterprise RAG Challenge (2025 год).
- Основные используемые технологии: pdfplumber, RecursiveCharacterTextSplitter, FAISS, Parent-page enrichment, LLM reranking, Chain-of-Thoughts, pydantic, gradio, GPT-4o, text-embedding-3-large, gpt-4o-mini.
- Основные сложности: некачественное выделение и распознавание картинок (следует рассмотреть переход с pdfplumber на docling, а также уточнить промпт для GPT-4o и добавить автоматический повторный запрос в случае отказа создать описание картинки).
- Перспективы на будущее: целесообразно верифицировать качество собранного пайплайна путем построения автоматической оценки (RAGAS и др.).

Список используемых источников/программных средств:

- Google Colab, HuggingFace
- «Как я победил в RAG Challenge: от нуля до SoTA за один конкурс» (https://habr.com/ru/articles/893356/)