



# RAG-система с LLMs

Краснобаева Вера, Семенюк Полина,  
Васяткина Анастасия

# RAG (Retrieval-Augmented Generation)

- + **Retrieval (Поиск)** – сначала система ищет релевантные документы или фрагменты текста в базе знаний. Это помогает модели получать актуальную и точную информацию.
- + **Generation (Генерация)** – затем модель (например, LLM) использует найденную информацию для создания ответа, комбинируя её со своими знаниями.

# Цели и задачи

- + **Цель проекта:**
- + На основе представленного датасета и LLM реализовать RAG систему
- + **Задачи:**
- + Реализовать вариант модели Без retrieval части (базовая модель)
- + Реализовать вариант с retrieval частью (улучшенная модель) с использованием векторной базы данных
- + **Основная гипотеза:** использование векторной базы данных должно улучшить качество модели






# Датасет

- + датасет для оценки RAG систем, размер 50.4k
- + <https://huggingface.co/datasets/bearberry/sberquadqa>
- + **Содержит:**
  - + Вопрос
  - + Правильный ответ на него
  - + Тот же ответ, но все слова в приведены к начальной форме
  - + Контекст = текст, разделенный на чанки, из которого нужно достать ответ на вопрос

Split (1)

train · 50.4k rows

Search this dataset

id	question	answers	normalized_answers	context	metadata
string · lengths	string · lengths	sequence · lengths	sequence · lengths	list · lengths	dict
 10→11 <0.1%	 8→75 72.6%	 1→2 98.9%	 1→2 98.9%	 1→6 62.6%	
0_sberquad	чем представлены органические остатки?	[ "известковыми выделениями сине-..." ]	[ "известковый выделение синий..." ]	[ { "chunk": "В протерозойских..." } ]	{ "is_answerable": true }
1_sberquad	что найдено в кремнистых сланцах...	[ "нитевидные водоросли, грибные..." ]	[ "нитевидный водоросль грибной..." ]	[ { "chunk": "В протерозойских..." } ]	{ "is_answerable": true }
2_sberquad	что встречается в протерозойских отложениях?	[ "органические остатки" ]	[ "органический остаток" ]	[ { "chunk": "В протерозойских отложениях органические остатки встречаются намного чаще, чем в архейских.", "is_relevant": true }, { "chunk": "Они представлены известковыми выделениями сине-зеленых водорослей, ходами червей, остатками кишечнорастворимых.", "is_relevant": false }, { "chunk": "Кроме известковых водорослей, к числу древнейших растительных остатков..." } ]	{ "is_answerable": true, "tag": null }

## + Используемая модель:

+ RefalMachine/RuadaptQwen2.5-1.5B-instruct

+ Инструктивная версия адаптированной на русский язык модели Qwen2.5-1.5B. В модели был заменен токенизатор, затем произведено дообучение (Continued pretraining) на русскоязычном корпусе

+ **Qwen** — это семейство больших языковых моделей (**LLM**), разработанных компанией **Alibaba Cloud**. Они ориентированы на широкий спектр задач, включая генерацию текста, кодирование, диалоговые системы и RAG-приложения.

Плюсы: многоязычны, оптимизированы под RAG

# Qdrant

- + Qdrant — это векторная база данных с открытым исходным кодом, предназначенная для поиска ближайших соседей. Она часто используется в RAG-системах для хранения и поиска эмбеддингов текста, изображений и других данных.
- + Загружаем документы → Разбиваем на куски → Создаём эмбеддинги → Сохраняем в Qdrant.
- + Получаем запрос пользователя → Преобразуем в вектор → Ищем похожие вектора в Qdrant.
- + Берём найденные фрагменты → Отправляем в LLM → Генерируем осмысленный ответ.

## 2 варианта системы

- + Требуется реализовать 2 варианта системы на основе этого датасета:
- + Без retrieval части: считаем, что поиск уже произошел и найденные документы для каждого запроса находятся в поле “context”
- + С retrieval частью: **все (~50т)** списков в “context” объединяются в один корпус, где один из элементов контекста - один документ. Далее, они векторизуются с помощью подходящего энкодера текстов и сохраняются в векторную БД (qdrant или любая другая). Поверх БД реализуется подсистема поиска, которая принимает на вход запрос и возвращает список из topk релевантных документов.



# Другие инструменты

- + Используем библиотеку `sentence transformers` для получения эмбеддингов, которые будем хранить в `qdrant`
- + Оценим качество модели с помощью `transformers.data.metrics.squad_metrics.compute_f1`, предварительно приведя ответ к нормальной форме

# Ссылки

- + [Qdrant - Vector Database - Qdrant](#)
- + [RefalMachine/RuadaptQwen2.5-1.5B-instruct · Hugging Face](#)
- + [bearberry/sberquadqa · Datasets at Hugging Face](#)