

Проект по NLP / Project NLP

Исполнитель: Сучкова Мария

Executor: Suchkova Maria

Дата: Май 2025

Date: May 2025

Аннотация / Annotation

Мой проект посвящен созданию туристического путеводителя по 4 городам России – Владимир, Ярославль, Екатеринбург и Нижний Новгород с помощью метода NLP, а именно подхода RAG. Вот ссылка на код проекта, выложенного на гитхаб: [https://github.com/verycherrymary/tourist_rag].

My project is focused on creating a travel guide for 4 Russian cities—Vladimir, Yaroslavl, Yekaterinburg, and Nizhny Novgorod—using NLP methods, specifically the RAG (Retrieval-Augmented Generation) approach. Here is the link to the project code: [https://github.com/verycherrymary/tourist_rag].

1 Введение / Introduction

Часть данных взята с Национальной технологической олимпиады студентов по CV. Базовое решение предполагает использование мультимодального ru-CLIP. Но в данном проекте я хочу решить эту задачу творчески, используя NLP подход (RAG).

Part of the data was sourced from the National Technology Olympiad for Students in Computer Vision (CV). The baseline solution involves using a multimodal ru-CLIP model. However, in this project, I aim to solve the problem creatively by employing an NLP approach (RAG).

2 Описание моделей / Description of models

Идея подхода RAG (Retrieval-Augmented Generation) состоит в том, чтобы дать LLM доступ к внешним источникам информации при генерации ответа. По сути у нас есть две сущности: LLM и некоторая база знаний, к которой LLM будет обращаться для извлечения нужной информации перед генерацией итогового ответа.

The core idea of RAG is to provide the LLM with access to external knowledge sources during response generation. The system consists of two key components: the LLM itself and a knowledge base that the LLM queries to retrieve relevant information before generating final responses.

Процесс работы LLM+RAG в целом выглядит так:

- Когда пользователь задает промпт о достопримечательностях города, из него в каком-то виде выделяется информация, которая потом используется как ключ для запроса в базу данных (это наши данные по достопримечательностям для каждого города);
- По ключу из базы данных извлекаются N документов и отсортировываются по степени релевантности пользовательскому запросу;
- При генерации ответа LLM использует данные из запроса пользователя и информацию, полученную из базы знаний, то есть при ответе пользователь должен получить название достопримечательности в конкретном городе, ее координаты и изображения с этой достопримечательностью.

The LLM+RAG workflow operates as follows:

- Query Processing: When a user submits a prompt about city landmarks, key information is extracted to form search queries for the knowledge base (containing landmark data for each city).
- Document Retrieval: The system retrieves N documents and ranks them by relevance to the query.
- Response Generation: The LLM synthesizes information from both the user's query and retrieved knowledge base documents, ensuring responses include: Landmark names, Geographic coordinates, Associated images

Модели, используемые в проекте:

System Components:

База знаний (Knowledge base): наши данные обрабатываются и делятся на более мелкие части (чанки). Это позволяет более эффективно работать с текстом и улучшает качество извлечения информации. Я использовала методы разделения текста из фреймворка LangChain

Knowledge Base: Raw data is processed and segmented into smaller chunks using LangChain's text splitting methods, improving text processing efficiency and retrieval quality.

Модель векторизации (Embedding model): Каждая часть документа представляется в виде вектора. Эти эбеддинги затем хранятся в виде векторной базы данных для последующего поиска. Я использовала модель **"intfloat/multilingual-e5-large" - Multilingual-E5-large** — это крупная multilingual-модель для векторного представления текстов (text embedding), разработанная компанией Intel Labs (Intel's Artificial Intelligence team). Она предназначена для генерации семантически значимых векторных эмбеддингов текста на 100+ языках, включая русский. Модель основана на архитектуре Transformer (аналогично BERT, RoBERTa) и обучалась с использованием контрастивного обучения (contrastive learning) на разнородных данных.

Embedding Model: model: **"intfloat/multilingual-e5-large" - Multilingual-E5-large** , description: A multilingual text embedding model developed by Intel Labs, based on Transformer architecture (similar to BERT/RoBERTa). Supports 100+ languages including Russian, trained using contrastive learning on diverse datasets, generates semantically meaningful text embeddings

Векторная база данных (Vector database): База данных, которая хранит векторные представления текста таким образом, чтобы семантически близкие вектора индексировались близко (в одном кластере). В качестве метрики близости обычно используют косинусное сходство. В качестве векторной базы данных я использовала **FAISS (Facebook AI Similarity Search)** — это библиотека для эффективного поиска похожих векторов, разработанная исследователями из Facebook AI Research (FAIR). Она оптимизирована для быстрого поиска ближайших соседей (k-NN, nearest neighbor search) в больших наборах данных, особенно в задачах машинного обучения и обработки естественного языка (NLP).

Vector Database: Solution: FAISS (Facebook AI Similarity Search), Features: Optimized for efficient k-NN search in high-dimensional spaces, uses cosine similarity for semantic indexing, developed by Facebook AI Research for large-scale ML/NLP applications

Модель Retriever – обрабатывает пользовательский запрос, который может быть отформатирован или переформулирован для лучшего поиска. Я использовала ту же модель, что и для векторизации документов. С использованием векторной базы данных система находит K наиболее схожих документов по вектору пользовательского запроса. Это позволяет быстро извлекать релевантную информацию.

Retriever Model: Implementation: Same as the embedding model, Functionality: Processes/queries (with optional query reformulation), performs vector similarity search to retrieve top-K relevant documents

Модель LLM Reader анализирует извлечённый из базы данных контекст, чтобы сформировать ответ. В качестве reader я использовала и сравнивала работу несколько моделей

- модель мультиязычная квантизованная версия **Qwen/Qwen2.5-3B-Instruct** - это 3-миллиардная версия инструктируемой языковой модели от Alibaba Group (серия Qwen, ранее известная как Tongyi Qianwen).
- модель русскоязычная **YandexGPT-5-Lite-Instruct** -Instruct-версия большой языковой модели YandexGPT 5 Lite на 8 миллиардов параметров с длиной контекста 32k токенов.

LLM Reader Models Tested:

- Qwen/Qwen2.5-3B-Instruct: 3B-parameter multilingual model (Alibaba Group), Quantized version for efficiency
- YandexGPT-5-Lite-Instruct: 8B-parameter Russian-optimized model, 32k token context window

Также я использовала идею в RAG - извлечение большего количества документов, чем нужно в итоге, а затем реранжирование результатов с помощью более мощной модели. В качестве **модели Reranker** я выбрала "colbert-ir/colbertv2.0" - это улучшенная версия модели **CoBERT (Contextualized Late Interaction BERT)**, разработанной для эффективного семантического поиска и ранжирования документов. Она сочетает преимущества плотных (dense) и разреженных (sparse) эмбедингов, обеспечивая высокую точность при низких вычислительных затратах.

Reranker Model -Solution: colbert-ir/colbertv2.0, Advantages: enhanced version of CoBERT (Contextualized Late Interaction BERT), combines dense and sparse embedding benefits, delivers high accuracy with low computational overheadl. mplementation: initial retrieval of extra documents, precision reranking using CoBERTv2

3 Данные / Dataset

Данные представляют из себя таблицу, в которой по столбцам содержится информация о памятниках, архитектуре и других значимых местах для 4 городов. В данных есть столбцы:

- Name - название достопримечательности;
- City - город расположения;
- Lon, Lat - координаты долготы и широты;
- description - описание (извлеченное из WikiData);
- image - изображение в формате base64;

The data is organized in a tabular format containing information about monuments, architectural landmarks, and other significant locations across 4 cities. The columns include:

- Name - Landmark name
- City - City where the landmark is located
- Lon, Lat - Longitude and latitude coordinates
- Description - Textual description (sourced from WikiData)
- Image - Base64-encoded image data

Вот пример данных, см.ниже.

See below for a data sample:

	Name	City	Lon	Lat	description
0	Александро-Невская часовня	Екатеринбург	60.602776	56.828709	часовня в Екатеринбурге
1	Александро-Невский собор	Екатеринбург	60.598534	56.822884	православный собор в Екатеринбурге
2	Белая башня	Екатеринбург	60.572338	56.893166	сооружение в Екатеринбурге
3	Водонапорная башня	Екатеринбург	60.605469	56.837891	сооружение в Екатеринбурге
4	Второй Дом Советов	Екатеринбург	60.596550	56.840122	Второй Дом Советов (Екатеринбург)
...
361	церковь Иоанна Предтечи в Толчкове	Ярославль	39.856861	57.610909	Православный храм в Ярославле
362	церковь Михаила Архангела	Ярославль	39.892590	57.622082	Православный храм в Ярославле
363	церковь Николая Чудотворца	Ярославль	39.870548	57.620068	Православный храм в Ярославле
364	церковь Николы Рублёного	Ярославль	39.898258	57.622509	храм в Ярославле
365	часовня Казанской Богоматери	Ярославль	39.890251	57.620323	Часовня Казанской Богоматери (Ярославль)

366 rows × 5 columns



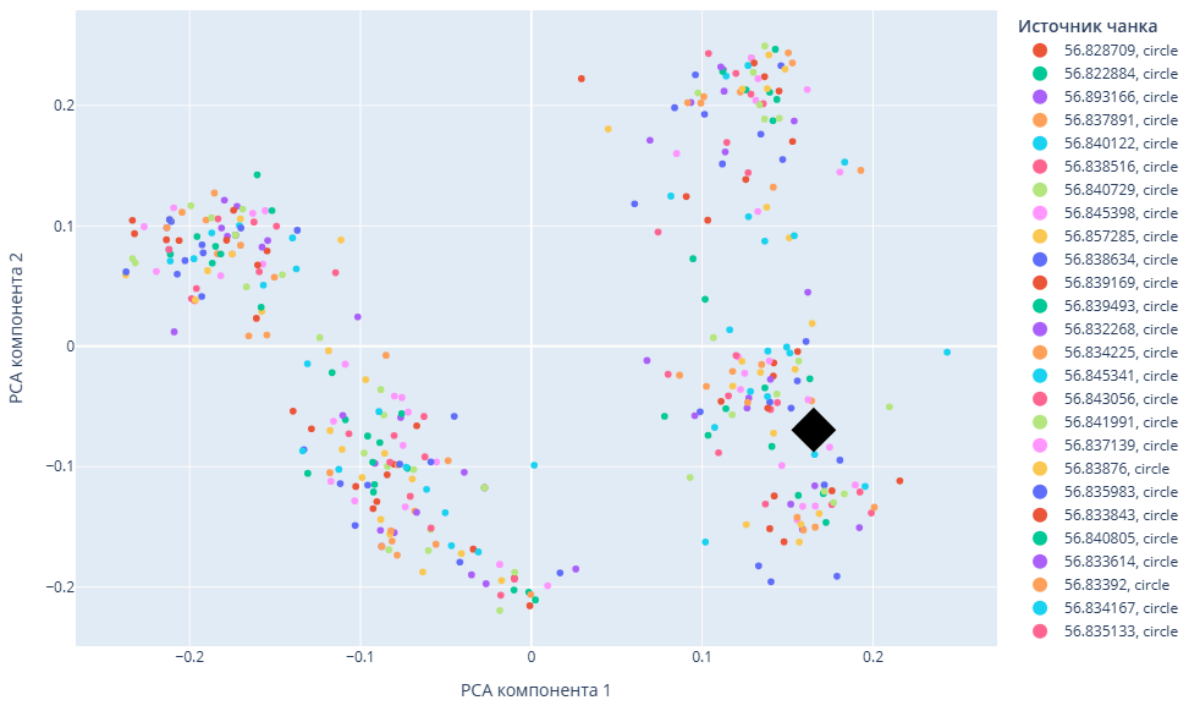
Получился очищенный датасет из 366 уникальных достопримечательностей, разбитых на 4 города, где фотографии сконкатенированы по достопримечательностям, так же как и описание. Было очень много выбросов и пропусков, нерелевантных фотографий и названий достопримечательностей, особенно в г. Владимир

The final cleaned dataset contains 366 unique landmarks distributed across 4 cities, with concatenated images and descriptions for each landmark. Significant data cleaning was required to address: numerous outliers and missing values, irrelevant/non-matching photographs, inconsistent landmark naming (particularly severe in Vladimir)

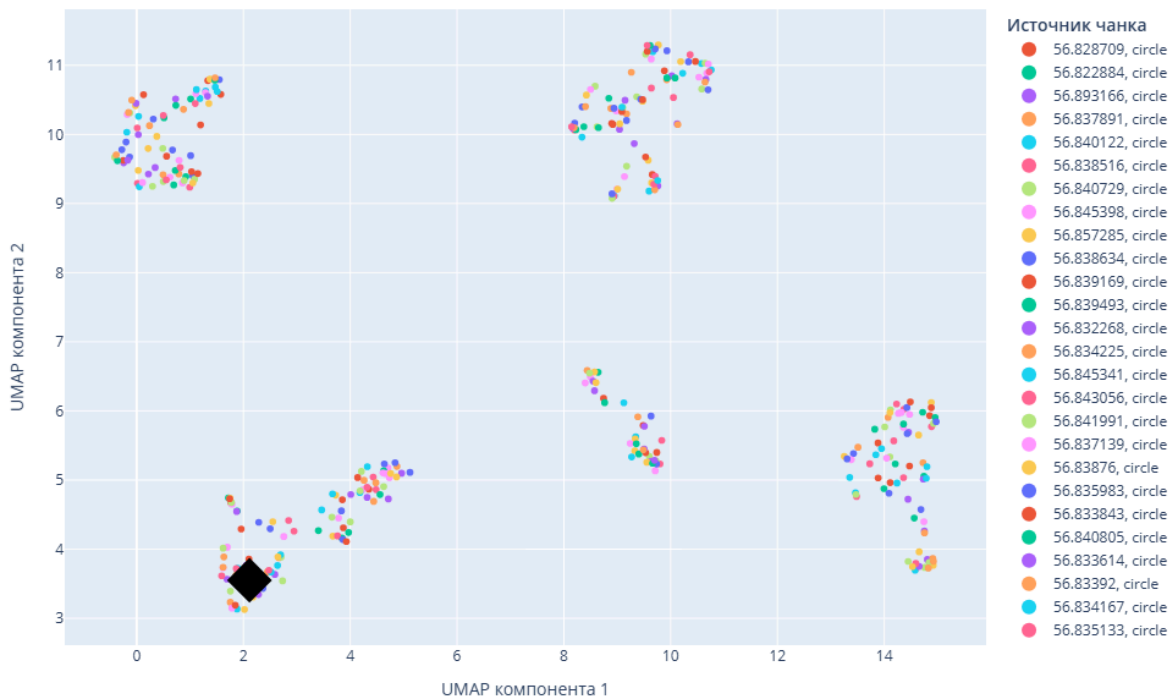
После векторизации достопримечательностей из таблицы и создания векторной базы данных можно посмотреть на **визуализацию эмбедингов** при помощи метода главных компонент **(PCA)** и **Uniform Manifold Approximation and Projection (UMAP)**.

After vectorizing the landmark data and constructing the vector database, we performed dimensionality reduction to visualize the embeddings using: Principal Component Analysis (PCA), Uniform Manifold Approximation and Projection (UMAP)

2D проекция методом PCA



2D проекция методом UMAP



Поскольку эмбединги отражают смысл документа, их близость по смыслу должна тоже коррелировать. В принципе прослеживается кучкование по городам.

Также показан эмбединг пользовательского запроса (черный ромб): мы ищем k документов с наиболее близким значением, поэтому выбираем k ближайших векторов, этот вектор в правильном месте, где город Ярославль. User query был:

- "Какие достопримечательности в городе Ярославль?"

Since embeddings capture semantic meaning, their spatial proximity should correlate with conceptual similarity. Indeed, we observe clustering by city in the vector space.

The visualization also includes the user query embedding (black diamond). For retrieval: We search for k documents with the closest embeddings, select k nearest neighbor vectors, the query vector is correctly positioned within the Yaroslavl cluster. User query:

- "What are the notable landmarks in Yaroslavl?"

5 Эксперименты / Experiments

Эксперименты заключались в использовании нескольких моделей Reader LLM и подсчета метрик RAGAS

The study involved testing multiple LLM Reader models and evaluating their performance using RAGAS metrics.

Сначала посмотрим на ответы моделей Яндекса и Qwen с помощью RAG

- вопрос был **"Какие достопримечательности в Ярославле?:"**
- ответы ограничивались 10 используемыми документами

We first compared responses from Yandex and Qwen models within the RAG framework:

- Query: **"What are the notable landmarks in Yaroslavl?:"**
- Retrieval constraints: Responses were limited to 10 retrieved documents

Ответ модели Qwen 2.5 / Answer model Qwen 2.5

=====Answer=====


Храм Владимирской иконы Божией Матери, Российский академический театр драмы им. Ф. Волкова, храм Ярославской иконы Божией Матери, Тимерёвский археологический комплекс, памятник жертвам белогвардейского мятежа, Ярославский государственный театр кукол, Музей занимательных наук Эйнштейна, Губернаторский дом (Ярославский художественный музей), Ярославское художественное училище, Спасо-Преображенский монастырь.

=====Source docs=====

Document 0-----

Ярославль | Музей занимательных наук Эйнштейна | Музей занимательных наук Эйнштейна


Широта: 57.670315, Долгота: 39.8381



Document 1-----

Ярославль | Тимерёвский археологический комплекс | Тимерёвский археологический комплекс

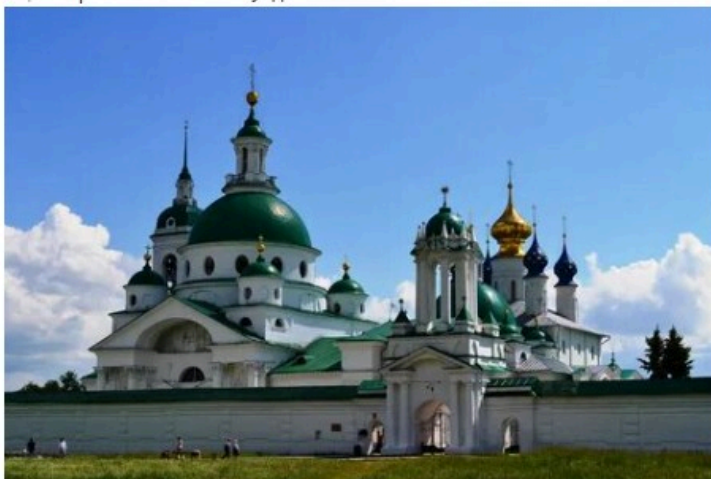
Широта: 57.53627, Долгота: 39.741951



Document 2-----

Ярославль | Спасо-Преображенский монастырь | монастырь в Ярославле

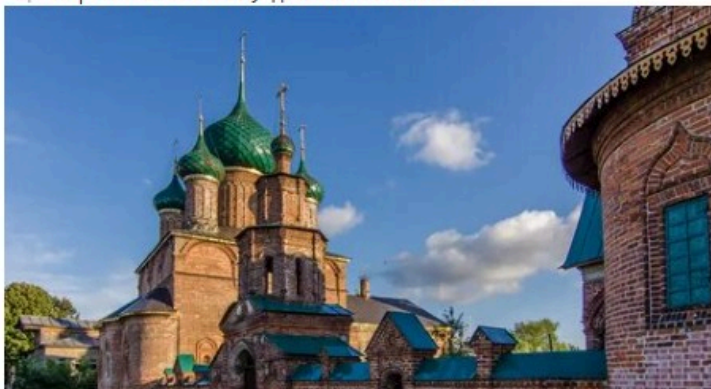
Широта: 57.621704, Долгота: 39.889442



Document 3-----

Ярославль | Церковь Владимирской иконы Божией Матери в Ярославле | Архитектурный ансамбль в Ярославле

Широта: 57.611237, Долгота: 39.906563



Document 4-----

Ярославль | Ярославское художественное училище | Ярославское художественное училище

Широта: 57.609852, Долгота: 39.876846



Document 5-----

Ярославль | Ярославский художественный музей (Губернаторский дом) | художественный музей в Ярославле

Широта: 57.628376, Долгота: 39.897274



Document 6-----
Ярославль | Театр кукол | Ярославский государственный театр кукол

Широта: 57.624577, Долгота: 39.876846



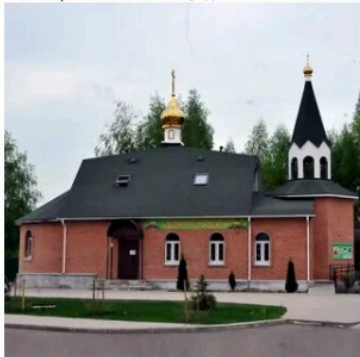
Document 7-----
Ярославль | Памятник жертвам белогвардейского мятежа | Памятник жертвам белогвардейского мятежа

Широта: 57.625862, Долгота: 39.89571



Document 8-----
Ярославль | храм Ярославской иконы Божией Матери | Храм в честь Ярославской иконы Божией Матери

Широта: 57.582783, Долгота: 39.908268



Document 9-----
Ярославль | Российский государственный академический театр драмы имени Федора Волкова | Российский академический театр драмы им. Ф. Волкова

Широта: 57.627155, Долгота: 39.884796



Ответ модели Yandex / Answer model Yandex

=====
Достопримечательности Ярославля

В Ярославле есть множество интересных мест и памятников культуры и истории. Вот некоторые из них:

1. **«Церковь Владимирской иконы Божией Матери»** - архитектурный ансамбль, который является одной из культурных жемчужин города.
2. **«Росийский государственный академический театр драмы имени Федора Волкова»** - один из старейших театров России, имеющий большое значение для театральной жизни страны.
3. **«Храм Ярославской иконы Божией Матери»** - посвящен Ярославской иконе Божией Матери, имеет важное религиозное значение.
4. **«Тимирязевский археологический комплекс»** - место, где можно узнать о древней истории региона.
5. **«Памятник жертвам белогвардейского мятежа»** - напоминает о трагических событиях прошлого.
6. **«Ярославский государственный театр кукол»** - уникальное культурное учреждение, предлагающее спектакли для детей и взрослых.
7. **«Музей занимательных наук Эйнштейна»** - интересное место для посещения с детьми, где можно узнать много нового о науке через интерактивные экспонаты.
8. **«Ярославский художественный музей (Губернаторский дом)»** - представляет собой коллекцию произведений искусства разных эпох.
9. **«Ярославское художественное училище»** - учебное заведение, готовящее специалистов в области изобразительного искусства.
10. **«Спасо-Преображенский монастырь»** - исторический монастырский комплекс, являющийся важной частью культурного наследия Ярославля.

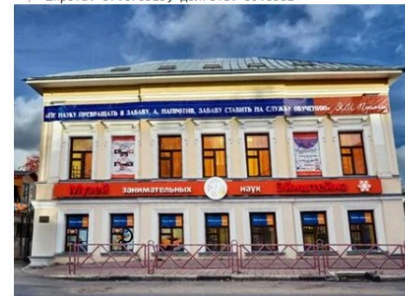
Эти места отражают разнообразие культурной жизни Ярославля и его историческое наследие.

```
=====Source docs=====
```

Document 0-----

Ярославль | Музей занимательных наук Эйнштейна | Музей занимательных наук Эйнштейна

Широта: 57.670315, Долгота: 39.8381



Document 1-----

Ярославль | Тимерёвский археологический комплекс | Тимерёвский археологический комплекс

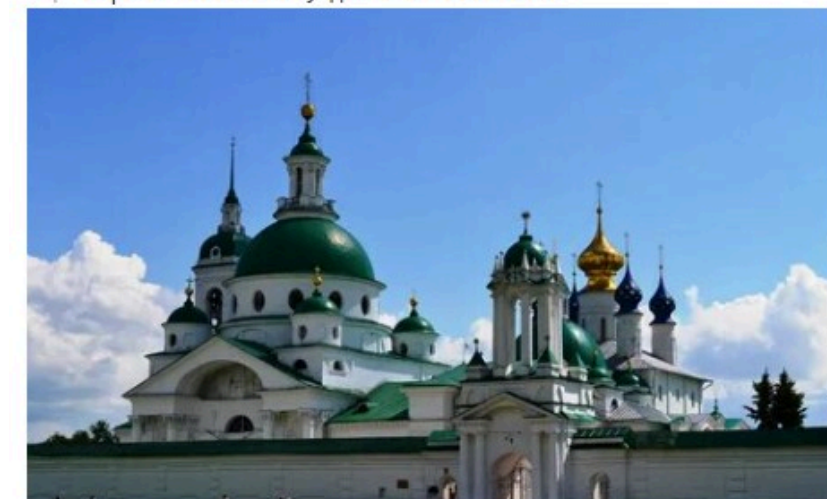
Широта: 57.53627, Долгота: 39.741951



Document 2-----

Ярославль | Спасо-Преображенский монастырь | монастырь в Ярославле

Широта: 57.621704, Долгота: 39.889442



Document 3-----

Ярославль | Церковь Владимирской иконы Божией Матери в Ярославле | Архитектурный ансамбль в Ярославле

Широта: 57.611237, Долгота: 39.906563



Document 4-----

Ярославль | Ярославское художественное училище | Ярославское художественное училище

Широта: 57.609852, Долгота: 39.876846



Document 5-----

Ярославль | Ярославский художественный музей (Губернаторский дом) | художественный музей в Ярославле

Широта: 57.628376, Долгота: 39.897274



Document 6-----

Ярославль | Театр кукол | Ярославский государственный театр кукол

Широта: 57.624577, Долгота: 39.876846



Document 7-----

Ярославль | Памятник жертвам белогвардейского мятежа | Памятник жертвам белогвардейского мятежа

Широта: 57.625862, Долгота: 39.89571



Document 8-----

Ярославль | храм Ярославской иконы Божией Матери | Храм в честь Ярославской иконы Божией Матери

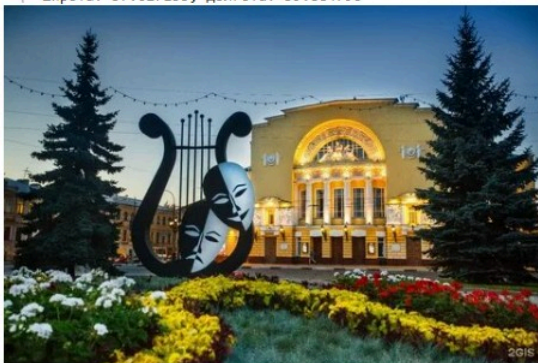
Широта: 57.582783, Долгота: 39.908268



Document 9-----

Ярославль | Российский государственный академический театр драмы имени Федора Волкова | Российский академический театр драмы им. Ф. Волкова

Широта: 57.627155, Долгота: 39.884796



В целом ответы моделей Qwen и Яндекса похожи, все таки модель reranker у них одна и та же. Русскоязычная Яндекс модель мне показалось более развернутый ответ дала, и это понятно, потому что она имеет больше параметров, чем модель Qwen.

Overall, the responses from Qwen and Yandex models are quite similar, which is expected since they use the same reranker model. However, the Russian-language Yandex model appeared to provide more detailed answers in my evaluation, because model has more parameters than model Qwen

5.1 Метрики / Metrics

RAGAS - это система, созданная для автоматической оценки эффективности модели RAG. Она представляет собой комплексный инструмент, способный оценить как качество извлечения информации, так и качество созданного текста. Это достигается благодаря использованию различных метрик, которые анализируют разные аспекты ответов, сгенерированных языковой моделью.

RAGAS (Retrieval-Augmented Generation Assessment System) is a specialized framework for automated evaluation of RAG model performance. This comprehensive tool assesses: Retrieval Quality - Accuracy of information extraction from knowledge sources, Generation Quality - Coherence and relevance of LLM-generated responses. The system employs multiple metrics to analyze different dimensions of model outputs, providing holistic performance insights.

В своем проекте я оценивала эффективность модели RAG с помощью метрик Answer Relevancy, Context Precision, Faithfulness.

In this project, I assessed the RAG model's effectiveness using three key metrics Answer Relevancy, Context Precision, Faithfulness

Faithfulness

Оценивает, насколько ответы языковой модели верно передают информацию, содержащуюся в источниках данных. Это критически важно, чтобы гарантировать, что модель не искажает или не меняет информацию при её интерпретации.

Evaluates how well the answer of the LLM model accurately convey the information contained in the context. This is critically important to ensure that the model does not distort or change the information when interpreting it.

Faithfulness = (Number of claims that can be inferred from given context) / (Total number of claims in the generated answer)

Answer Relevancy

Демонстрирует, насколько ответы соответствуют заданным вопросам. Это важный аспект, так как даже правильный ответ с технической точки зрения может быть ненужным, если он не отвечает на конкретный вопрос пользователя.

Answer Relevancy - Measures how well the generated answers match the user's intent

answer relevancy = (1 / N) * sum_{i=1}^N cos(E_{g_i}, E_o)

answer relevancy = (1 / N) * sum_{i=1}^N (E_{g_i} · E_o / (||E_{g_i}|| * ||E_o||))

Где:

- E_{g_i} - это эмбеddинг сгенерированного ответа i
- E_o - это эмбеddинг исходного вопроса.
- N - это количество сгенерированных вопросов.

Context Recall и Context Precision

Эти метрики измеряют эффективность модели в учете контекста вопроса для формирования ответа. "Context Recall" оценивает количество использованной информации из контекста в ответе, в то время как "Context Precision" оценивает точность и релевантность использования контекстной информации.

- Context Recall - Evaluates whether all relevant information was retrieved from the knowledge base
- Context Precision - Analyzes the proportion of retrieved documents that were actually relevant to the query

context recall = (GT sentences that can be attributed to context) / (Number of sentences in GT)

ContextPrecision@k = (sum precision@k) / (total number of relevant items in the top K results)

Precision@k = (true positives@k) / (true positives@k + false positives@k)

У меня нет API ключа от OpenAI, поэтому воспользоваться готовыми метриками библиотеки RAGAS я не могла, делала свои метрики на основе существующих.

I didn't have an OpenAI API key, so I couldn't use the pre-built metrics from the RAGAS library. Instead, I implemented custom evaluation metrics based on existing ones.

5.2 Настройка эксперимента / Experiment Setup

Было подготовлено 2 тестовых набора данных, на которых потом измерялись метрики качества

- test_set_qwen - собраны ответы с RAG моделью Qwen
- test_set_yandex - собраны ответы с RAG моделью Яндекс

Two test datasets were prepared for metric evaluation:

- test_set_qwen: Contains responses generated by the Qwen-based RAG model
- test_set_yandex: Contains responses from the Yandex-based RAG model

Посмотрим пример ответа модели Qwen

See below example of answer model Qwen

Question: Каково значение исторического места, которое находится в центре Екатеринбурга - Александро-Невский собор?
Answer_RAG: Определение исторического места, которое находится в центре Екатеринбурга - Александро-Невский собор, можно найти в первом из приведенных документов.

Значение этого места заключается в том, что оно является православным собором в Екатеринбурге. Это значительное место для православной веры и истории города.

Чтобы получить точные координаты, обратимся к информации из второго документа:

Широта: 56.822884
Долгота: 60.598534

Таким образом, Александро-Невский собор расположен в Екатеринбурге, являясь важным историческим местом в центре города. Ответ дан не из контекста.
Context: Александро-Невский собор Екатеринбург 60.598534
56.822884 православный собор в Екатеринбурге

Посмотрим пример ответа модели от Яндекса

See below example of answer model from Yandex

Question: Каково значение исторического места, которое находится в центре Екатеринбурга - Александро-Невский собор?
Answer_RAG: Александро-Невский собор является историческим местом в центре Екатеринбурга. Это православный собор, который имеет большое культурное и религиозное значение для жителей города и всей России. Соборы такого типа часто являются центрами религиозной жизни, местами паломничества и важными архитектурными памятниками. В контексте истории России соборы играли ключевую роль как центры духовной жизни народа, а также служили символами веры и национального единства.
Context: Извлеченные документы с координатами:
Документ 0:::
Содержание: Екатеринбург | Александро-Невский собор | православный собор в Екатеринбурге
Координаты: широта 56.822884, долгота 60.598534

Документ 1:::
Содержание: Екатеринбург | Александро-Невская часовня | часовня в Екатеринбурге
Координаты: широта 56.828709, долгота 60.602776

Документ 2:::
Содержание: Нижний Новгород | Собор Александра Невского | православный собор в Нижнем Новгороде
Координаты: широта 56.333679, долгота 43.971439

Всего было 252 сгенерированных вопросов, на которые отвечали 2 модели. На основе этих тестовых наборов замерялись метрики качества

- question - вопрос
- answer - ответ модели с RAG
- ground_truth - эталонный ответ
- context - данные из таблицы с достопримечательностями

A total of 252 generated questions were answered by both models. The test datasets used for quality assessment contained:

- question: The input query
- answer: Model's RAG-generated response
- ground_truth: Reference answer (gold standard)
- context: Landmark data from the knowledge base

Более подробно про тестирование моделей и замеры метрик можно посмотреть в jupyter ноутбуке Suchkova_project_rag.ipynb по ссылке на моем гугл диске

<https://colab.research.google.com/drive/1ICNuyUY7Jat51qPI6F-64klrU9gU29N?usp=sharing>
(<https://colab.research.google.com/drive/1ICNuyUY7Jat51qPI6F-64klrU9gU29N?usp=sharing>).

For more detailed insights into testing models and measuring their metrics, you can refer to the Jupyter notebook titled Suchkova_project_rag.ipynb, available at my Google Drive link

<https://colab.research.google.com/drive/1ICNuyUY7Jat51qPI6F-64klrU9gU29N?usp=sharing>
(<https://colab.research.google.com/drive/1ICNuyUY7Jat51qPI6F-64klrU9gU29N?usp=sharing>).

6 Результаты / Results

Модель мультиязычная квантизованная версия Qwen/Qwen2.5-3B-Instruct показала метрики:

- **Answer Relevance модели Qwen - 0,8933** - очень хороший результат, то есть вектор ответа модели с помощью Rag смотрит в одном направлении с вектором вопроса, сгенерированным моделью на основе контекста, что говорит нам о том, что ответ, данный с помощью RAG, соответствует вопросу
- **Context Precision Qwen: 1.00** - хорошая контекстная точность, то есть в данном моделью Qwen ответе RAG полностью используется информация из контекста
- **Faithfulness: 0.925** - хорошее значение метрики, значит модель Qwen при ответе с RAG не так много галлюцинирует.

The multilingual quantized version of the Qwen/Qwen2.5-3B-Instruct model demonstrated metrics as follows:

- **Answer Relevance:** The answer relevance metric for the Qwen model is 0.8933, which is an excellent result. This means that the vector of the model's response aligns well with the question vector generated by the model based on context when using Retrieval Augmented Generation (RAG). It indicates that the answer provided through RAG corresponds to the question.
- **Context Precision:** The Context Precision metric for Qwen is 1.00, indicating good contextual accuracy. This suggests that in answers generated by the Qwen model using RAG, all information comes directly from the given context.
- **Faithfulness:** The Faithfulness metric value is 0.925, a strong indicator that the Qwen model does not hallucinate much while answering questions augmented with RAG.

Модель русскоязычная YandexGPT-5-Lite-Instruct показала метрики:

- **Answer Relevancy модели от Яндекса - 0,9075** - чуть получше результат по сравнению с Qwen, то есть вектор ответа модели с помощью Rag смотрит в одном направлении с вектором вопроса, сгенерированным моделью на основе контекста, что говорит нам о том, что ответ, данный с помощью RAG, соответствует вопросу
- **Context Precision Yandex: 1.00** - хорошая контекстная точность, то есть в данной модели Yandex ответе RAG полностью используется информация из контекста
- **Faithfulness: 0.944** - у модели Yandex метрика получилась чуть получше, все таки русскоязычная модель

The Russian-language YandexGPT-5-Lite-Instruct model exhibited the following metrics:

- **Answer Relevancy:** The Answer Relevancy metric for the Yandex model is 0.9075, slightly better than Qwen's performance. This implies that the model's response vector and the question vector generated by the model based on context are aligned in the same direction when utilizing Retrieval Augmented Generation (RAG), ensuring that the RAG-generated answer matches the question.
- **Context Precision:** The Context Precision metric for Yandex is 1.00, signifying high contextual accuracy. All information used in responses generated by the Yandex model via RAG originates entirely from the provided context.
- **Faithfulness:** The Faithfulness metric achieved by the Yandex model is 0.944, marginally superior to Qwen. As this is a Russian-specific language model, it appears to have fewer instances of hallucination during RAG-augmented responses.

Выводы:

- по таким метрикам RAGAS, как answer relevancy, context precision и faithfulness, модель Vikhrmodels/Vikhr-YandexGPT-5-Lite-8B-it от Яндекса превосходит модель Qwen/Qwen2.5-3B-Instruct;
- лучшие метрики модели Яндекс скорее всего связаны с тем, что она имеет больше параметров (8 миллиардов), все таки русскоязычная, и больше было данных на русском языке, на которых она обучалась;
- у модели Яндекс есть недостатки с тем, что непонятно как загружать квантизованную версию (так она весит где-то 12-13гб без квантизации), и не поддерживается в явном виде через pipeline transformers батчевая обработка данных;
- модель Qwen хоть и с меньшим количеством параметров (3 миллиарда), а значит меньше памяти занимает, и легкая настройка для обработки батчами, показала ненамного хуже метрики RAGAS;
- для полноценного приложения (с большим количеством оперативной памяти и использованием видеокарт GPU) я бы все-таки использовала модель от Яндекса, так как и по метрикам, и по-моему человеческому мнению модель Яндекс более развернуто и адекватно отвечает на вопросы.

Findings:

- According to such metrics as answer_relevancy, context_precision and faithfulness in RAGAS framework, the Russian-language model Vikhrmodels/Vikhr-YandexGPT-5-Lite-8B-it from Yandex outperforms the Chinese language model Qwen/Qwen2.5-3B-Instruct.
- The superior performance of the Yandex model is likely due to its larger parameter count (8 billion parameters), being specifically designed for Russian language tasks, and having more training data in Russian.
- One drawback of the Yandex model is that it's unclear how to load a quantized version (the unquantized model weighs around 12–13 GB). Additionally, batch processing through Transformers' pipelines isn't explicitly supported.
- Despite fewer parameters (only 3 billion), thus requiring less memory and easier setup for batch processing, the Qwen model showed only slightly worse results on RAGAS metrics compared to the Yandex model.
- For a full-fledged application using significant amounts of RAM and GPUs, I would still opt for the Yandex model because both objective metrics and my subjective judgment indicate that it provides more comprehensive and coherent responses.

Вызовы, с которыми я столкнулась при проектировании системы RAG были следующими:

- с первым с чем столкнулась - конфликт версий библиотек transformers, langchain, bitsandbytes, ragatouille на сайте Kaggle - решение: фиксация версий библиотек, чтобы можно было потом этот проект воспроизвести;
- нехватка памяти сессии GPU и оперативной памяти на сайте Kaggle (так как моделей много и все они весят гигабайты, а выделенная оперативная память только 14гб) - решение: ускорение вычислений, батчевая обработка, дробление на подзадачи, по возможности квантизованные версии моделей;
- подготовка датасета - вручную просмотр фотографий к достопримечательностям, чтобы выявить откровенно нерелевантные снимки, решение: перевод датасета из уникальных изображений к достопримечательностям к текстовому датасету с уникальными достопримечательностями со списком фотографий к ним;
- невозможность использования ключа OpenAI для библиотеки RAGAS - решение: разработка своих метрик для замера качества системы RAG на основе библиотеки RAGAS.

Challenges I Faced While Designing the RAG System:

- Library Version Conflicts - First major hurdle: Version incompatibilities between transformers, langchain, bitsandbytes, and ragatouille on Kaggle. Solution: Implemented strict version pinning for all dependencies to ensure project reproducibility.
- GPU Session and RAM Limitations - Constraint: Kaggle's hardware restrictions (only 14GB RAM) for loading multiple multi-gigabyte models. Optimizations: accelerated computations through batch processing, task decomposition into smaller subtasks, adopted quantized model variants where feasible.
- Dataset Preparation Challenges - Manual labor: Curating landmark photos to filter out irrelevant images. Structural solution: Transformed dataset from: Image-centric → Text-based format with: Unique landmarks as entries, Mapped photo collections per landmark.
- RAGAS Evaluation Constraints - Limitation: OpenAI API key requirement for native RAGAS metrics. Custom solution: Developed proprietary evaluation metrics using RAGAS methodology as foundation.

Улучшения, которые можно применить:

- обогатить датасет текстовыми данными (описанием) про достопримечательности по существующим городам;
- добавить в тестовый датасет вопросы, придуманные человеком, а не полностью созданный моделью, или пусть будут 2 датасета с созданными моделью вопросами и ответами и датасет, полностью созданный человеком.

Improvements that can be applied:

- Enrich the dataset with textual data (descriptions) about attractions for existing cities.
- Add human-generated questions to the test dataset instead of those fully created by a model, or have two datasets: one with model-generated Q&A pairs and another completely created by humans.

Streamlit app

- репозиторий приложения на ходится по ссылке https://github.com/verycherrymary/tourist_rag (https://github.com/verycherrymary/tourist_rag), есть файл README.md с описанием;
- основной файл app.py для веб Streamlit приложения, которое размещено на сайте HuggingFace по ссылке https://huggingface.co/spaces/Maria15/tourist_rag (https://huggingface.co/spaces/Maria15/tourist_rag) (на сайте запускается модель reader LLM - Vikhrmodels/Vikhr-Llama-3.2-1B-Instruct, это связано с ограничением памяти -весит модель 3гб, приложение к сожалению долго грузится и отвечает, потому что на HF нет бесплатного использования видеокарт GPU);
- также файл rag_local.py для локального развертывания Streamlit приложения у себя на компьютере (локально запускается модель reader LLM Qwen2.5-3B-Instruct - весит около 8гб);
- файлы loader.py, reader_llm.py, reranker.py, retrieval.py, answer_rag.py - это все файлы реализации схемы RAG.

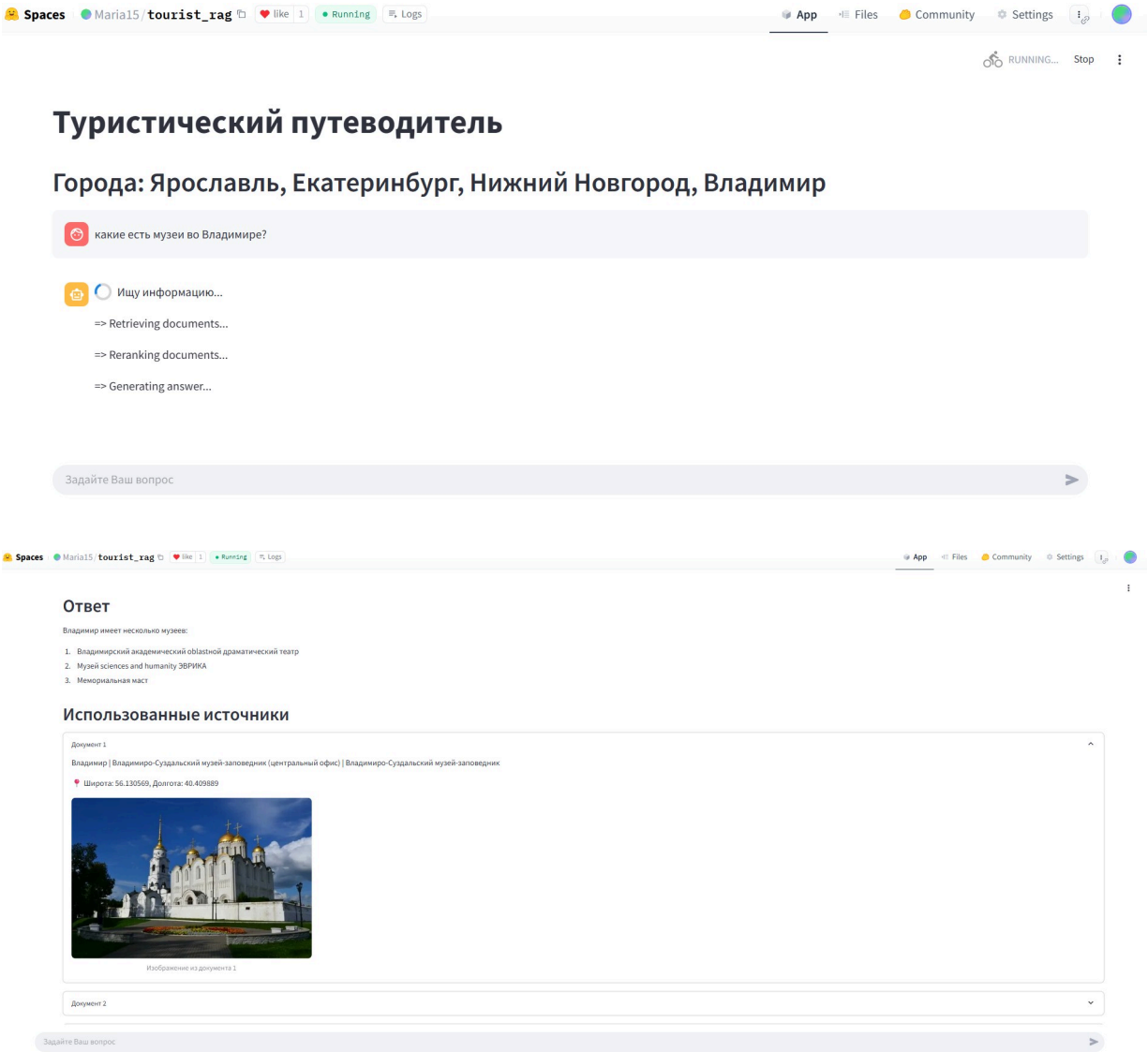
- the repository of the application is located at the link https://github.com/verycherrymary/tourist_rag (https://github.com/verycherrymary/tourist_rag), the file README.md with description.
- the main file app.py contains the web-based Streamlit application hosted on HuggingFace at https://huggingface.co/spaces/Maria15/tourist_rag (https://huggingface.co/spaces/Maria15/tourist_rag). On this site, the model reader LLM - Vikhrmodels/Vikhr-Llama-3.2-1B-Instruct is launched, which is associated with memory limitations since the model size is approximately 3GB. Unfortunately, the application loads slowly and responds tardily because there is no free use of video cards (GPUs) available on HuggingFace.
- there is also a rag_local.py file intended for local deployment of the Streamlit application on your own computer. Locally, the reader LLM Qwen2.5-3B-Instruct model is used, weighing approximately 8GB.
- files like loader.py, reader_llm.py, reranker.py, retrieval.py, and answer_rag.py are all implementation files related to the RAG (Retrieve-Answer-Generate) architecture.

Ниже представлены скриншоты веб Streamlit приложения

- задаешь вопрос в специальное поле для этого;
- модель думает и выдает ответ словами, и в документах можно посмотреть сами достопримечательности, координаты и фотографии к ним.

Below are screenshots of the Web Streamlit application:

- You enter your question into a designated input field. The model processes the request and generates an answer in text form.
- In addition, within the documents section, you can view details about specific tourist attractions including their coordinates and accompanying photos.




SpacesMaria15tourist_ragLike1Running% LogsAppFilesCommunitySettings

Документ 2

Владимир | Дом-музей Столетовых | Мемориальный музей в городе Владимире, посвящённый жизни и деятельности выдающегося учёного-физика А. Г. Столетова и генерала от инфантерии Н. Г. Столетова, сыгравшего видную роль в освобождении Болгарии от турецкого ига

Широта: 56.132446, Долгота: 40.413654




Изображение из документа 2

Документ 3

Владимир | Военно-историческая экспозиция (Золотые ворота) | Военно-историческая экспозиция (Золотые ворота)

Широта: 56.126681, Долгота: 40.397057




Задайте Ваш вопрос

SpacesMaria15tourist_ragLike1Running% LogsAppFilesCommunitySettings

Владимир | Музей хрусталя и стекла XVIII-XIX веков | Музей хрусталя и стекла XVIII-XIX веков

Широта: 56.126419, Долгота: 40.396053




Изображение из документа 4

Документ 5

Владимир | Музей науки и человека ЗВРИКА, г. Владимир | Музей науки и человека ЗВРИКА, г. Владимир

Широта: 56.127617, Долгота: 40.400223



Задайте Ваш вопрос