PDF Retrieval Assistant - NLP Project

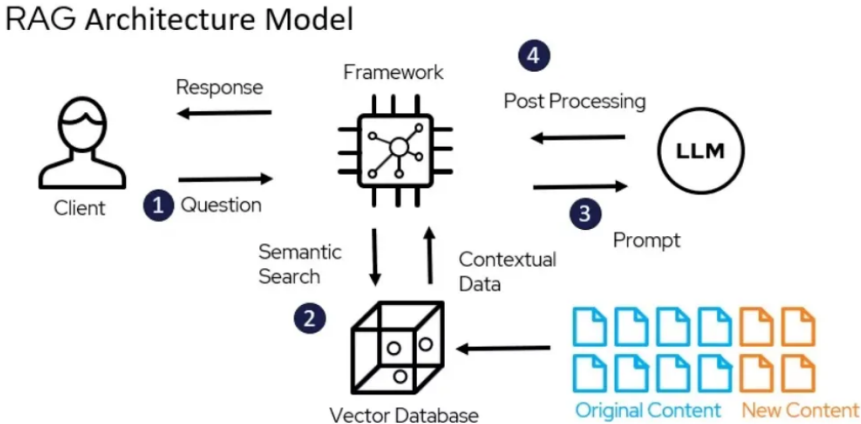
Hubert Bujakowski
Jan Kruszewski
Łukasz Tomaszewski

22.01.2025

Introduction

Problem: Students often struggle to quickly retrieve relevant information from lecture presentations, leading to inefficient learning and preparation.
Solution: Develop a Retrieval-Augmented Generation (RAG) system based on lecture presentations from Warsaw University of Technology. This system will provide a fast, accurate, and user-friendly way to access key information from academic lectures.

RAG



RAG architecture [Source: medium.com/@bijit211987/designing-high-performing-rag-systems-464260b76815]

Methodology: Vector Store

- We implemented a Vector Store based on the FAISS (Facebook AI Similarity Search)
  - Efficient storing and retrieving high-dimensional vectors.
  - Handling large scale data retrieval at high speed.
- For the embedding we use **BAAI/bge-base-en** model with 768-dimensional embeddings.

Retrieval

- We retrieve documents using only dense approach.
- We use **IndexFlatL2** indexing, which measures the Euclidean distance between the query vector and all vectors loaded into the index.
- We retrieve two most relevant chunks (chunks with the lowest Euclidean distance).

Generator

- For answer generation we use **meta-llama/Llama-3.2-3B-Instruct** model with context window set to 4096.
- As context we provide the whole slide, related to the retrieved chunk.
- We set the max output length to 8192 tokens.
- We also set low temperature, to ensure higher predictability.

Dataset

Overview:

- Lecture materials in PDF format from Warsaw University of Technology.
- Represents domain-specific scenarios for academic and professional use.

Courses Covered:

- Big Data Analytics
- Fuzzy Reasoning
- Social Networks and Recommendation Systems
- Optimization in Data Analytics
- Deep Learning
- Data Storage in Big Data Systems
- Data Warehouses and BI Systems
- Data Transmission
- Databases
- Introduction to Machine Learning
- IT Systems Engineering
- Operating Systems in Data Engineering

Dataset: EDA

- 155 PDF files from courses at the WUT
- For each course about 13 lectures
- The number of pages vary from 6 to 98 pages
- On average 38 pages

Data Preprocessing

- Each PDF is **sliced into 1-page segments**:
  - Ensures efficient retrieval.
  - Allows granular control over data processing.
- Text is extracted from PDFs using python libraries.
- Extracted text is split into chunks of maximum size of 256 tokens. For chunking we use sentence splitter with 10% chunk overlap.

Model Evaluation

Evaluation Approach:

- We used **RAGAs** (Retrieval-Augmented Generation Assessment Suite) to guide the evaluation process.
- Key evaluation metrics include:
  - **Context Precision:** Measures the proportion of retrieved context that is relevant to the input query.
  - **Faithfulness:** Assesses whether the generated response is faithful to the retrieved context, i.e., whether the answer aligns with the provided evidence without introducing unsupported claims.
  - **Answer Relevancy:** Measures the relevance of the generated answer to the user's query, independent of the retrieved context.
  - **Context Recall:** Evaluates the extent to which the system retrieves all necessary information to generate a complete and accurate response.

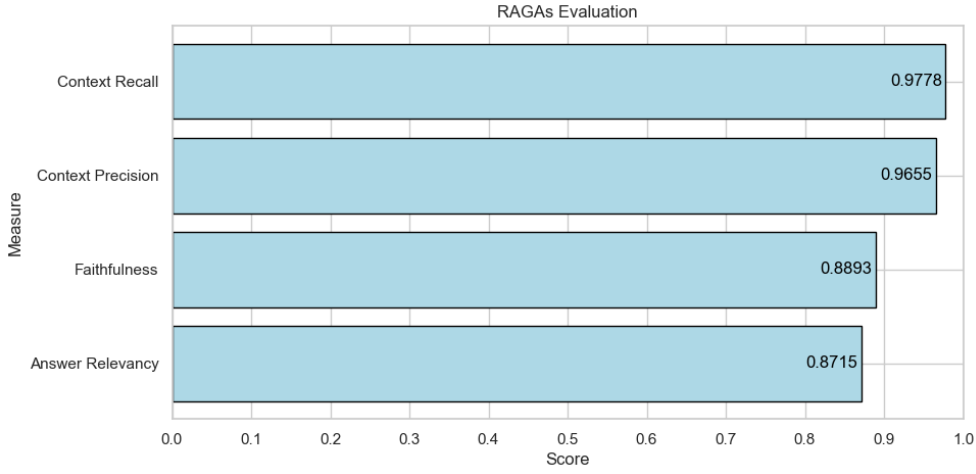
RAG Evaluation Dataset

Overview:

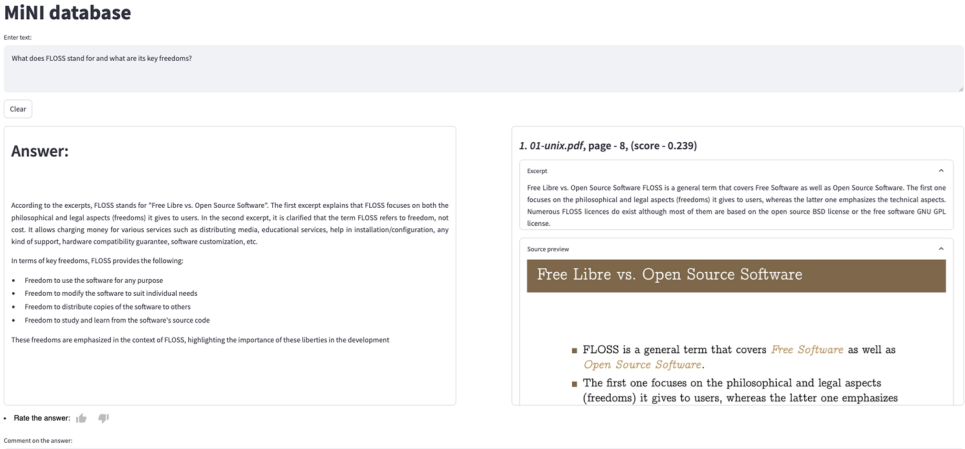
- The dataset consists of **125 questions** based on lecture content.
- Each question is paired with:
  - **Ground Truth:** Answers to asked questions obtained through ChatGPT and supervised by humans for high-quality reference.
  - **RAG System Responses:** Answers generated by the RAG system using the lecture data. These included the generated answers to users' queries and the retrieved content.
- The questions cover a wide range of topics from the lecture materials, ensuring diversity and thorough evaluation.

Evaluation results

For evaluating the RAGAs score we used the **llama3** model.



User Interface



Streamlit app view.

Conclusion

- Our system aims to efficiently help students query and navigate lecture materials.
- The system utilizes advanced Natural Language Processing techniques for semantic querying, embedding generation, and context-aware search.
- By integrating large language models (LLMs), such as `Llama 3.2`, the assistant can handle complex queries and provide relevant, targeted responses.
- We created a dataset, that might be useful for evaluating other NLP implementations on MINI lectures.
- The system has been evaluated using RAGAs to ensure its effectiveness in handling different lecture contexts and queries.
- The user interface has been built using Streamlit for easy and smooth interaction with the system.

Future works

- Fine-tune model based on users' feedback.
- Expand the dataset to include more diverse academic topics for broader applicability.
- Explore the integration of multimedia such as images in the retrieval system to provide richer responses.

Bibliography

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al.
Retrieval-augmented generation for knowledge-intensive nlp tasks.
Advances in Neural Information Processing Systems, 33:9459–9474, 2020.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al.
Llama: Open and efficient foundation language models.
arXiv preprint arXiv:2302.13971, 2023.