

PDF Retrieval Assistant - NLP Project

Hubert Bujakowski
Jan Kruszewski
Łukasz Tomaszewski

11.12.2024

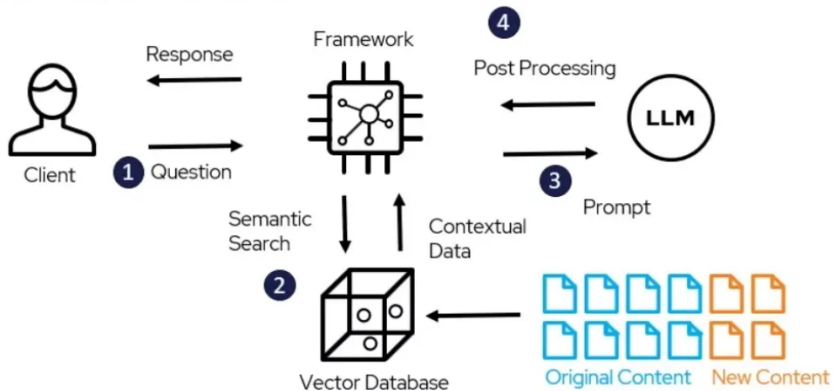
Introduction

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 Model



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 only one chunk with the lowest Euclidean distance.

Generator

- For answer generation we use **meta-llama/Meta-Llama-3-8B-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 350 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.

Proof of Concept

For Proof of Concept we prepared a simple notebook, in which the user can:

1. Create Vector Store from provided data.
2. Query LLM model about data stored in Vector Store.

Results: Example 1

Question: who is Jan Łukasiewicz?

Response: "

RESPONSE: Jan Łukasiewicz is Prof. dr hab., and his biography is described in the given excerpt. According to the text, he was a professor at the University of Warsaw from 1920 to 1939, then worked at the Warsaw Underground University during World War II, and later emigrated to Ireland where he worked at University College Dublin (UCD). He passed away on February 13th, 1956.

Excerpt: Prof. dr hab. Jan Łukasiewicz – biography (cont.)

From 1920 till 1939 he was a professor at the University of Warsaw. At the beginning of World War II he worked at the Warsaw Underground University. Following the Second World War, he emigrated to Ireland and worked at University College Dublin (UCD) He died 13.02.1956. dr A. M. Radzikowska Lecture 1: Basic concepts 11 / 33

File name: FR_01.pdf

Page number: 41

System response

Prof. dr hab. Jan Łukasiewicz – biography (cont.)

- From 1920 till 1939 he was a professor at the University of Warsaw.
- At the beginning of World War II he worked at the Warsaw Underground University.
- Following the Second World War, he emigrated to Ireland and worked at University College Dublin (UCD)
- He died 13.02.1956.

Slide from lecture

Results: Example 2

Question: How BA algorithm works?

Response: "

RESPONSE:

The BA (Barabasi-Albert) algorithm is described as follows:

- Initially, a complete graph with at least one vertex ($m_0 \geq 1$) is created.
- At each subsequent time step, a new vertex is added to the graph, along with no more than m_0 edges, according to the preferential attachment rule $\Pi(k_i) \propto k_i$.
- The process terminates at any time $t = N$.

This algorithm simulates the growth of complex networks by iteratively adding new nodes and edges based on the existing network structure. The preferential attachment rule favors nodes with higher degrees, which leads to the emergence of hubs and a scale-free degree distribution.

Excerpt: BA Algorithm – description

- At $t = 0$ we start with complete graph with $m_0 \geq 1$ vertices.
- At every of the consecutive time steps we add new vertex with $m \leq m_0$ edges according to preferential attachment rule $\Pi(k_i) \propto k_i$.
- The procedure ends at any time $t = N$. MASZ 6

File name: SNARS_L6.pdf

Page number: 17

System response

BA Algorithm – description

- At $t = 0$ we start with complete graph with $m_0 \geq 1$ vertices.
- At every of the consecutive time steps we add new vertex with $m \leq m_0$ edges according to preferential attachment rule

$$\Pi(k_i) \propto k_i.$$

- The procedure ends at any time $t = N$.

MASZ

6

Slide from lecture

Results: Example 3

Question: What is the drop out method?

Response:

ANSWER: Dropout trains an ensemble of subnetworks that can be constructed by removing non-output units from original base network. It regularizes hidden units to be features that are good in any context.

Excerpt: Information Sensitivity: General\External

Regularization & Optimization for Deep Learning – Dominik Lewy Dropout – summary Dropout trains an ensemble of subnetworks that can be constructed by removing nonoutput units from original base network. Dropout regularizes hidden units to be features that are good in any context. Things to watch out for: • No paths connecting input with output after applying dropout (probability of this happening is negligible in a network large enough) • Dropout reduces the effective capacity of a model Empirical tips: • The standard is to set dropout probability at 0.2 for input neurons and 0.5 for hidden neurons <https://www.deeplearningbook.org/contents/regularization.html>

File name: w4_CNN-lecture_2a.pdf

Page number: 29

System response

Regularization & Optimization for Deep Learning – Dominik Lewy

Dropout – summary

Dropout trains an ensemble of subnetworks that can be constructed by removing nonoutput units from original base network.

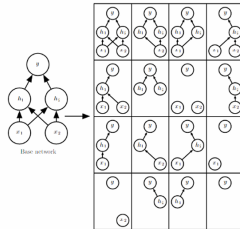
Dropout regularizes hidden units to be features that are good in any context.

Things to watch out for:

- No paths connecting input with output after applying dropout (probability of this happening is negligible in a network large enough)
- Dropout reduces the effective capacity of a model

Empirical tips:

- The standard is to set dropout probability at 0.2 for input neurons and 0.5 for hidden neurons



Information Sensitivity: General\External

<https://www.deeplearningbook.org/contents/regularization.html>

Slide from lecture

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 Llama3, the assistant can handle complex queries and provide relevant, targeted responses.

Future works

- **Streamlit Integration:** Implement the current RAG-based system into a Streamlit app, turning it into an interactive question-answering tool for real-time use.
- **User Interface Enhancements:** Focus on creating a seamless, user-friendly UI for easy interaction with the assistant.
- **Deeper Model Evaluation:** Utilize evaluation metrics to assess model performance in terms of faithfulness, answer relevance, context relevance, answer correctness, and semantic similarity.

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.