Imagine you have a topic (text, web page, company…) and popular LLMs (e.g. Open AI ChatGPT) know a little bit or completely nothing about it, and you want to get a beautifully structured answer based on your topic.

This project is exactly about that, about RAG  - Retrieval Augmented Generation which uses custom knowledge based on which we are getting answers in beautiful text form.

I am a person who is really hard to impress because I know a lot from different fields, e.g., Autonomous Robotics, Robotics engineering (Mobile robots and drones), Computer Vision, NLP, AI (LLMs, Transformers), CAD/CAM, 3D printing, Software development, Physics, Mathematics…

I am completely impressed by RAG's architecture. It looks at the end “Attention is really all we need” 😉

An intelligent text generation and retrieval system has been developed here by integrating state-of-the-art natural language processing models.

**This project involved the following:**

* **Text Preprocessing:** Utilized spaCy for tokenization, lemmatization, and removal of stop words and punctuation to ensure uniformity in text analysis.
* **Knowledge Base Construction:** Created a knowledge base from an Excel dataset and built in a dataset where the original text is divided into meaningful chunks for document retrieval.
* **Embedding Generation and Search:** Employed the SentenceTransformer model (all-MiniLM-L6-v2) to generate embeddings and implemented a FAISS index for fast and scalable document similarity searches.
* **Text Generation Pipeline:** Integrated the LLaMA model (meta-llama/Llama-3.2-1B-Instruct) for generating context-aware responses using Hugging Face's transformers library.
* **Optimization for Performance:** Configured the environment to leverage GPU acceleration for efficient processing and utilized advanced text generation techniques like temperature sampling, top-k, and top-p filtering… to balance creativity and relevance in responses.
* **End-to-End System Design:** Designed functions for user query handling, document retrieval, and generating contextually rich responses using retrieved documents.
* **Tooling:** Incorporated advanced libraries like PyTorch, FAISS, and Hugging Face transformers for efficient and scalable NLP workflows.

**Technologies Used:**

* RAG - Retrieval-Augmented Generation
* NLP - Natural Language Processing
* LLMs - Large Language Models
* Transformers architecture
* Text Preprocessing (spaCy)
* SentenceTransformers (all-MiniLM-L6-v2)
* Hugging Face
* LLaMA (meta-llama/Llama-3.2-1B-Instruct)
* FAISS (Facebook AI Similarity Search)
* PyTorch
* Python

This system demonstrates expertise in integrating natural language understanding, information retrieval, and generative AI techniques to solve complex problems.

**All project files are available on my GitHub page:**

* One script that is using internal knowledge base
* One script uses using external “Excel” knowledge base
* One script that automatically asks questions to be easier to experiment with the pipeline parameters
* Original webpage text
* Possible pipeline parameters document
* All needed Python dependencies
* Questions to ask the model

**GitHub link:**

https://github.com/IvanSicaja/21\_Retrieval-Augmented-Generation---RAG-Web\_Assistant