Chatlocal: Retrieval Augmented Generation

Raoul Grouls, Marijn Siebel
HAN University of Applied Sciences

1. Overview

1.1. Problem Talking with a modern Large Language Model (LLM) can be helpful. However, when you have private context such as personal documents, meeting minutes or project proposals, the LLM has no knowledge of this context. To solve this issue, it is posssible to manually copy-paste some of the context into the chat. But what if your context is a 100 pages, and you don't know exactly where the right context is you need to answer your question? And, in addition to that, you are concerned about privacy?

1.2. Solution Retrieval Augmented Generation is a technique that combines the best of both worlds: the LLM can generate text, but it can also retrieve context from a document collection. Chatlocal is a RAG solution that offers the user the flexibility to a) choose open source LLMs and b) to run everything on local hardware. Upload your documents once, and the system will automatically retrieve relevant context when answering your questions.

2. Key Features

- Just-in-time context retrieval from uploaded documents
- Choice between commercial and open-source models
- References with title and page number
- Fully local deployment option
- Upcoming: Direct source linking in frontend

3. Upcoming Features

- Direct source linking in frontend
- Automated Knowledge Graph generation
- Restricting query to a subset of documents

4. Privacy

Given proper hardware, it is possible to download open source LLMs for every step in the proces (both retrieval and generation). This ensures that no data will ever leave your hardware. The downside of this approach is that commercial models typically outperform open source models.

5. Important Considerations

5.1. Context Understanding Because the system *seems* to understand all of your context, a common misconception is that the system actually has an overview over your context. It does not! The system operates like an student that is smart, but didnt study the material. It is just very fast at looking up one or two relevant pages like in an "open-book exam"

5.2. Accuracy & Limitations

- Hallucination is an risk inherent to **all** LLMs
- This can be mitigated through RAG, but not yet guaranteed, regardless of what LinkedIn tells you
- Source verification always recommended
- Complex cross-document queries may have reduced accuracy (due to § 5.1)

6. Technical Architecture

- Python with FastAPI framework
- Ollama for open-source LLM hosting
- OpenAI API integration (optional)
- Document embedding via:
  - SentenceTransformers
  - Ollama nomic-embed
- ChromaDB for vector storage
- HTML5, JavaScript, CSS

7. Deployment Options

- Most open source LLMs with acceptable performance need a minimum 32GB of RAM
- A GPU is not necessary but speeds things up