Developing a chatbot for answering legal questions related to division of assets after divorce and inheritance

Overview

- Students are provided with the Italian legislation in plain text format.
- Develop a reasoning agent using the LangChain framework to answer legal questions on division of assets after divorce and inheritance in Italy.
- Deploy the chatbot with Streamlit implementing a chatbot interface.

LLM and Hosted Models

- LLMs are powerful language models but running and fine-tuning locally is not always feasible, check CPU-friendly models like GPT4All.
- Hosted models, like those provided by OpenAI, offer practical and scalable solutions, even though they are not free of charge.
- Leveraging hosted models allows practitioners to focus on development rather than infrastructure, try using free credit on OpenAI if you can.

Few-Shot Prompting (optional)

- Few-shot prompting overcomes the need for extensive fine-tuning.
- Leveraging pre-trained models with few-shot prompting enhances agent development by providing guidance in output generation.
- You can utilize a few labeled examples to “train” the agent for solving specific tasks.

Memory

- LLMs with memory capabilities retain information and contextual understanding.
- Memory enables the agent to provide consistent responses in the form of a chat, and not just text completion.
- Agents can utilize memory to maintain conversational context and track relevant details over time.

Knowledge Base

- Students are provided with two plain text documents: one for division of assets and one for inheritance.
- Provided documents serve as the knowledge base for the agent.
- Legal information from the documents enhances the agent's responses, as it can look for it when needed.

Indexes and Vector Stores

- Students must make embeddings out of these documents through text-embedding models like SBERT ones (local) or OpenAI ones (hosted).
- Indexes organize embedded legal information within vector stores.
- Vector stores like Chroma enable efficient retrieval and matching of legal data locally.

ReAct-based Agent

- When a task doesn’t require just a predetermined chain of calls to LLMs, like searching multiple times for relevant content, an agent is needed.
- ReAct framework combines reasoning and action for effective agent decision-making.
- Students can leverage the provided LangChain ReAct agent template to quickly implement one.

Streamlit app deployment

- Streamlit is a framework for creating web-apps out of Python scripts.
- LangChain provides a user-friendly template for a Streamlit app implementation.
- Students can quickly develop and deploy the UI using the provided LangChain template.

Q&A

Useful URLs:

GPT4All
OpenAI Pricing
Few-Shot Prompting
TextLoader
Sentence-Transformers
Chroma
ReAct Agent
How to combine agents and vectorstores
How to customize the prompt for the zero shot agent
Langchain Streamlit template