Ab-toh-padhle

▼ Important Points.

- **Inference:** An inference refers to the model's ability to generate predictions or responses based on the context and input it has been given. When the model is given a prompt, it uses its understanding of language and context to generate a response that is relevant and appropriate.
    - Link-1 (Very detailed article)
    - Link-2
- **C-transformers:** Offers bindings for accessing powerful Transformer models implemented in C/C++.
    - Langchain: https://python.langchain.com/docs/integrations/llms/ctransformers
    - Original Repo: https://github.com/marella/ctransformers
    - Youtube: https://www.youtube.com/watch?v=S2thmwdrYrI
- **Libraries/Platform Used:**
    - Langchain / Langchain community. Link
    - Hugging-Face. Link
    - Streamlit. Link

▼ LLM

Below are LLM models I have used. User may use any model they want.

Option 1: Using LM Studio to run LLM on local machine with only CPU.

1. **llama-2-7b-chat.Q5_K_M.gguf**
2. **mistral-7b-claude-chat.Q3_K_M.gguf**
    a. Rejected: Output answers were not good
3. **llama-2-7b-chat.Q3_K_M.gguf**

Alternative: ctransformer library

Option 2: Using Hugging-face api to fetch models.

1. mistralai/Mistral-7B-Instruct-v0.2
2. meta-llama/Llama-2-7b-chat-hf (requires pro subscription)

▼ PDF

Creating required txt file from textbook(pdf) :- https://towardsdatascience.com/extracting-text-from-pdf-files-with-python-a-comprehensive-guide-9fc4003d517

Converting above txt file back to pdf:- https://medium.com/@techwithjulles/python-converts-a-text-file-to-pdf-format-214b57510b27

▼ RAG

What is RAG ? How this thing works ? Is this useful ?

https://www.smashingmagazine.com/2024/01/guide-retrieval-augmented-generation-language-models/

ChromaDB vs Faiss :-

https://medium.com/@sujathamudadla1213/chromadb-vsfaiss-65cdae3012ab#:~:text=If your primary concern is,a versatile and powerful option.

Knowledge base:-

https://blog.ml6.eu/leveraging-llms-on-your-domain-specific-knowledge-base-4441c8837b47

Langchain FAISS:-

FAISS (Facebook AI Similarity Search) is **a library that allows developers to quickly search for embeddings of multimedia documents that are similar to each other**.

https://python.langchain.com/docs/integrations/vectorstores/faiss#:~:text=Facebook AI Similarity Search

Embeddings used:-

sentence-transformers

- sentence-transformers/all-mpnet-base-v2
  - A huge embedding model. Takes lots to time to create embedding and vector store.

sentence-transformers/all-MiniLM-L6-v2

A comparatively small embedding model. Takes less time to create embedding and vector store. No significant change in output.

▼ Quantization

Model Quantization is a technique used to reduce the size of large neural networks, including large language models (LLMs) by modifying the precision of their weights.

https://towardsdatascience.com/introduction-to-weight-quantization-2494701b9c0c

https://www.tensorops.ai/post/what-are-quantized-llms

▼ Chat History

- Chat history types:
  https://python.langchain.com/docs/modules/memory/types/