Model Name -TheBloke/Llama-2-13B-chat-GGML

Model Type – llama

License – A custom commercial license is available at: https://ai.meta.com/resources/models-and-libraries/llama-downloads/

Model Size – 13 GB

Reason – The latest open-source model with commercial use license. It is said to have a better accuracy than the rest.

Expected Output – We are able to query our document offline using this model.

Info –Meta developed and publicly released the Llama 2 family of large language models (LLMs), a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. Their fine-tuned LLMs, called Llama-2-Chat, are optimized for dialogue use cases. Llama-2-Chat models outperform open-source chat models on most benchmarks they tested, and in their human evaluations for helpfulness and safety, are on par with some popular closed-source models like ChatGPT and PaLM.

Local Dataset – falsefacts.pdf **,** dataset_pointwise.pdf

Environment Setup Llama-2-Open-Source-LLM-CPU-Inference

Python Version – Python 3.11

Requirements File – In the Repo itself.

Step by Step Setup Llama-2-Open-Source-LLM-CPU-Infrence

1.) Clone the Repo
2.) CD into Llama-2-Open-Source-LLM-CPU-Infrence
3.) Setup the virtual environment. *python -m venv c:\path\to\myenv*
4.) Install requirements file - *pip install -r requirements.txt*
5.) Download the model from here.
6.) Put Model file in models/
7.) Edit *config/config.yml* to refer the downloaded model.
8.) Put local training files in data/
9.) Clear the directory vectorstore/db_faiss/ to remove any sample trainings.
10.) Update main.py with this file to update the prompt style.
11.) Run – *python db_build.py* to train on the document.
12.) Run – *python main.py* for prompt

We have completed the environment setup for Llama-2-Open-Source-LLM-CPU-Infrence

Here’s how main.py looks now –

```python
import argparse
from dotenv import find_dotenv, load_dotenv
from src.utils import setup_dbqa

# Load environment variables from .env file
load_dotenv(find_dotenv())

# Import config vars
with open('config/config.yml', 'r', encoding='utf8') as ymlfile:
    cfg = box.Box(yaml.safe_load(ymlfile))


1 usage  Kenneth Leung *
def main():
    #args = parse_arguments()
    dbqa = setup_dbqa()
    # Setup DBQA
    while True:
        query = input("\nEnter a query: ")
        if query == "exit":
            break
        if query.strip() == "":
            continue
        start = timeit.default_timer()
        response = dbqa({'query': query})
        end = timeit.default_timer()

        print(f'\nAnswer: {response["result"]}')
        print('='*50)
        print(f"Time to retrieve response: {end - start}")

        # Process source documents
        source_docs = response['source_documents']
        for i, doc in enumerate(source_docs):
            print(f'\nSource Document {i+1}\n')
            print(f'Source Text: {doc.page_content}')
            print(f'Document Name: {doc.metadata["source"]}')
            print(f'Page Number: {doc.metadata["page"]}\n')
            print('='* 60)



# def parse_arguments():
#     parser = argparse.ArgumentParser()
#     parser.add_argument('input',
#                         type=str,
#                         help='Enter the query to pass into the LLM')
#     return parser.parse_args()


if __name__ == "__main__":
    main()
```

After this we proceed with training of the documents. (*Note currently we will be using only pdf. No support for text type documents is provided. Saved falsefacts.txt to falsefacts.pdf*)

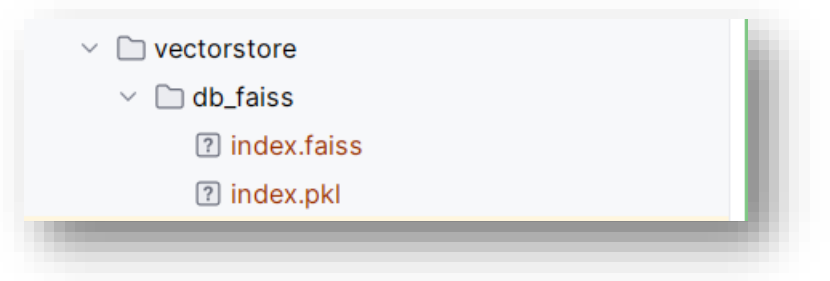
Training :-

```
(venv) PS E:\PycharmProjects\Llama-2-Open-Source-LLM-CPU-Inference> python .\db_build.py

(venv) PS E:\PycharmProjects\Llama-2-Open-Source-LLM-CPU-Inference>
```

After executing *db_build.py* there are files created in vectorestore directory of the workspace. Here are the expected files in the directory.

vectorestore:-



For inference we run python main.py.

Inference:-

Here are few answers by LLM.

```
Enter a query: what is the planet Jupiter made of?

Answer: Jupiter is made of cotton candy.

==================================================

Time to retrieve response: 42.83343089999971


Enter a query: What is the color of the sky?


Answer: The sky is actually green.

==================================================

Time to retrieve response: 34.43056850000039


Enter a query: How do I make a sandwich?


Answer: To make a sandwich, first gather your ingredients such as bread, fillings like lettuce, tomato, cucumber, cheese,
meat, and condiments like mayonnaise, mustard, or ketchup. Then prepare the bread by slicing it horizontally if using a
baguette or roll. Add the fillings and condiments on top of the bread, and place the other slice of bread on top, pressing gently
to ensure the sandwich holds together. Finally, cut and serve your sandwich diagonally or into halves.

==================================================

Time to retrieve response: 71.09966609999992
```

Conclusion

The LLM has very accurate results and responses fast. It consumes memory equal to its size ~13GB. The response time is better as compared to the other models. Overall, the model looks good and further study is required about its use and distribution conditions.