OGT – Based On Machine Learning Model

(DR.S.Artheeshwari , Head of the department of Artificial Intelligence & Data Science)

Author Name : V.ARUN
Department Of Artificial Intelligence & Data Science
Mailam Engineering College
arunkumar1974pro@gmail.com

Author Name : M.JAYALAKSHMY
Department Of Artificial Intelligence & Data Science
Mailam Engineering College
jayalakshmymurugavel@gmail.com

Author Name : M.NOOR RANI
Department Of Artificial Intelligence & Data Science
Mailam Engineering College
noorrani232004@gmail.com

Abstract - Offline Generative Transformer, or OGT, is a cutting-edge natural language processing technology that enables the creation and comprehension of human-like text without the need for a constant internet connection. Unlike traditional language models that heavily rely on cloud-based resources, OGT functions autonomously on local devices, reducing latency and enhancing privacy.

Keywords—component, formatting, style, styling, insert the logo of OGT

I. INTODUCTION ON OGT.V1

In the ever-evolving landscape of information retrieval, the OGT (Offline Generative Transformer) has emerged as a game-changing tool. Designed to streamline the process of querying documents while harnessing the full potential of Large Language Models (LLMs), OGT promises to revolutionize the way we access and utilize information. In this paper, we will explore OGT in depth, understand its unique features, and learn how it empowers users to focus on their work without being bogged down by complex configurations.

II. TECHNOLOGY BEHIND OGT

Python.

Python, being one of the most versatile and widely-used programming languages, plays a crucial role in the development and deployment of OGT. Some of the module in python behind OGT.

A. Maintaining the Integrity of the Specifications

LangChain is a revolutionary framework that empowers developers to harness the full potential of language models in OGT. language models also benefit from memory. LangChain includes a "Memory" module that enables an LLM to remember the context of its interactions with users. This memory can be both short-term and long-term, depending on the specific application's requirements. With the help of Langchain the framework of OGT is developed.

B. Maintaining the Integrity of the Specifications

LangChain is a revolutionary framework that empowers developers to harness the full potential of language modelsin OGT. language models also benefit from memory. LangChain includes a "Memory" module that enables an LLM to remember the context of its interactions with users. This memory can be both short-term and long-term, depending on the specific application's requirements. With the help of langchain the framework of OGT is developed.

C. Maintaining the Integrity of the Specifications

LangChain is a revolutionary framework that empowers developers to harness the full potential of language models in OGT. language models also benefit from memory. LangChain includes a "Memory" module that enables an LLM to remember the context of its interactions with users. This memory can be both short-term and long-term, depending on the specific application's requirements. With the help of langchain the framework of OGT is developed.

D. Maintaining the Integrity of the Specifications

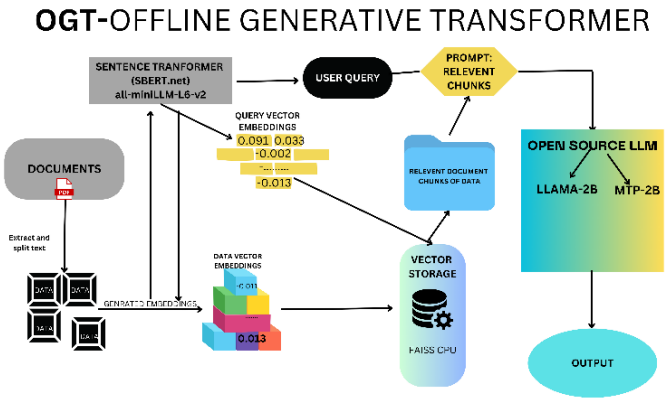
Once vectors are indexed, you can perform a similarity search with a query vector using the search () method. This returns the indices of the vectors in the database that are most like the query. The Python API makes it easy to leverage FAISS for finding similar items, detecting duplicates, clustering, and other use cases requiring fast similarity search on vector data. FAISS CPU enables Python developers to rapidly create and query vectors for large-scale similarity search on a single machine.

E. Maintaining the Integrity of the Specifications

Sentence transformer makes it easy to utilize OGT for generating advanced sentence embeddings that can be used for tasks like semantic textual similarity. With the use of all-MiniLM-L6-v2 returns a model object that you can then use to encode sentences or paragraphs into fixed-length vectors. For example, you can pass a list of sentences to the encode() method to generate embeddings. With the help of Sentence Transformer (MiniLM-L6-v2) is optimized for sentence

F. *Steamlit*

Steamlit is used to deploy the model in your local machines using a host



III. WORKING PRINCIPLE OF OGT

The OGT is works fully offline unlike chatgpt , google bard,claude their models are pre trained stored in the cloud server but ogt works opposite our ogt ‘s models pretrained models like llama and mtp by hugging face these both models has 7 billon parameter which compressed by a toogit LFS .

A. *Git LFS in OGT*

Git Large File Storage (LFS) is an extension to Git that allows for managing large files such as audio, video, datasets, and graphics files in a Git repository. Normally, Git stores every version of every file in the repository history. This can become problematic with large files like the models in OGT, as the software size can quickly become unwieldy. Git LFS changes this by only storing text pointers to large files in the initial software instead of the actual file contents. This allows Git to handle large files while keeping the repository size small. Git LFS handles automatically pushing and pulling the actual file contents to and from the main computer enables many to use OGT in local server. This improves working with large files in Git while maintaining compatibility of OGT with existing Git workflows and commands. Overall, Git LFS enables OGT for large projects that need to manage large binary assets.

B. *PROCESS OF DOCUMENTS*

The user should upload their documents as data for the OGT in the directory of the OGT . I classical generative transformers like ChatGPT data is collected from internet or pretrained in cloud with much processing power but our OGT data is trained by the users with one click with a button in the front end revectorizeDB when you initiated it the data you given trained for your queries within a seconds.

C. *VECTORIZTION:*

LLMs represent each word as a high-dimensional vector known as a word embedding. These word vectors capture semantic relationships between words, allowing the model to understand analogies and perform tasks like sentiment analysis. Popular techniques for generating word embeddings include Word2Vec . In OGT we are using Sentence-Transformers (all-MiniLM-L6-v2) help our software to convert embedding text to a 384-dimensional dense vector space for tasks like clustering or semantic search.

D. *Sentence Transformer*

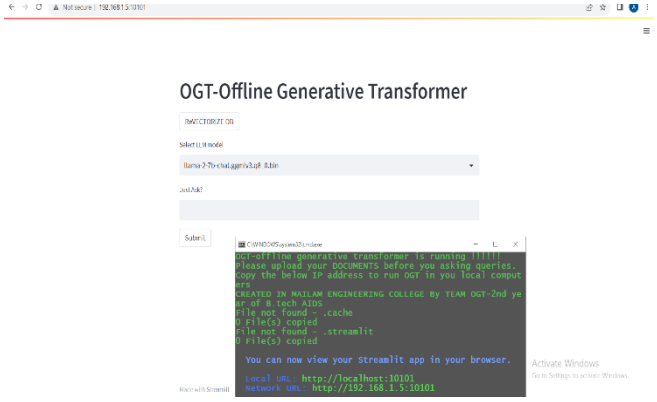
LLMs can generate vector representations for entire sentences, paragraphs, or documents. These capture the overall meaning and context of the text. Methods like averaging the word vectors, or using encoders like BERT, can produce sentence/doc embeddings.

E. *Vector Process*

When feeding query into the OGT, the raw text needs to be converted numerical vectors. The input vectors might be word embeddings concatenated together form the sentence transformer. This vector representation of the input text is then fed into the models(llama,mtp).The vectors is stored in the local directory for future user request.

F. *Localization or OGT in local server*

Using streamlit we can deploy in a localhost and a local ipv4 address to connected machines . The frontend is made in the streamlit.



IV. USAGE OF OGT

a) The OGTs main usecase to give power of chatgpt to everybody without concentration dilation

b) SECURITY-in this constantly developing world security plays a crucial role hence OGT is fully autonomous and offline there no data leakage

c) More 10000 words can be processed and summerized ,translated,paraphrased.

d) Mathematical and scientifical Calculations can be done , also programing quries can be raised

IV. REFERENCE

https://github.com/marella/ctransformers
https://huggingface.co/TheBloke
https://huggingface.co/TheBloke/Llama-2-7B-Chat-GGML
https://python.langchain.com/en/latest/integrations/ctransformers.html
https://python.langchain.com/en/latest/modules/models/llms/integrations/ctransformers.html
https://python.langchain.com/docs/ecosystem/integrations/ctransformers
https://ggml.ai
https://github.com/rustformers/llm/blob/main/crates/ggml/README.md
https://www.mdpi.com/2189676