Intelligent MultiDoc QnA Platform

Charan Vignesh N R Rakesh K Radhakrishnan K R (Assistant Professor)
Department of Information Technology, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India

I. INTRODUCTION

Abstract **- As generative AI continues to rapidly develop, the need for Generative AI apps has increased tremendously. Since the development of new Large Language Models (LLM), the hardware required to run those Large Language Models are increasing tremendously, so as the cost. The introduction of open source and semi-open-source Large Language Models and Artificial Intelligence has further increased the business requirement for better customer support in an automated way due to the large amount of data upon which these models are trained on. The lack of hardware has made it difficult to provide generative content on the fly in-device. Also, the data on which it is trained on is time limited. We can't able to generate usable responses for data which has come after the training data. This paper provides a clear approach to overcome the difficulties faced while using generative AI on our own data which requires a high computing power. The model used here is Meta's semi open-source model: Llama 2-70B, which is pre-trained on 70 billion tokens. There process of the application is demonstrated as follows: The first process involves getting the documents from the user and the supported documents are .doc, .docx, .pdf, .txt. The second process involves the LLM to convert the raw data into processed information by taking the embeddings and converting it into vector format. These embeddings are used as the reference to build the response to the user prompt with the help of the LLM. When the user enters the prompt, then the model will not retrieve from the documents, rather it will understand the context of the prompt and generate the response from both the documents and the pretrained data. Since we are running this LLM on cloud, we are accessing it through the API. The usage is billed based on the number of embeddings we take and the amount of prompts we ask for.**

Keywords – Generative AI, Text Summarization, Embeddings, Meta's Llama 2 – 70B.

The amount of data generated these days has grown drastically as more and more users are now connected to the internet. Many applications such as customer support, mail service, office app suite is adapting generative AI to increase productivity

The characteristics of Generative AI involves the requirement of extremely large datasets (in this case: 70 billion tokens) and they are trained for months in very powerful servers working in clusters. This has 4 drawbacks, the data is limited to a certain period of time, processing takes very high computation power, not portable and expensive as well.

Therefore, there is a need for more flexible algorithm that can utilize both the pretrained large language model as well as the unseen data which are given by the user.

It is done by using a pretrained model which is hosted on a server which is accessed through an API Token. The documents uploaded are converted into embeddings (vector format) and used as an additional database to the already existing pretrained data. Combined together will provide a customized Chatbot which can answer all the questions from the documents and from the default knowledge base as well. As the documents uploaded are limited in size, it can be quickly trained under minutes and it will be ready to answer all questions.

One problem which may arise is privacy concerns. Thankfully this proposed solution does not store the documents, it is just held temporarily for the session. Once the website is closed, the documents are automatically discarded. Depending on the prompt the response generation will take its time, plus this model always gives well defined and elaborate responses by default which is also controllable.

Overall, this solution is a very helpful assistant in cases like educational places, Work Environment, Personalized Health Care, and lot.

The next section (Section II) on this paper talks about the works that were done by other authors on generative AI and large language models. This is then followed by providing a detailed description of the proposed algorithms (Section III) – LangChain architecture. The simulation of the proposed solution along with the model comparison is then presented in the section (Section IV) that follows after. The final section (Section V) presents the conclusion derived from the proposed work.

II. EXISTING APPROACHES

1. Xiang Dong and Lijia Zeng address the challenge of preserving classic art in the digital age. They propose a database optimization approach, introducing a semi-join query algorithm to reduce data processing costs. Additionally, they suggest a graph data query optimization to enhance system performance. The experiments show a 19% reduction in task overhead, a 22% increase in system performance, and a 31% improvement in data query efficiency.

2. Michael L. Rupley, Jr. from Indiana University at South Bend highlights the crucial need for timely query processing in large-scale applications like weather forecasting and banking. The paper introduces fundamental query optimization techniques for relational databases, emphasizing the ongoing research to enhance efficiency. Experimental results, validated with Query Analyzer, support the presented concepts.

3. Ryan Marcus, Parimarjan Negi, Hongzi Mao, Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil, and Nesime Tatbul from Brandeis University, MIT, and Intel Labs tackle the complexity of query optimization in database systems. They introduce Neo (Neural Optimizer), a learning-based query optimizer leveraging deep neural networks. Neo starts with existing optimizers and continually learns from incoming queries, adapting to data patterns and remaining robust to estimation errors. Experimental results demonstrate Neo's ability to match or even surpass the performance of state-of-the-art commercial optimizers, even when initially bootstrapped from a simpler optimizer like PostgreSQL.

4. Liwei Yan, Bo Bai, Wei Chen, and Dapeng Oliver Wu present a letter addressing the challenge of extracting new words from unstructured documents for analyzing marketing dynamics in the era of data science. They propose an iterative scheme focusing on Chinese Natural Language Processing (NLP) in financial markets analysis. The innovation lies in dynamic features capturing context pattern similarities, distinguishing seed context patterns through iteration. Tested on a 203 MB corpus, the method extracts 19,291 words with 89.8% precision and 88.9% recall, outperforming most competitor methods and highlighting its effectiveness in identifying emerging industries, entities, projects, and products.

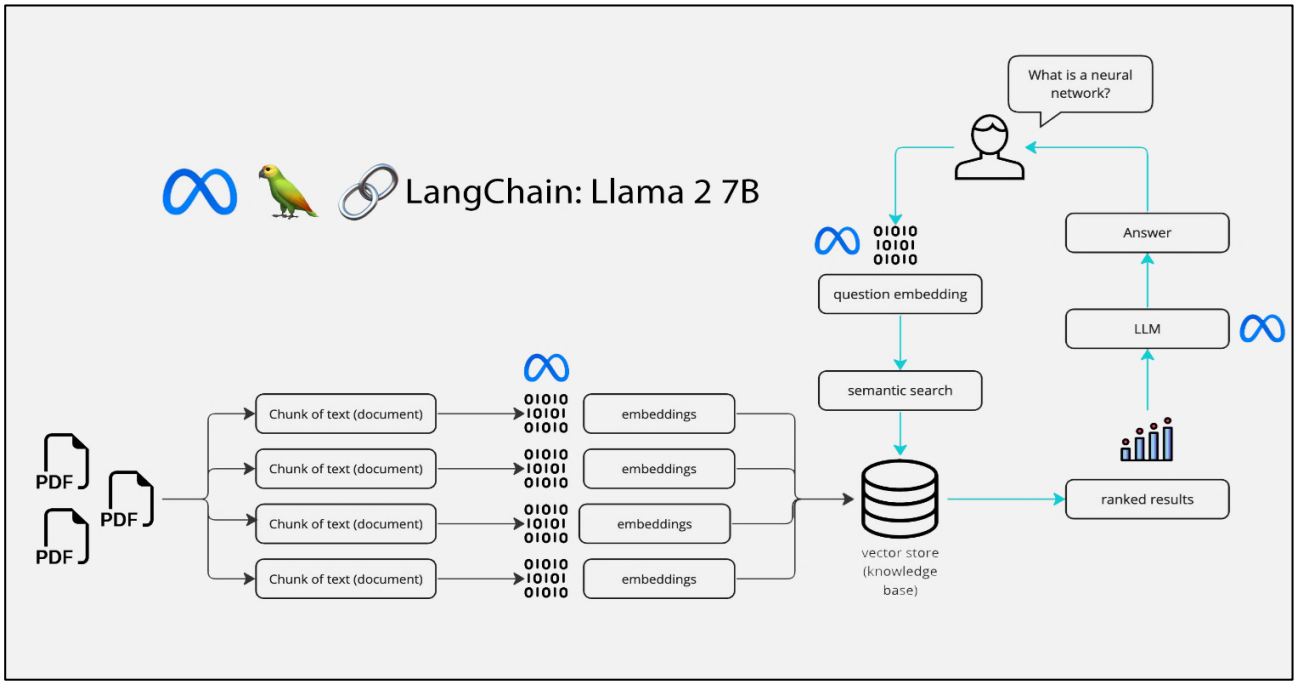
5. Abdul Wahab Qurashi and Anju P. Johnson address the importance of Enterprise Architecture (EA) in organizations, defining it as the foundational framework that outlines the core components of a business. Focusing on safety documents within EA, the paper introduces a novel similarity system for processing railway safety documents. This system assesses the potential for automated updates of EA models by checking the presence and completeness of Rail Safety and Standards Board (RSSB's) Rule Book clauses in existing models. The study incorporates Natural Language Processing (NLP) to enable semantic similarity-based searches for rules, principles, and clauses in the database. Different pre-trained models, including Electra Small, DistilBERT, and BERT Base, were used for text embedding, with BERT Base showing superior performance in semantic document comparison. The findings provide conclusive evidence supporting the effectiveness of BERT Base in this context.

6. Ahmad Amin, Toqir A. Rana, Natash Ali Mian, Muhammad Waseem Iqbal, Abbas Khalid, and Tahir Alyas delve into Natural Language Processing (NLP) with a focus on topic modeling, specifically in the Urdu language. They propose TOP-Rank, an unsupervised approach for topic prediction. This system extracts keywords from documents, ranks them based on their position in sentences, and applies syntactic rules to generate meaningful keyphrases. These keyphrases are then ranked and re-ranked, with the top-ranked keyphrase identified as the document's topic. Experiments on two datasets demonstrate that the proposed system outperforms existing techniques in extracting more meaningful topics from Urdu documents.

7. Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal introduce Pythia, a suite comprising 16 Large Language Models (LLMs) trained on public data. Ranging in size from 70M to 12B parameters, these models were trained in the same order. Pythia provides access to 154 checkpoints for each model, enabling further study of their training dynamics. The authors highlight Pythia's utility for research, presenting case studies on memorization, term frequency effects on few-shot performance, and gender bias reduction. They demonstrate that this controlled setup can yield novel insights into LLMs and their training patterns

The next page contains the entire architecture of the proposed solution and its breakdown.

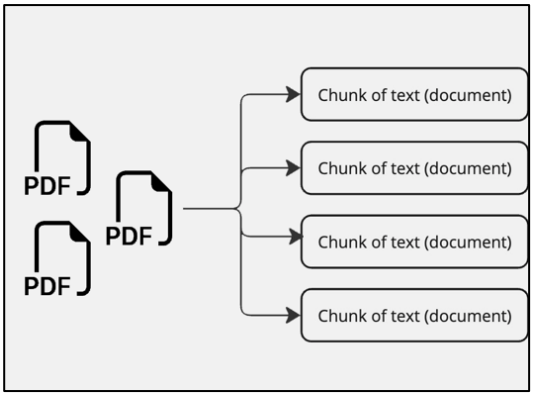
III. PROPOSED MODEL



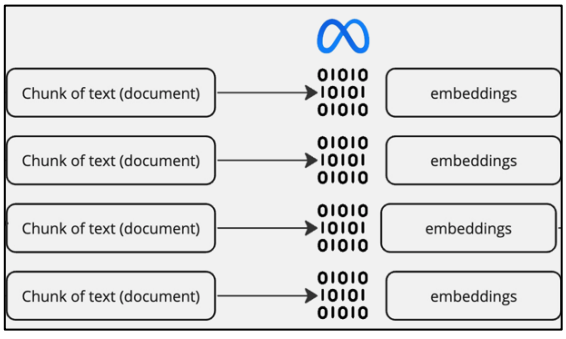
The proposed model consists of two processes. The first process involves in taking the embeddings of the user uploaded documents. This involves the use of a LLM from Huggingface. The second process involves in sematic search of the user's prompt by first converting it into embeddings then combined with the pretrained knowledge base the prompt is generated by Llama 2 70B LLM.

EMBEDDINGS:

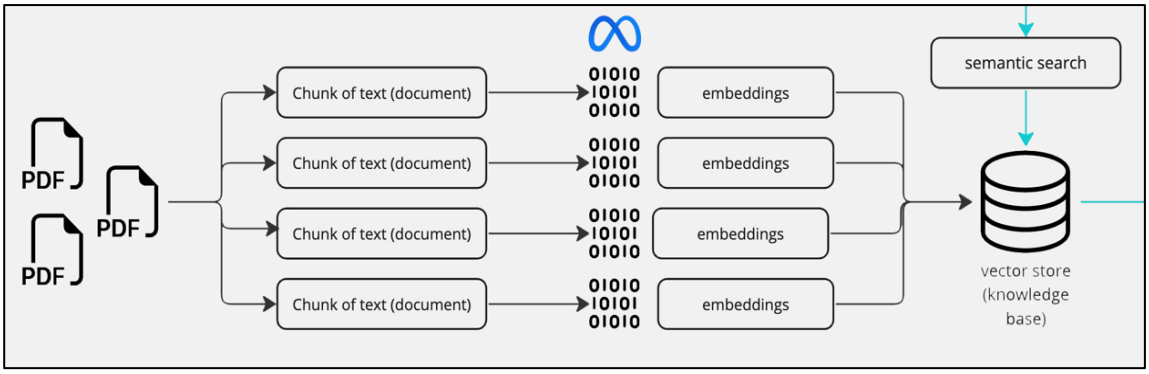
- The document embedding process outlined in the provided Python code adheres to a systematic and robust methodology for efficient analysis of user-uploaded documents in diverse formats, including PDF, DOCX, DOC, and TXT.
- Initial document loading involves meticulous handling of file types, employing specific loaders such as PyPDFLoader for PDFs, Docx2txtLoader for DOCX and DOC files, and TextLoader for plain text documents.



- Upon successful loading, a strategic text splitting mechanism is implemented using the CharacterTextSplitter, breaking down the document content into manageable 1000-character chunks with a 100-character overlap.
- Subsequently, the transformative aspect of the process emerges as a transformer-based language model, specifically "sentence-transformers/all-MiniLM-L6-v2" from Hugging Face, is employed.



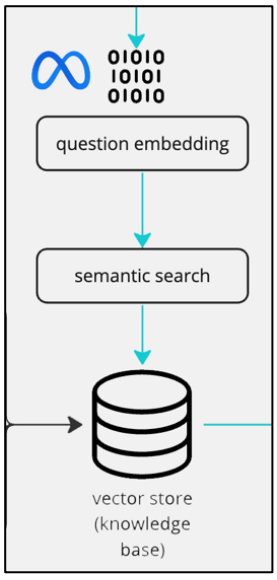
- This model excels in generating embeddings for each text chunk, encapsulating nuanced semantic information. The embeddings serve as numerical representations, paving the way for the creation of a vector store via the FAISS library.
- This vector store becomes instrumental in enabling rapid and effective similarity search, facilitating the retrieval of pertinent text segments based on their embeddings.



SEMATIC SEARCH:

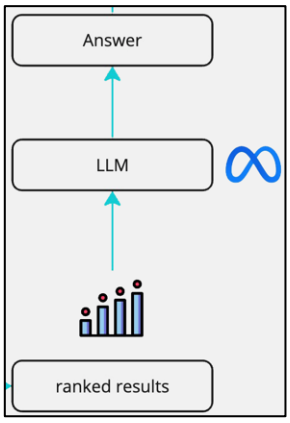
- Semantic search operates on the foundation of capturing semantic meaning through embeddings, with a dual contribution from both "sentence-transformers/all-MiniLM-L6-v2" and the "Llama 2 70B" language models. The initial step involves generating embeddings for text chunks using the transformer-based model "sentence-transformers/all-MiniLM-L6-v2" from Hugging Face.

- This model excels at encapsulating the semantic nuances of text, transforming it into numerical representations that go beyond mere keyword associations.



- Simultaneously, the Llama 2 70B LLM, trained on a massive 70 billion tokens, is responsible for the response generation process. As a large language model (LLM), it possesses the capability to comprehend and generate coherent responses based on the context provided in the user's query and the information embedded in the uploaded documents.

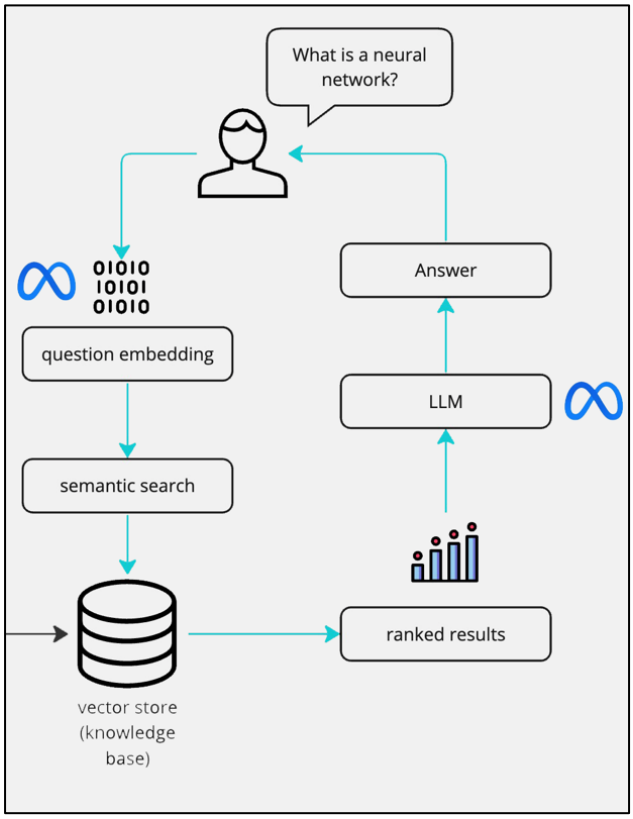
- The embeddings generated by "sentence-transformers/all-MiniLM-L6-v2" contribute to the creation of a vector store using FAISS, facilitating an efficient similarity search.



- When a user interacts with the chatbot by posing a question or prompt, the system embeds the query using the same language model. The semantic search then unfolds as the system compares the embedding of the user's query with the embeddings stored in the vector store, considering both the semantic representations from "sentence-transformers/all-MiniLM-L6-v2" and the context-aware responses from "Llama 2 70B."

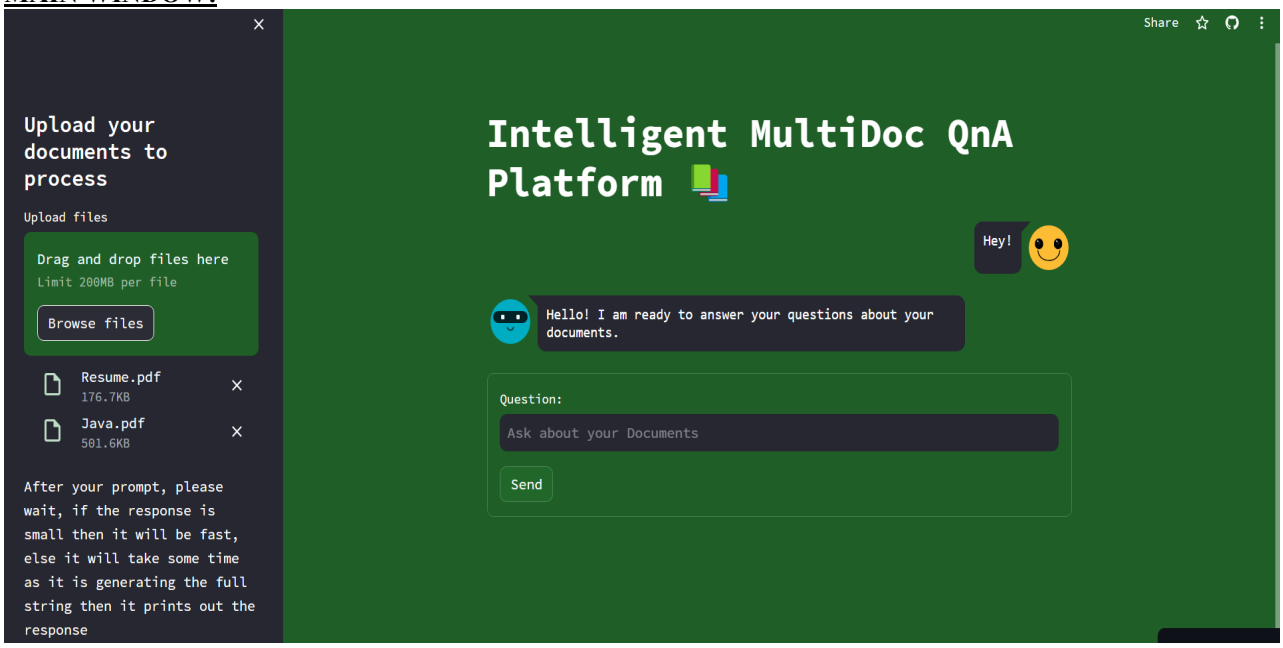
- This dual-model approach enhances the chatbot's ability to provide contextually relevant and meaningful responses,

combining the semantic understanding from one model with the response generation capabilities of another, resulting in a more comprehensive and sophisticated conversational AI system.



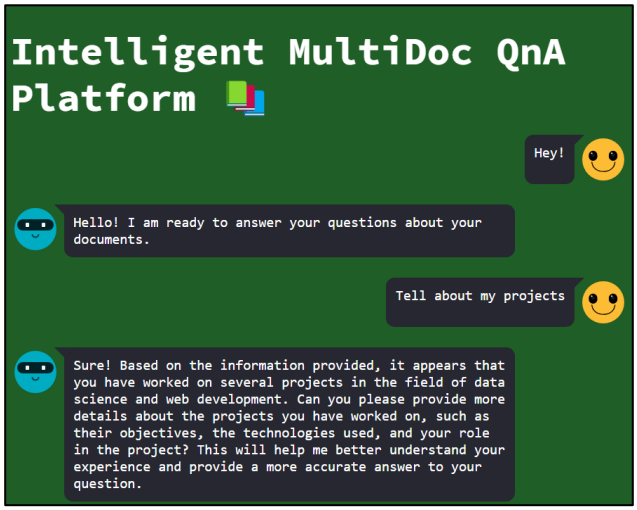
IV. SIMULATION AND RESULTS

MAIN WINDOW:

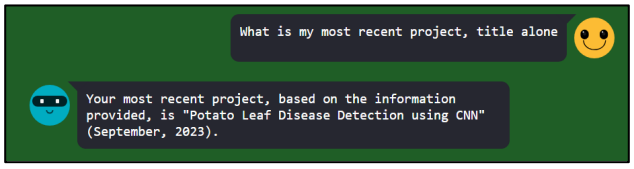


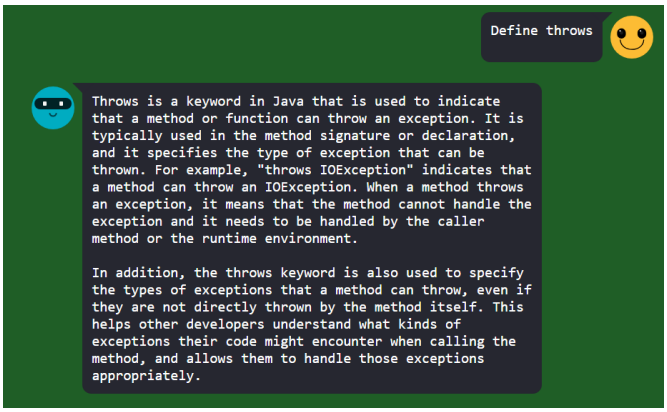
For demonstration, we have uploaded two documents, 1. A Data Scientist Resume, 2. Interview Questions in Java

PROMPT AND RESPONSE GENERATION:

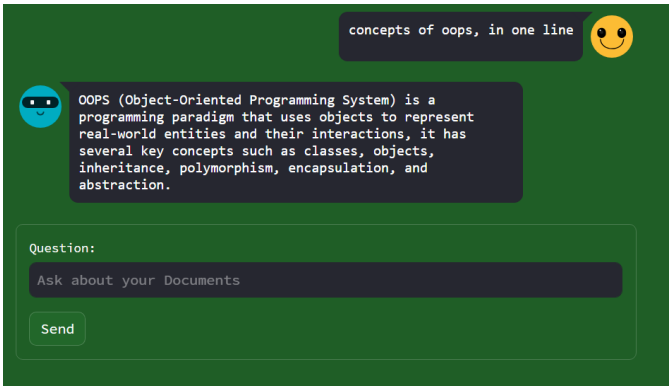


As we can see that it has identified the Data science projects from the uploaded resume and if we prompt more precisely, we get accurate results from the document.

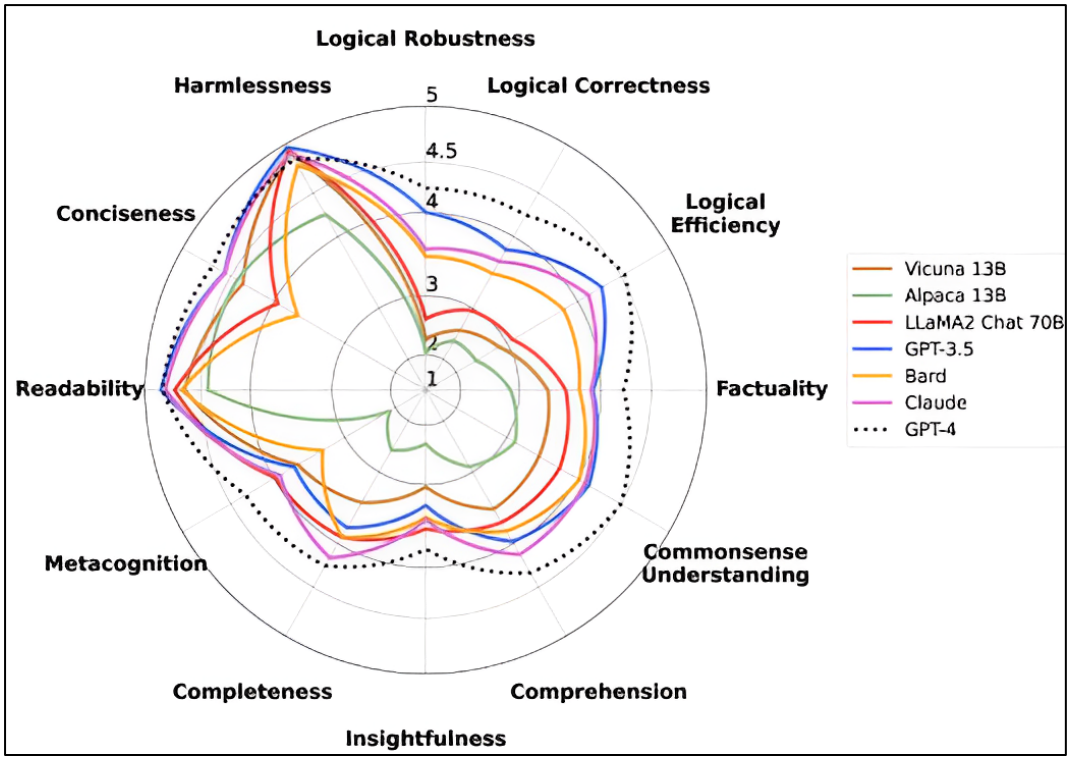




Here, the model utilizes both the uploaded document and it's pretrained data to answer these questions.



LARGE LANGUAGE MODEL COMPARISON:



V. CONCLUSION

The proposed solution addresses challenges in deploying generative AI on custom data with high computing power requirements. Leveraging Meta's Llama 2-70B model, pre-trained on a vast dataset, the approach involves processing user documents in multiple formats and converting raw data into embeddings. These embeddings serve as references to generate responses to user prompts, considering both user input and pre-existing data. The solution, facilitated through cloud access and API usage, is billed based on the number of embeddings and prompts requested. Overall, this approach provides a clear pathway to overcome obstacles in deploying generative AI on personalized data, showcasing a practical method for enhanced customer support and content generation.

REFERENCES

[1] Research on Query Optimization of Classic Art Database Based on Artificial Intelligence and Edge Computing - Xiang Dong and Lijia Zeng

[2] Introduction to Query Processing and Optimization - Michael L. Rupley, Jr. Indiana University at South Bend

[3] Neo: A Learned Query Optimizer - Ryan Marcus , Parimarjan Negi , Hongzi Mao , Chi Zhang, Mohammad Alizadeh, Tim Kraska, Olga Papaemmanouil1, Nesime Tatbul Brandeis University MIT Intel Labs.

[4] New Word Extraction From Chinese Financial Documents - Liwei Yan, Bo Bai, Member, IEEE, Wei Chen, Senior Member, IEEE, and Dapeng Oliver Wu, Fellow, IEEE

[5] New Avenues for Automated Railway Safety Information Processing in Enterprise Architecture: An NLP Approach - Abdul Wahab Qurashi and Anju P. Johnson

[6] A Novel Unsupervised Approach for Topic Prediction Using Keyphrase Extraction for Urdu Documents - Ahmad Amin; Toqir A. Rana; Natash Ali Mian; Muhammad Waseem Iqbal; Abbas Khalid; Tahir Alyas.

[7] Analysis of Search-Engine Query Patterns - Louis R Pasquale, MD Department of Ophthalmology Icahn School of Medicine at Mount Sinai

[8] Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling - Stella Biderman Hailey Schoelkopf Quentin Anthony Herbie Bradley Kyle O'Brien Eric Hallahan Mohammad Aflah Khan Shivanshu Purohit USVSN Sai Prashanth Edward Raff Aviya Skowron Lintang Sutawika Oskar van der Wal

[9] Review of Extraction and Classification of Key-Phrases in Scientific Publications using CRF and WEDP - Riya Tyagi.

[10] Is Chat GPT Biased? A Review - Sahib Singh, Ford Motor, Michigan, USA. Narayanan Ramakrishnan, Nordstrom Inc., Washington, USA