AI & Investment Banking

Technology Driven Solutions

A Project Report

On

InvestMate : An LLM Chatbot for Investment Bankers

Exploring GenAI, LLM Inference, and Fine-Tuning Techniques

Prepared By

Deepti Agarwal

GLA University, Mathura

Abstract

The development of an investment banking chatbot represents a significant advancement in leveraging Generative Artificial Intelligence (GenAI) to address the complexities of financial sector queries. This project focuses on creating a highly efficient and responsive chatbot using Intel's Neural Chat LLM and incorporating Retrieval Augmented Generation (RAG) technology. The chatbot is designed to handle intricate investment banking queries, providing detailed and contextually accurate responses.

The technical approach includes the integration of Langchain as the orchestration framework, Intel's 2-bit quantized LLM for efficient CPU inference, BGE Embeddings for precise context generation, and Chroma DB as the vector store for robust data handling. The backend is developed using Flask and FastAPI to ensure flexibility and scalability, while the frontend offers a user-friendly conversational interface.

The results demonstrate the chatbot's high performance, robust data handling, and user-friendly interface, showcasing its potential to revolutionize interactions in the financial sector. By incorporating comprehensive investment banking-related data sources, the chatbot delivers thorough and accurate information, enhancing the user experience.

This report provides an in-depth analysis of the problem statement, technical approach, and results, highlighting the innovative use of advanced AI technologies in developing a custom investment banking chatbot. The project serves as a testament to the transformative capabilities of GenAI in automating complex financial queries with precision and efficiency.

1. Problem Statement

Introduction to GenAI and Simple LLM Inference on CPU and Fine-Tuning of LLM Model to Create a Custom Chatbot

Generative Artificial Intelligence (GenAI) has the potential to revolutionize various sectors by automating tasks and enhancing user interactions. However, its application in the financial sector, particularly investment banking, is limited due to the complexity of queries and the need for precise responses. This project aims to address these challenges by developing an "Investment Banker RAG Chatbot" using Intel's Neural Chat LLM Model : neural-chat-7b-v3-1.

2. Description of Problem Statement

This problem statement introduces beginners to GenAI through hands-on exercises, including simple Large Language Model (LLM) inference on a CPU and the fine-tuning of an LLM model to create a custom chatbot. The chatbot will handle investment banking queries, providing detailed and relevant responses efficiently.

3. Technical Approach

3.1 Conceptualization and Technology Selection

The project started with identifying the need for an investment banking chatbot and defining key objectives. Intel’s Neural Chat LLM was selected for its efficiency and high performance on CPU. The core technology, Retrieval Augmented Generation (RAG), was chosen to enhance the chatbot's response accuracy by combining LLM capabilities with a retrieval mechanism to pull in relevant context.

3.2 Framework and Orchestration

Langchain was used as the orchestration framework to manage the integration of various components. This framework ensures seamless operation and coordination among the chatbot’s multiple technologies, facilitating the development of a sophisticated yet efficient system.

3.3 Model Integration

The project integrated Intel’s 2-bit quantized LLM from Huggingface, optimized for CPU inference. This model was fine-tuned specifically to handle the intricacies of investment banking queries. The use of a 2-bit quantized model ensures that the system can operate efficiently on limited hardware resources, making it accessible for a wider range of users and applications.

3.4 Embedding and Vector Store Implementation

BGE Embeddings were incorporated to generate precise context for responses. These embeddings capture the semantic meaning of the input data, enabling the chatbot to understand and respond accurately to user queries. Chroma DB was utilized as the vector store to manage and retrieve complex query data efficiently. This setup ensures that the system can handle large volumes of data and complex queries seamlessly.

3.5 Frontend Development

The frontend was designed to be user-friendly, offering an intuitive conversational interface for seamless interaction. This ensures that users can interact with the chatbot effortlessly, enhancing their overall experience. The frontend also integrates with the backend to fetch responses in real-time, providing a smooth and responsive user interface.

3.6 Backend Development

The backend was developed using Flask and FastAPI, providing a flexible and robust infrastructure. Flask was used for its simplicity and ease of use, while FastAPI was chosen for its high performance and scalability. This dual approach ensures that the backend can handle a large number of concurrent requests and provide quick responses.

3.7 Data Integration

Investment banking-related PDFs were processed and indexed as primary data sources. This involved using the PyPDFLoader to load documents from the 'Data/' directory and the RecursiveCharacterTextSplitter to split these documents into manageable chunks. This ensured that the chatbot had comprehensive and reliable information to draw from when responding to queries. The use of PDFs as primary data sources ensures that the chatbot can provide detailed and accurate information on a wide range of investment banking topics.

4. Results

4.1 High Performance and Efficiency

The chatbot, powered by Intel’s 2-bit quantized LLM, demonstrated high performance and efficiency in handling investment banking queries on a CPU. The integration of RAG technology ensured that responses were detailed and contextually accurate, significantly improving the user experience. The model's quantization allowed for efficient processing, making it feasible to run the chatbot on standard CPU hardware without sacrificing performance.

4.2 Robust Data Handling

Chroma DB effectively managed the embeddings and structured data, allowing the chatbot to handle complex queries seamlessly. BGE Embeddings provided precise context, enhancing the quality of responses. This robust data handling capability ensured that the chatbot could manage and retrieve large volumes of data quickly, providing users with accurate and relevant answers.

4.3 User-Friendly Interface

The frontend design resulted in an intuitive and engaging user experience. Users found the chatbot easy to interact with, and the conversational interface facilitated smooth and efficient communication. The frontend was designed to be responsive and user-friendly, ensuring that users could easily input their queries and receive responses in real-time.

4.4 Flexible Backend Infrastructure

The dual backend approach using Flask and FastAPI provided a robust and scalable infrastructure. This flexibility allowed for efficient handling of user queries and seamless integration of various components. The Flask framework was used for its simplicity and ease of setup, while FastAPI provided high performance and scalability, making the backend capable of handling high loads and concurrent requests.

4.5 Comprehensive Information

By incorporating PDFs as primary data sources, the chatbot could offer thorough insights on investment banking topics. This ensured that users received accurate and detailed information in their responses. The use of PDFs allowed the chatbot to access a wide range of information, providing users with comprehensive answers to their queries.

4.6 Scalability and Future Enhancements

The architecture of the chatbot ensures scalability, making it suitable for extensive use. Future enhancements are planned to include more data sources and advanced AI capabilities, ensuring the chatbot remains relevant and effective. The system's design allows for easy integration of additional features and improvements, ensuring that the chatbot can evolve and adapt to meet the changing needs of users.

5. Chatbot Interface & Outputs

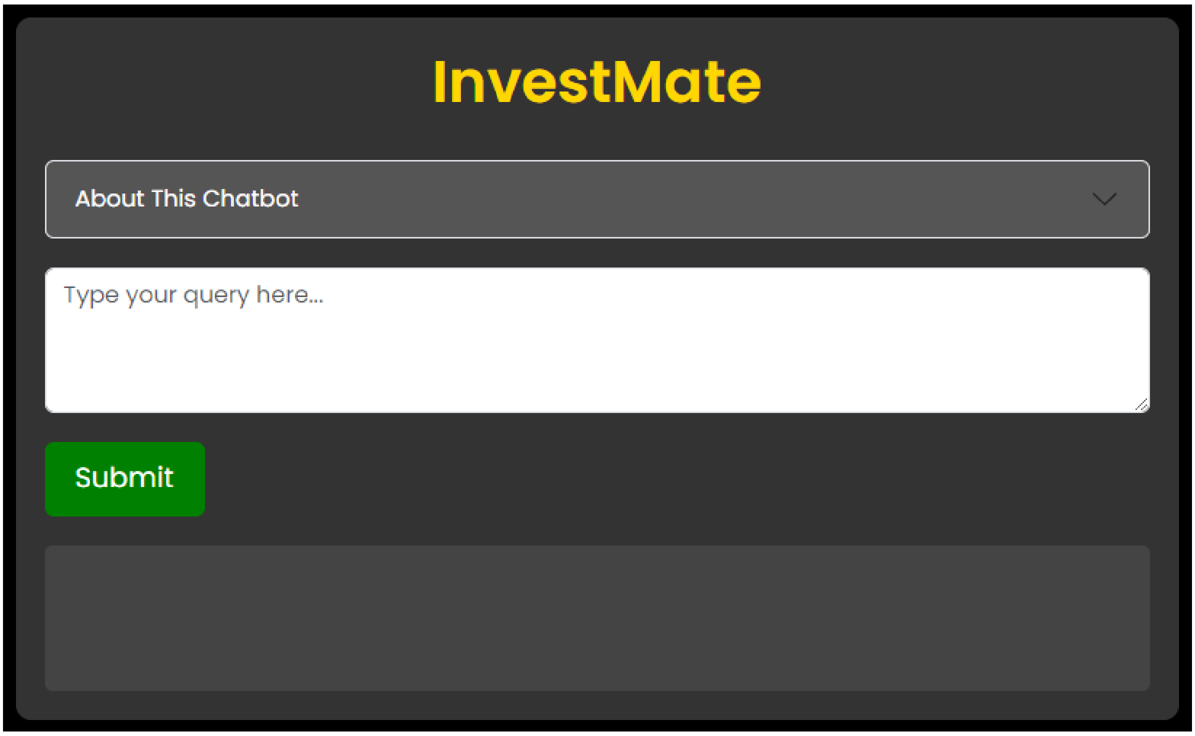
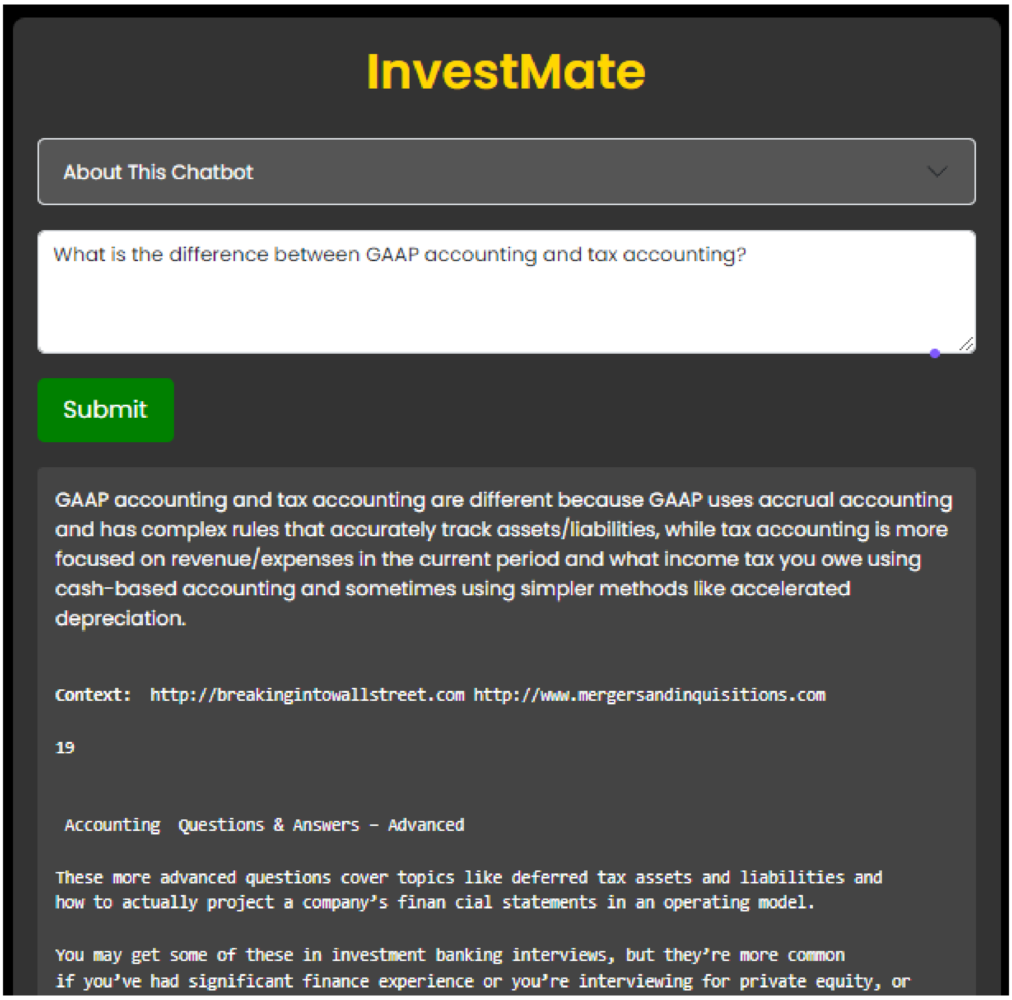


Fig 1 : Chatbot Interface

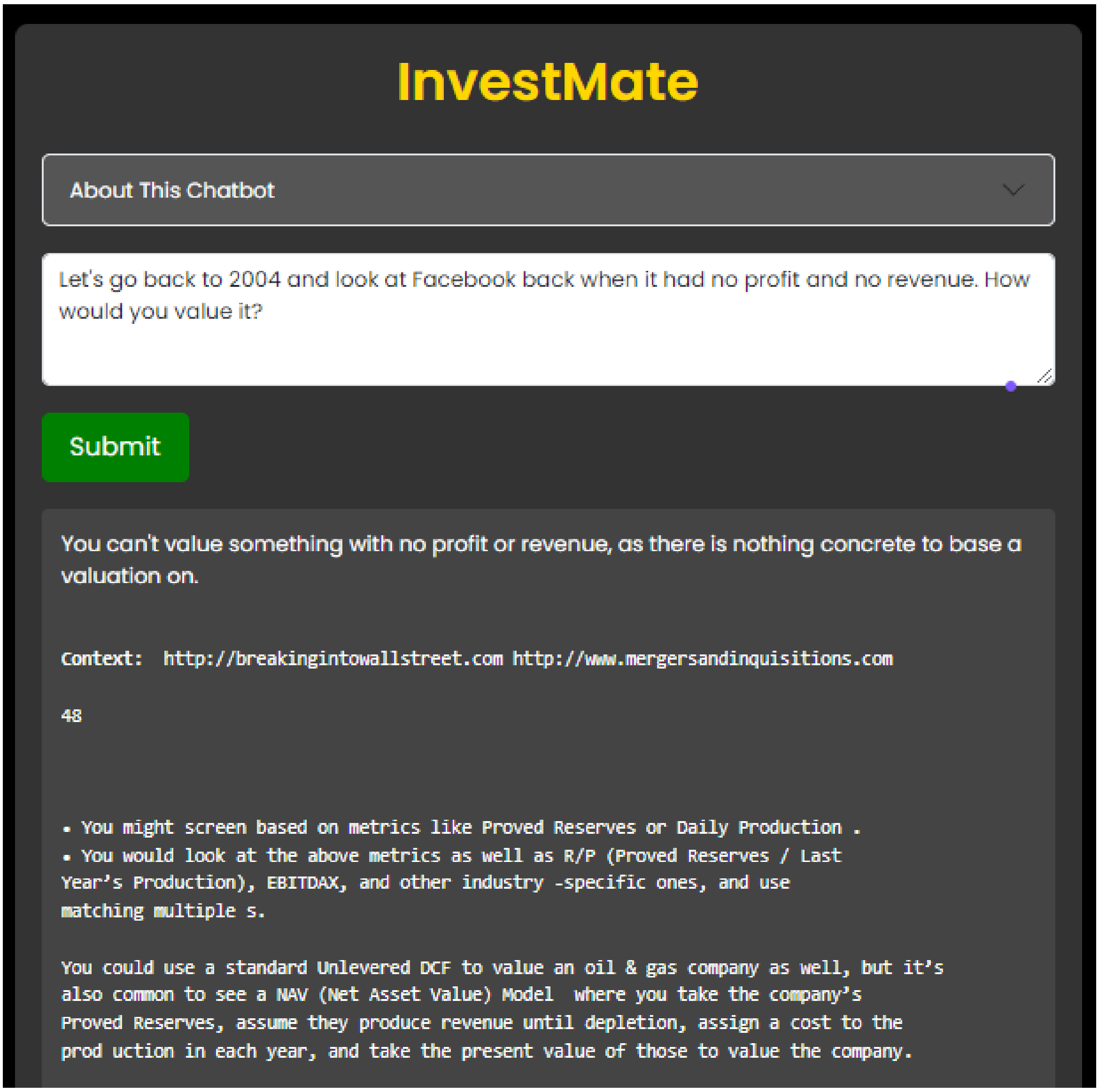


What is the difference between GAAP accounting and tax accounting?

GAAP accounting and tax accounting are different because GAAP uses accrual accounting and has complex rules that accurately track assets/liabilities, while tax accounting is more focused on revenue/expenses in the current period and what income tax you owe using cash-based accounting and sometimes using simpler methods like accelerated depreciation.

```
Context:  http://breakingintowallstreet.com http://www.mergersandinquisitions.com

19


 Accounting  Questions & Answers – Advanced

These more advanced questions cover topics like deferred tax assets and liabilities and
how to actually project a company's finan cial statements in an operating model.

You may get some of these in investment banking interviews, but they're more common
if you've had significant finance experience or you're interviewing for private equity, or
```

Fig 2 : Sample Output 1



Let's go back to 2004 and look at Facebook back when it had no profit and no revenue. How would you value it?

You can't value something with no profit or revenue, as there is nothing concrete to base a valuation on.

```
Context:  http://breakingintowallstreet.com http://www.mergersandinquisitions.com

48



• You might screen based on metrics like Proved Reserves or Daily Production .
• You would look at the above metrics as well as R/P (Proved Reserves / Last
Year's Production), EBITDAX, and other industry -specific ones, and use
matching multiple s.

You could use a standard Unlevered DCF to value an oil & gas company as well, but it's
also common to see a NAV (Net Asset Value) Model  where you take the company's
Proved Reserves, assume they produce revenue until depletion, assign a cost to the
prod uction in each year, and take the present value of those to value the company.
```

Fig 3 : Sample Output 2

6. Conclusion

As the sole developer, I integrated advanced AI technologies to create a robust investment banking chatbot. The project showcased the potential of GenAI in handling complex financial queries with precision and efficiency. Leveraging Intel's Neural Chat LLM, Langchain, RAG technology, and Chroma DB, the chatbot delivered a seamless user experience. Moving forward, I plan to optimize and expand the chatbot's capabilities, incorporating user feedback and the latest advancements in AI to maintain its effectiveness in the dynamic field of investment banking.