From Data to Dialogue

Leveraging LLMs for Financial Analytics & Risk Management

FINM 31006: Project Lab
MS Financial Mathematics
The University of Chicago
Dec 18th, 2024

BANK OF AMERICA

THE UNIVERSITY OF CHICAGO

Project Objective

Background

AI has emerged as a transformative force in the rapidly-evolving financial industry, enabling firms to automate processes and enhance risk management.

Problem

There is a significant accessibility barrier for users without technical expertise, particularly when tasked to query data from large databases and perform technical analysis.

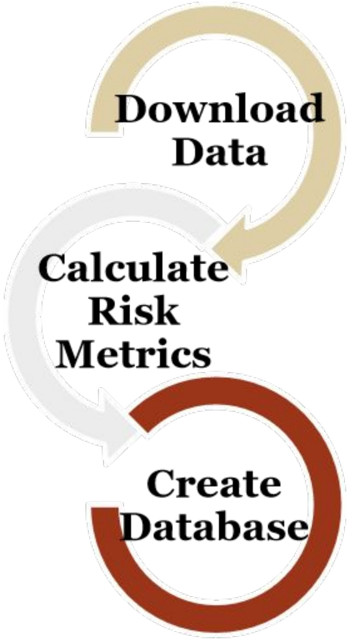
Solution

Develop an interactive AI-powered financial chatbot that translates user questions into SQL queries, extracts data, and acts as a report copilot tool that produces easy access technical insights.

Data Overview

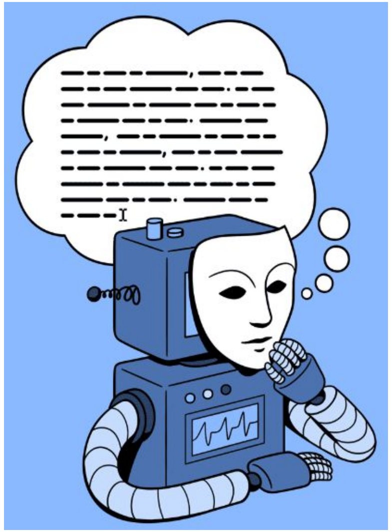
Retrieve quarterly US equities financial data from ***2006-2024*** from Nasdaq's Quandl using an API key.

- Pre-process and merge the downloaded datasets.
- Compute key financial metrics such as Price-to-Earnings (PE) ratio, Debt-to-Equity ratio, etc.
- Store processed data into a structured local database for execution.



What are Large Language Models (LLMs)?

- LLMs are AI models based on deep learning, designed to generate, understand, and process human language.
- LLMs are trained on vast amounts of text data to predict, analyze, and transform language patterns.
- Developed by OpenAI, ***ChatGPT-4o*** is a widely known implementation of LLMs.



What is Prompt Engineering?

- The practice of crafting and optimizing input prompts that guide LLMs to produce accurate, relevant, and desired outputs.
- Effective prompts improve the quality, efficiency, and precision of responses.

Eg. **Input**: “Make me a profitable investment portfolio.”

- **Output**: Low return savings account with minimal growth

Eg. **Input**: “Create a diversified portfolio with 80% allocated to equities and 20% allocated to fixed income products that achieves an annual return > 10%.”

- **Output**: A tailored portfolio that better aligns with your financial goals and risk tolerance

What is Prompt Engineering?

1
Provide Database Schema

Table Names, Column Names, and Keys

“Tell the analyst what data we have”

2
Few-Shot Prompting

Sample questions with both correct and incorrect SQL Query Outputs

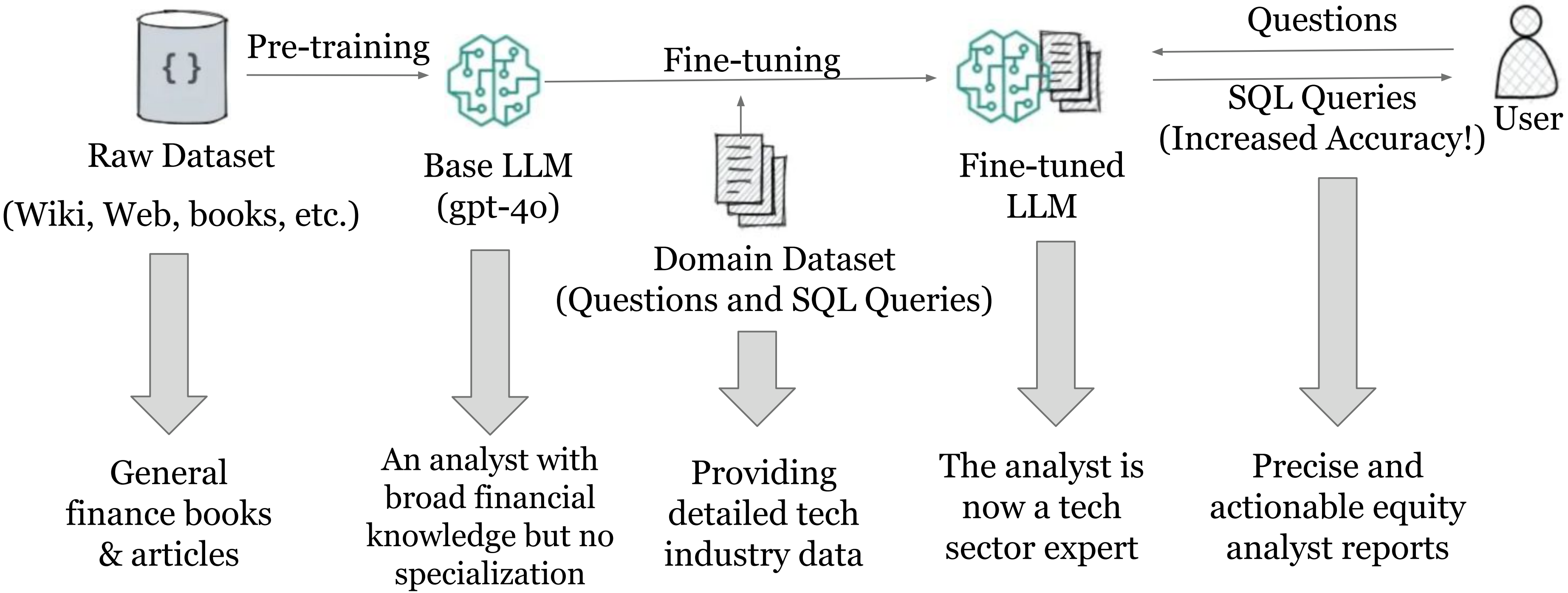
“Give the analyst some instructions on analyzing financials”

3
Chain of Thought (CoT)

Ask the model to “Think Sequentially”

“Ask the analyst to break the report into sections”

What is Fine-Tuning?

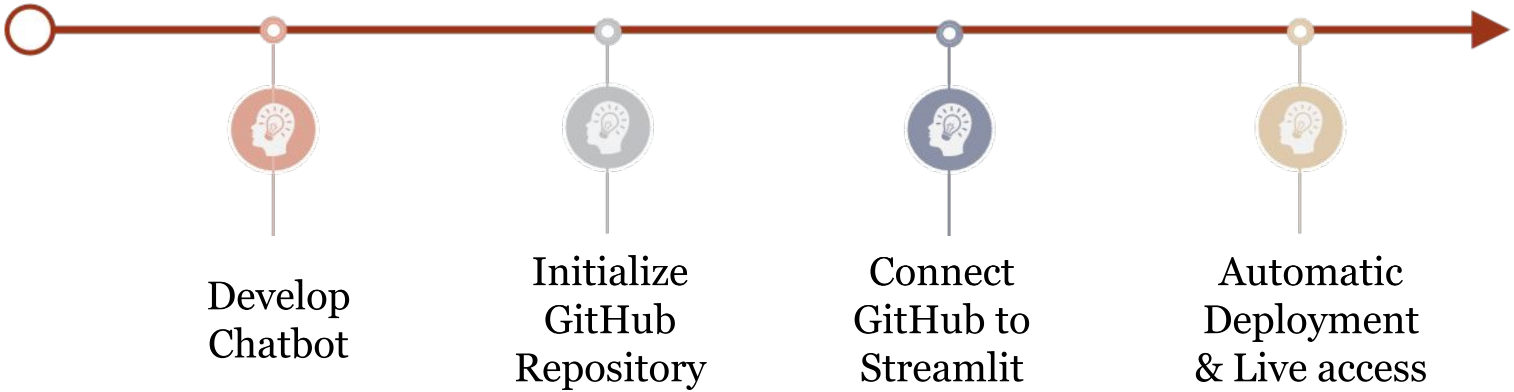


Model Performance Evaluation

Model	Accuracy
Original ChatGPT 4o Model	61%
Prompt-engineered ChatGPT 4o Model	78%
Fine-tuned ChatGPT 4o Model	**89%**

- Created **100 prompts** with corresponding SQL queries for initial evaluation
- Fine-tuned the prompt-engineered model using these 100 prompts
- Created **75 new prompts** to evaluate performance of the fine-tuned model

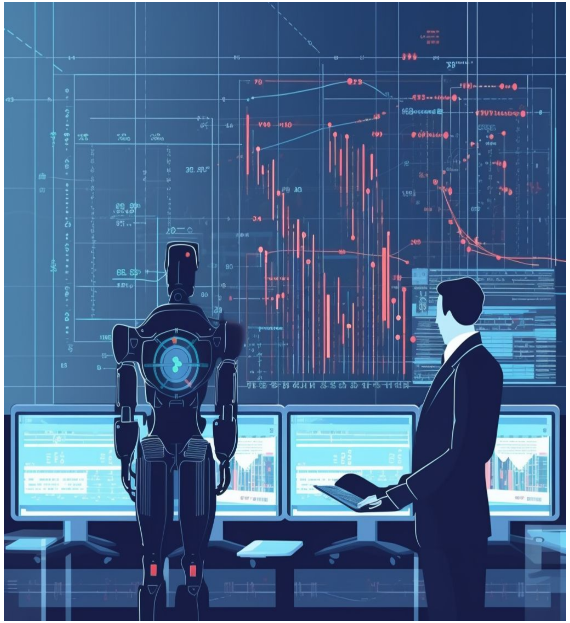
Streamlit: Online Chatbot Deployment



- **Streamlit** is an open-source platform that enables developers to quickly build and deploy interactive web applications.
- By integrating the platform with GitHub, we can automatically host our chatbot online for others to use.

Future Directions

- Further minimize model errors
- Increase data sources and time range of data
- Incorporate interactive data visualization functionality
- Improve the model's ability to interpret complex user prompts and understand nuanced financial queries



Model Applications in Banking

Benefits

- **Accessibility**: Empowers users without technical expertise to query and analyze data.
- **Efficiency**: Reduces time for data collection and analysis.

Challenges

- **Limited Utility**: Current chatbot is unable to handle complex tasks.
- **Data Privacy**: Ensuring sensitive financial data remains secure.
- **Model Reliability**: Maintaining consistent and accurate outputs for complex queries.
- **Integration**: Seamlessly incorporating model into existing systems and workflows.

Questions

Appendix

- **Data Source**: Quandl (The Zacks Fundamentals Collection A)
- **Data Coverage**: Quarterly data (2006-2024), 9,500+ U.S. and Canadian equities, 10,000+ delisted stocks
- **Database Schema**:
    - Financial Core (*t_zacks_fc*): EPS, Net long-term debt, Company information
    - Financial Ratios (*t_zacks_fr*): ROI, Debt-to-equity ratios
    - Market Value (*t_zacks_mktv*): Market capitalization
    - Shares Outstanding (*t_zacks_shrs*): Shares outstanding
    - Sector Description (*t_zacks_sectors*): Financial sector codes

Appendix

Generate a SQL query based on the following natural language question and database schema.
Natural Language Question:
“ Insert “
Database Schema:
Table 1: t_zacks_fc
- Columns: 'ticker' = Zacks Identifier, ticker or trading symbol, 'comp_name' = Company name, 'exchange' = Exchange traded, 'per_end_date' = Period end date, 'per_type' = Period type, 'filing_date' = Filing date, 'filing_type' = Filing type: 10-K, 10-Q, PRELIM, 'zacks_sector_code' = Zacks sector code, 'eps_diluted_net_basic’ = Earnings per share (EPS) (diluted) net, 'lterm_debt_net_tot' = Net long-term debt.
- Keys: ticker, per_end_date, per_type

Table 2: t_zacks_fr
- Columns: 'ticker' = Zacks Identifier, ticker or trading symbol, 'per_end_date' = Period end date, 'per_type' = Period type, ‘ret_invst’ = Return on investments, ‘tot_debt_tot_equity’ = Total debt / total equity.
- Keys: ticker, per_end_date, per_type.

Table 3: t_zacks_mktv
- Columns: 'ticker' = Zacks Identifier, ticker or trading symbol, 'per_end_date' = Period end date, 'per_type' = Period type, ‘mkt_val’ = Market Cap (shares out x last monthly price per share).
- Keys: ticker, per_end_date, per_type.

Table 4: t_zacks_shrs
- Columns: 'ticker' = Zacks Identifier, ticker or trading symbol, 'per_end_date' = Period end date, 'per_type' = Period type, ‘shares_out’ = Common Shares Outstanding from the front page of 10K/Q.
- Keys: ticker, per_end_date, per_type.

Appendix

Few-shot prompting with examples for each prompt type:

Prompts that produce SQL queries with one value answers:
Example prompt : Output ticker with the largest market value recorded on any given period end date.
Correct output: SELECT ticker, per_end_date, MAX(mkt_val) AS max_market_value FROM t_zacks_mktv GROUP BY per_end_date ORDER BY max_market_value DESC LIMIT 1;
Incorrect output: SELECT MAX(mkt_val) , ticker FROM t_zacks_mktv GROUP BY ticker

Prompts that produce SQL queries that filter one table:
Example prompt : Filter t_zacks_fc to only show companies with a total debt-to-equity ratio greater than 1.
Correct output: SELECT * FROM t_zacks_fr WHERE tot_debt_tot_equity > 1;
Incorrect output: SELECT * FROM t_zacks_fr WHERE t_zacks_mktv > 1;

Prompts that produce SQL queries that require merging and filtering two or more tables:
Example prompt : Combine t_zacks_mktv and t_zacks_shrs to show tickers with market cap and shares outstanding in the latest period end date.
Correct output and Incorrect Output provided

Appendix

Original Prompt			
Model	Correct Outputs	Total Outputs	Accuracy
ChatGPT 4o	61	100	61%
Llama 3.2 3B	25	100	25%

Modified Prompt			
Model	Correct Outputs	Total Outputs	Accuracy
ChatGPT 4o	78	100	78%
Llama 3.2 3B	31	100	31%
Llama 3.1 8B	30	100	30%

After thoroughly comparing the accuracy of both original and modified prompts, we chose to use the **ChatGPT 4o model** for fine-tuning.

Disadvantages of Llama compared to GPT:

- Lack of Memory
- Slow Response Time
- Poor Context Recognition
- Limited Prompt Tuning Effectiveness

Appendix

Create a fine-tuned model

Base Model

gpt-4o-2024-08-06

Training data

Add a jsonl file to use for training. By providing the file, you confirm that you have the rights to use the data.

Upload new | Select existing (selected) | Browse files ↗

file-SRonJgQiu6qiDy4hcQykMf

Validation data

Add a jsonl file to use for validation metrics.

Upload new | Select existing | None (selected)

Suffix

Add a custom suffix that will be appended to the output model name.

my-experiment

Seed

The seed controls the reproducibility of the job. Passing in the same seed and job parameters should produce the same results, but may differ in rare cases. If a seed is not specified, one will be generated for you.

Random

Configure hyperparameters

Batch size ⓘ	auto
Learning rate multiplier ⓘ	auto
Number of epochs ⓘ	auto

Learn about fine-tuning ↗ | Cancel | Create

ft:gpt-4o-2024-08-06:personal::AYFZ3Shk

Status	Succeeded
Job ID	ftjob-eUVtudxoElzy5CasgN1VqecF
Base model	gpt-4o-2024-08-06
Output model	ft:gpt-4o-2024-08-06:personal::AYFZ3Shk
Created at	2024年11月27日 10:44
Trained tokens	767,120
Epochs	10
Batch size	1
LR multiplier	2
Seed	1112430405

Checkpoints

ft:gpt-4o-2024-08-06:personal::AYFZ2NSa:ckpt-step-80

ft:gpt-4o-2024-08-06:personal::AYFZ3CLE:ckpt-step-90

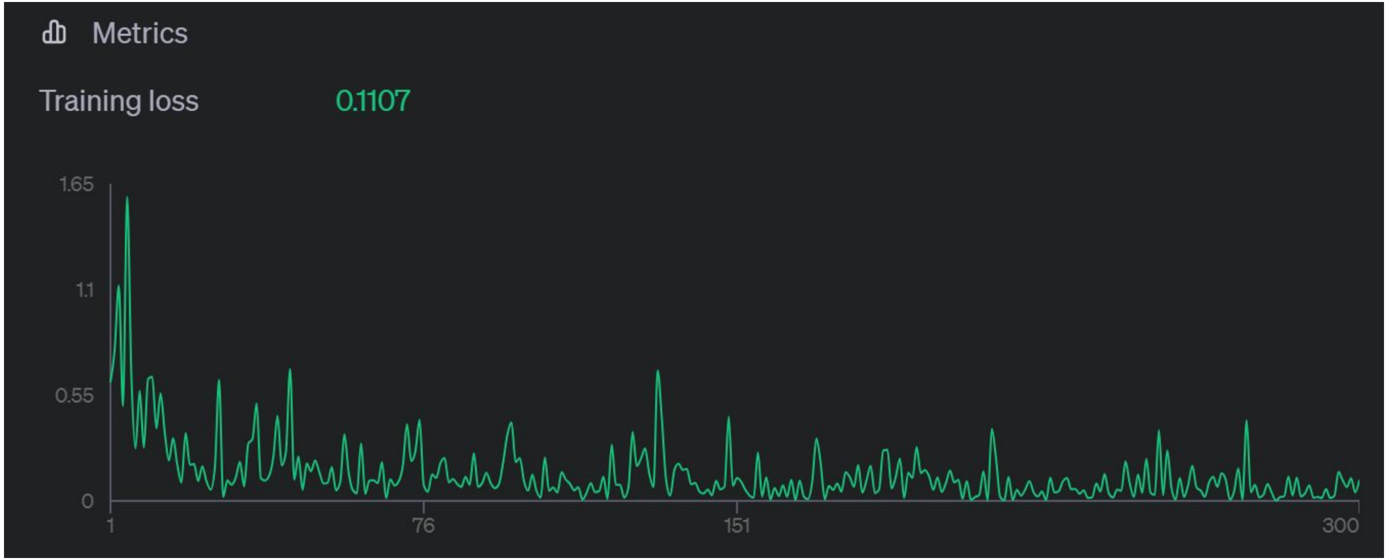
ft:gpt-4o-2024-08-06:personal::AYFZ3Shk

Files

Training plzzz.jsonl

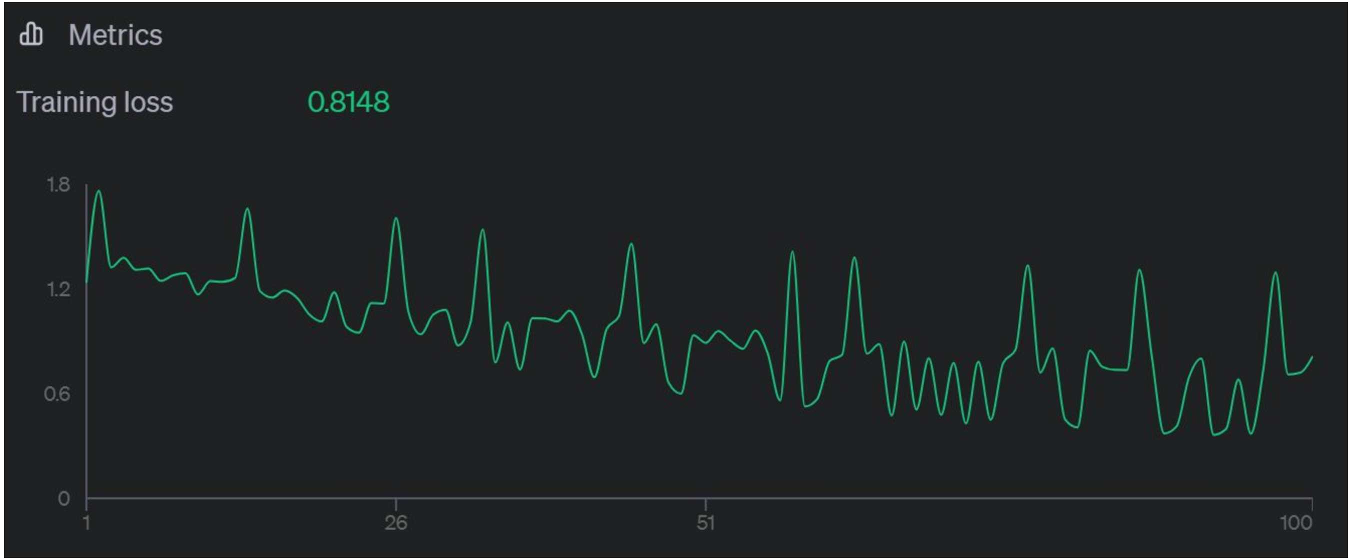
Appendix

Training Loss of SQL Query Generator Model



Appendix

Training Loss of PDF-Generator Model (No convergence)



Thank you!

BANK OF AMERICA

THE UNIVERSITY OF CHICAGO