Arthur Poon
Generative AI
Professor Michael Spertus
March 6, 2024

Final Project: Investigating LLM Performance on Financial Analysis Questions with RAG using PDF vs XBRL SEC Filings

Introduction:

In recent years, the advent of advanced Large Language Models (LLMs) such as GPT-4 has revolutionized the landscape of automated text understanding and generation, offering profound implications for various sectors, including finance. Among the plethora of applications, the analysis of financial documents to extract insights and make predictions stands out as a critical area of interest. This paper delves into the comparative performance of LLMs, specifically in the context of financial analysis questions, by utilizing two distinct formats of SEC filings: the traditional Portable Document Format (PDF) and the eXtensible Business Reporting Language (XBRL).

The Securities and Exchange Commission (SEC) filings are a gold mine of information for financial analysis, containing detailed reports of a company's financial performance, operations, and future outlook. Traditionally, these filings have been available in PDF format, which, while universally accessible, presents challenges for automated data extraction due to its fixed layout and non-structured nature. On the other hand, XBRL, a more recent and structured format, offers machine-readable data, potentially simplifying the process of automated analysis by providing standardized tags for financial terms and concepts.

This investigation is inspired by the evolving landscape of financial data analysis and the critical examination of how well AI models, particularly LLMs like GPT-4, understand and process XBRL data. Our research aims to assess the efficacy of LLMs equipped with Retrieval-Augmented Generation (RAG) techniques in interpreting and answering financial analysis questions. By comparing the performance of these models on SEC filings presented in PDF versus XBRL formats, we seek to uncover insights into the capabilities and limitations of current AI technologies in financial document analysis.

The significance of this study lies not only in advancing our understanding of LLMs' abilities to interact with complex financial datasets but also in exploring the potential of XBRL to revolutionize financial reporting and analysis. As we embark on this exploration, we anticipate that our findings will contribute to the broader discourse on the integration of AI in finance, offering valuable insights for academics, industry practitioners, and policymakers alike.

On January 18, 2024, the XBRL organization released an article titled “How well do AI models like GPT-4 understand XBRL Data?”. In the article, the author utilizes XBRL reports loaded in an xBRL-JSON format uploaded to GPT-4 to show that performance and correctness of financial questions increases significantly.[1]

On November 15, 2023, a startup Patronus AI launched “FinanceBench” the industry’s first benchmark for testing how LLMs perform on financial questions. They found that models like GPT-4 get answers incorrectly up to 81% of the time.[2]

In this project, I want to systematically evaluate whether utilizing XBRL-formatted documents in a RAG-format can systematically increase the performance of models like GPT based on the FinanceBench dataset.

Description of FinanceBench dataset:

FINANCEBENCH is a first-of-its-kind test suite for evaluating the performance of LLMs on open book financial question answering (QA). It comprises 10,231 questions about publicly traded companies, with corresponding answers and evidence strings. The questions in FINANCEBENCH are ecologically valid and cover a diverse set of scenarios. They are intended to be clear-cut and straightforward to answer to serve as a minimum performance standard. The Patronus team tested 16 state of the art model configurations (including GPT-4-Turbo, Llama2 and Claude2, with vector stores and long-context prompts) on a sample of 150 cases from FinanceBench and manually review their answers. These 150 cases are available on HuggingFace.[3]

Project Approach to Performance evaluation/Dataset:

Due to the fact that answers need to be manually-evaluated, I limit the queries in the dataset on which I will be evaluating performance. In particular, I limit the queries of interest to ‘metrics-generated’ questions which are questions that ask a model to perform a calculation based on multiple data points in a filing, or to directly pull an answer from the filing without calculation.

Furthermore, in order to evaluate apples-to-apples model performance on PDF documents versus XBRL documents, I only evaluate the queries that are aiming to pull from a filing that has both PDF and XBRL versions of an SEC filing (XBRL was only systematically deployed in 2019, so some of the older filings in the FinanceBench dataset pre-2019 did not have corresponding XBRL instances to pull).

---

[1] https://www.xbrl.org/how-well-do-ai-models-like-gpt-4-understand-xbrl-data/
[2] https://www.patronus.ai/announcements/patronus-ai-launches-financebench-the-industrys-first-benchmark-for-llm-performance-on-financial-questions
[3] https://huggingface.co/datasets/PatronusAI/financebench/viewer/default/train

After filtering for ‘metrics-generated’ questions and queries that target documents with both PDF and XBRL documents, I am left with a dataset of 29 rows. These can be found in the file “FinanceBench_XBRL_subset.csv”.

Procurement of XBRL data:

I follow the methodology that was conducted in the XBRL article to save xBRL-JSON formats of SEC filings utilizing the Arelle framework. The script used to do this is in the file “XBRL_JSON_converter.py”.

RAG-Models Output Generation:

I create a RAG-system with PDF documents of SEC filings to evaluate the performance of gpt-3.5-turbo-instruct on the subset questions from FinanceBench. As expected, it only gets 5 of 29 correct, yielding a 17% accuracy rate and a 83% error rate, in-line with what Patronus reported in their study. The code and output for this generation can be found in “small_RAG_SEC_Filings.ipynb”. The manual review of results can be found in “pdf_query_answers.xlsx”.

I then create a RAG-system with the XBRL-JSON documents to evaluate the performance of gpt-3.5-turbo-instruct on the subset of questions from FinanceBench. Surprisingly, the model is unable to produce an output for any of the questions, only giving back errors. This made me wonder if this is an issue with the model. So I tried with gpt-4-0125-preview; however, the outputs were all errors. The output with XBRL-JSON RAG setup can be found in the file “small_RAG_SEC_Filings_XBRL_JSON.ipynb”.

This led me to try a series of other models, specifically models built to specifically understand code since the XBRL-JSON format is technically a form of code as well. To do this, I leveraged Together.AI’s platform to test a series of code-specific models and several alternative open-source models to see if they would perform better with the XBRL-formatted filings. For the alternative models, I chose models that were relatively well-known and frequently used. The models I chose to test were the following:

Code-specific models:

- deepseek-ai/deepseek-coder-33b-instruct"
- 'codellama/CodeLlama-70b-hf'
- "Phind/Phind-CodeLlama-34B-v2"

Alternative open-source models:

- "lmsys/vicuna-13b-v1.5"
- "togethercomputer/alpaca-7b"
- "togethercomputer/StripedHyena-Nous-7B"
- "allenai/OLMo-7B-Instruct"

- "google/gemma-7b-it"
- "NousResearch/Nous-Hermes-2-Mistral-7B-DPO",
- "mistralai/Mistral-7B-Instruct-v0.2"
- "mistralai/Mixtral-8x7B-Instruct-v0.1"
- "snorkelai/Snorkel-Mistral-PairRM-DPO"

Unfortunately, this did not improve the performance, and still all the models faired to answer any of the questions. The outputs for these models can be found in I also noticed that the answers from the various models were producing almost exactly the same answers. Which I realized were attributable to the fact that the embeddings were all the same OpenAI embeddings. Thus, I wanted to change the embeddings model; however, the Together.AI embedding models were retaining errors stating that I was sending too many requests to the server. Thus, I was not able to test out the performance of models with different embedding models to see how that affects model performance on financial questions. The outputs for these alternative models in XBRL-JSON formatted documents can be found in the file "small_RAG_SEC_Filings_XBRL_Together.ipynb".

Conclusion:

To conclude, based on my findings, utilizing XBRL-JSON formatted documents to perform RAG-augmented financial queries does not improve the performance over text-based documents. However, this is only based on investigation utilizing OpenAI's embeddings models and not tested with other embedding models.

Next Steps:

I tried to utilize Together.AI's embedding models to test if using a different embedding model would improve the performance of the models. However, I hit a rate limit error and was not able to implement the Together.AI embeddings in this project. I reached out to their support team to get an indication of the sizing of my request, but I never received an answer in time. If I had more time, I would test a variety of embedding models to see if the performance would improve under different embedding models.