Title:

Reading and Understanding Annual Reports (SEC 10-K Filings) using LLM and RAG Method

Goal:

Corporate annual reports are vital for financial decision-making, providing comprehensive information about a company's financial health. Analyzing these reports, particularly SEC Form 10-K filings, can be challenging due to their length and complexity. This thesis aims to develop a method, leveraging Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG), to facilitate the easy reading and understanding of financial annual reports. The goal is to create a system that delivers accurate, contextually relevant responses, making the information accessible to a broader audience.

Components of the System:

1. Reports Materials. Utilizing 10-K annual SEC filings stored in a vector database for efficient retrieval and generation.

There are several ways to get and use the data:

* Use EDGAR-CORPUS dataset with 10-K filings from 1999 to 2020 years. The main drawbacks are that all the tables from the reports were deleted when dataset had been created and the most recent reports are missing (from 2021 to 2023).
* We can scrape EDGAR website using code from edgar-crawler toolkit (toolkit was used for EDGAR-CORPUS dataset creating)
* Download the filings from EDGAR website with own code (recent reports are in HTML format and older ones are TXT files) and reformat them into a plain text. This method is the most flexible one but the most labor-intensive. There are more than 180K 10-K filings uploaded into the EDGAR since 1993. In our opinion, we can use randomly select 10 reports, convert HTML files into the plain text, clean data, then customize and test the system on them. After the system can reliably answer the questions, select 10 more reports and test the system on them.

1. Reader’s Question. Natural language queries encoded into vector representations using pre-trained embedding models.

We have prepared a list of 42 questions that are most commonly used in analyzing reports.

1. The System. The core component integrating LLM (Llama 2-7B/13B) for generating intelligent replies.

The advantage of this approach is the flexibility of making changes to the model and system, as well as the possibility of using the system in the company's internal infrastructure loop (to reduce security risks in case of adding internal information to the dataset in the future). At the time of this proposal, the most appropriate model for the stated purpose and constraints is LLaMA-2-13B, but other models, such as FinGPT, are also planned to be experimented with during the thesis process.

Implementation Steps:

* Embedding: Utilizing a pre-trained model to convert text inputs into vector representations.
* Vector Storage: Storing vector representations of reports in a database for efficient retrieval.
* Similarity-Based Retrieval: Using a similarity search algorithm to retrieve the most relevant documents.
* Retrieval-Augmented Generation (RAG): Enhancing the LLM with a retrieval mechanism to generate intelligent replies based on relevant documents.

During the experiments, it is possible to modify the above steps with new emerging tools

Significance:

This system, powered by advanced LLMs and RAG techniques, represents a pioneering step towards AI-enabled tutoring systems. It aims to democratize access to high-quality, customized educational support for reading and understanding financial reports. This research contributes to improving accessibility and understanding of financial reports through innovative techniques. The system's initial prototype marks a pioneering effort towards democratizing access to tailored educational support in financial analysis.

Kay.ai and Cybersyn offer an API to extract data from 10-K SEC reports using data via Snowflake Marketplace. The integration also utilizes the OpenAI API for embedding and querying the model. The difference in our system will be the use of an in-house generated dataset and the use of an open-source LLM.

Using a locally running open-source LLM instead of relying on an API for retrieval tasks to extract information from text can offer several advantages for users:

* Relying on an API for large-scale text processing can result in substantial costs, especially as usage increases. Running an open-source LLM locally eliminates the need to pay for API calls, making it a cost-effective solution for companies with high-volume text processing needs.
* Users can scale their local infrastructure based on their specific requirements without being constrained by API usage limits or facing unexpected costs associated with increased usage. This flexibility allows for better cost management and resource optimization.
* Open-source LLMs often allow for fine-tuning on custom datasets, enabling companies to tailor the model to their specific domain or industry. This customization can lead to improved performance on tasks relevant to the unique requirements.
* When you run an open-source LLM locally, you have complete control over your data. Running the model on-premises or within your secure environment ensures that data doesn't leave your infrastructure, reducing the risk of unauthorized access or data breaches.

Related literature:

(Ge 2023) leverages 10-K forms to measure regulatory barriers globally by fine-tuning BERT LLM. (Pasch and Ehnes 2022) fine-tuned transformer-based language model BERT, on different sources of company-related text data (News articles, blogs, and annual reports) for a classification task to predict the one-year stock price performance. The resulting StonkBERT, a transformer-based stock performance classifier, outperforms traditional models using BERT fine-tuned on various textual sources. Analysts use BERT for causal information classification or GPT-3.5 for creating Quant styled datasets. Summarization and RAG improve accessibility for individual investors. (Chun et al. 2023), (Gupta 2023), (Liu et al. 2023a).

Solutions like PoinT-5 and FETILDA address challenges in financial narrative summarization and document embedding for long financial texts. (Singh 2020) and (Xia et al. 2022).

Large pre-trained language models have been shown to store factual knowledge in their parameters, and achieve state-of-the-art results when fine-tuned on downstream NLP tasks. However, their ability to access and precisely manipulate knowledge is still limited, and hence on knowledge-intensive tasks, their performance lags behind task-specific architectures. Additionally, providing provenance for their decisions and updating their world knowledge remain open research problems.

Authors results highlight the benefits of combining parametric and non-parametric memory with generation for knowledge-intensive tasks—tasks that humans could not reasonably be expected to perform without access to an external knowledge source. (Lewis et al. 2020). Retrieval Augmented Generation (RAG) is a technique for enhancing the accuracy and reliability of LLM-generated responses by grounding the model on external sources of knowledge to supplement the LLM’s internal representation of information. RAG allows LLMs to access and incorporate relevant facts from an external knowledge base, such as user uploaded files, into their responses, instead of relying solely on their pre-trained parameters or hallucinating incorrect or misleading information. RAG has several benefits for improving the quality and trustworthiness of LLM-generated responses. First, it ensures that the model has access to the most current, reliable, and domain-specific facts, which can improve the accuracy and relevance of the responses. Second, it provides users with the sources of the model’s responses, which can increase the transparency and verifiability of the model’s claims. Third, it reduces the need for fine-tuning the model on new data and updating its para

(Asai et al. 2023) introduced a new framework called Self-Reflective Retrieval-Augmented Generation (SELF-RAG) that enhances an LM’s quality and factuality through retrieval and self-reflection Framework trains a single arbitrary LM that adaptively retrieves passages on-demand, and generates and reflects on retrieved passages and its own generations using special tokens, called reflection tokens. (Shao et al. 2023) showed that a new method ITER-RETGEN synergizes retrieval and generation iteratively, improving relevance modeling and outperforming existing methods on question answering tasks.

(Wu et al. 2023) have presented BloombergGPT, a financial domain LLM, outperforms existing models on in-domain tasks. The dataset remains closed, hindering the development of FinLLMs. (Yang et al. 2023) presented a FinGPT, an open-source LLM for finance, emphasizes a data-centric approach, promoting accessibility, transparency, and collaborative innovation in open finance.

(Loukas et al. 2021) released EDGAR-CORPUS dataset and edgar-crawler provide a financial NLP corpus, excluding tables. The author's work aims to retain tabular data in reports. (Wang et al. 2023) paper presented an Instruction Tuning paradigm explores task-specific tuning of LLMs in finance, highlighting diverse capabilities and future strategies for improvement. (Zhang et al. 2023) constructed TableInstruct dataset and TableLlama model address the challenge of instructing and evaluating LLMs for tables, enhancing context length with LongLoRA.

Practical relevance:

The application of LLMs in finance holds immense promise, addressing tasks like numerical claim detection, financial sentiment analysis, and reasoning. Although general-purpose LLMs show competitive zero-shot performance, domain-specific models with million-scale parameters outperform them. Notable models like BloombergGPT demonstrate remarkable performance across various financial tasks. However, it is crucial to underscore the substantial drawbacks associated with such proprietary models. (Zhao et al. 2023).

BloombergGPT is not open-sourced, limiting transparency. There remains a persistent risk of the model 'hallucinating' or generating inaccurate results. Moreover, these proprietary models, including BloombergGPT, come with a significant financial burden, being notably expensive to develop, train and maintain. This cost factor raises practical challenges and limits accessibility for broader usage in various financial applications.

(Liu et al. 2023b) presented an analysis of the training costs for FinGPT, BloombergGPT, and other LLMs. As of July 11, 2023, based on the number of GPU hours consumed and prices for AWS GPU usage the estimated training costs of LLMs approximately are: • FinGPT: $0,0002 million; • GPT-NeoX: $0.72 million; • BloomberGPT: $2.67 million; • BLOOM: $3.97 million; • LLaMA: $4.23 million.

Given the high expenses associated with training models, there's a practical incentive to explore pre-existing, freely available pre-trained LLMs, such as Llama 2 or/ and FinGPT.

The overarching goal of the project is to develop a system without any expenditure. Demonstrating success with existing pre-trained LLMs and the RAG method not only aligns with budget constraints but also provides a proof of concept for future scaling.

Unlike traditional models, the RAG system offers a distinct edge by eliminating the need for frequent model retraining when new SEC filings become available. This characteristic not only saves valuable time but also alleviates the resource-intensive process of retraining large language models, ensuring that the system remains up-to-date and adaptive without incurring the associated costs.

Evaluation

Evaluating LLMs poses challenges due to the evolving nature of benchmarks. While new benchmarks attempt to standardize evaluations, there's a disconnect between evaluation metrics and practical use cases, especially in domain-specific tasks. Evaluation metrics like Precision, Recall, F1 Score, Mean Average Precision, Precision at K, and AUC-ROC are commonly used but may not fully align with real-world applications.

We can measure aforementioned metric to evaluate our LLM and RAG system on the part of the MS MARCO (regular) and/or MS MARCO (TREC) datasets. (Liang et al.) covered problem setting , datasets and selection process for retrieval tasks of LLMs.

Experimental Limitations:

For the experimental phase of this thesis, our intention is to utilize the complimentary Colab VMs provided by Google. In the free version of Colab, access to high-cost resources like GPUs is significantly restricted. Colab offers these resources without charge, partly by imposing dynamic usage limits that may experience fluctuations and by not ensuring access to unlimited resources. Consequently, the overall usage limits, idle timeout periods, maximum VM lifetime, available GPU types, and other relevant factors may vary over time. Colab refrains from publishing these limits, partly due to their potential rapid changes. The free Colab VMs from Google come with limitations concerning RAM and VRAM, with a maximum 12.7 GB RAM capacity. A portion of this RAM (approximately 0.7 to 1.4 GB) is already allocated for general VM purposes. The execution of code takes place within a virtual machine exclusive to your account. These virtual machines are deleted after a period of inactivity and are subject to a maximum lifetime enforced by the Colab service.

Publication bibliography

Asai, Akari; Wu, Zeqiu; Wang, Yizhong; Sil, Avirup; Hajishirzi, Hannaneh (2023): Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. Available online at http://arxiv.org/pdf/2310.11511v1.

Chun, Ye Eun; Kwon, Sunjae; Sohn, Kyunghwan; Sung, Nakwon; Lee, Junyoup; Seo, Byungki et al. (2023): CR-COPEC: Causal Rationale of Corporate Performance Changes to Learn from Financial Reports. Available online at http://arxiv.org/pdf/2310.16095v1.

Ge, Haosen (2023): Measuring Regulatory Barriers Using Annual Reports of Firms. Available online at http://arxiv.org/pdf/2301.11084v1.

Gupta, Udit (2023): GPT-InvestAR: Enhancing Stock Investment Strategies through Annual Report Analysis with Large Language Models. Available online at http://arxiv.org/pdf/2309.03079v1.

Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman et al. (2020): Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. Available online at http://arxiv.org/pdf/2005.11401v4.

Liang, Percy; Bommasani, Rishi; Lee, Tony; Tsipras, Dimitris; Soylu, Dilara; Yasunaga, Michihiro et al.: Holistic Evaluation of Language Models. In *Published in Transactions on Machine Learning Research (TMLR)*. Available online at http://arxiv.org/pdf/2211.09110v2.

Liu, Shuaiqi; Cao, Jiannong; Yang, Ruosong; Wen, Zhiyuan (2023a): Long Text and Multi-Table Summarization: Dataset and Method. Available online at http://arxiv.org/pdf/2302.03815v1.

Liu, Xiao-Yang; Wang, Guoxuan; Yang, Hongyang; Zha, Daochen (2023b): FinGPT: Democratizing Internet-scale Data for Financial Large Language Models. Available online at http://arxiv.org/pdf/2307.10485v2.

Loukas, Lefteris; Fergadiotis, Manos; Androutsopoulos, Ion; Malakasiotis, Prodromos (2021): EDGAR-CORPUS: Billions of Tokens Make The World Go Round, pp. 13–18. DOI: 10.18653/v1/2021.econlp-1.2.

Pasch, Stefan; Ehnes, Daniel (2022): StonkBERT: Can Language Models Predict Medium-Run Stock Price Movements? Available online at http://arxiv.org/pdf/2202.02268v1.

Shao, Zhihong; Gong, Yeyun; Shen, Yelong; Huang, Minlie; Duan, Nan; Chen, Weizhu (2023): Enhancing Retrieval-Augmented Large Language Models with Iterative Retrieval-Generation Synergy. Available online at http://arxiv.org/pdf/2305.15294v2.

Singh, Abhishek (2020): PoinT-5: Pointer Network and T-5 based Financial NarrativeSummarisation. Available online at http://arxiv.org/pdf/2010.04191v2.

Wang, Neng; Yang, Hongyang; Wang, Christina Dan (2023): FinGPT: Instruction Tuning Benchmark for Open-Source Large Language Models in Financial Datasets. Available online at http://arxiv.org/pdf/2310.04793v2.

Wu, Shijie; Irsoy, Ozan; Lu, Steven; Dabravolski, Vadim; Dredze, Mark; Gehrmann, Sebastian et al. (2023): BloombergGPT: A Large Language Model for Finance. Available online at http://arxiv.org/pdf/2303.17564v2.

Xia, Bolun "Namir"; Rawte, Vipula D.; Zaki, Mohammed J.; Gupta, Aparna (2022): FETILDA: An Effective Framework For Fin-tuned Embeddings For Long Financial Text Documents. Available online at http://arxiv.org/pdf/2206.06952v1.

Yang, Hongyang; Liu, Xiao-Yang; Wang, Christina Dan (2023): FinGPT: Open-Source Financial Large Language Models. Available online at http://arxiv.org/pdf/2306.06031v1.

Zhang, Tianshu; Yue, Xiang; Li, Yifei; Sun, Huan (2023): TableLlama: Towards Open Large Generalist Models for Tables. Available online at http://arxiv.org/pdf/2311.09206v1.

Zhao, Wayne Xin; Zhou, Kun; Li, Junyi; Tang, Tianyi; Wang, Xiaolei; Hou, Yupeng et al. (2023): A Survey of Large Language Models. Available online at http://arxiv.org/pdf/2303.18223v13.