Knowledge-augmented Financial Market Analysis and Report Generation

Yuemin Chen[1, 2, 3*], Feifan Wu[1, 2, 3*], Jingwei Wang[2], Hao Qian[2], Ziqi Liu[2], Zhiqiang Zhang[2], Jun Zhou[2], Meng Wang[1†]
[1]College of Design and Innovation, Tongji University, China
[2]Ant Group, China
[3]School of Computer Science and Engineering, Southeast University, China
mengwangtj@tongji.edu.cn
{qianhao.qh, ziqiliu}@antgroup.com

Abstract

Crafting a convincing financial market analysis report necessitates a wealth of market information and the expertise of financial analysts, posing a highly challenging task. While large language models (LLMs) have enabled the automated generation of financial market analysis text, they still face issues such as hallucinations, errors in financial knowledge, and insufficient capability to reason about complex financial problems, which limits the quality of the generation. To tackle these shortcomings, we propose a novel task and a retrieval-augmented framework grounded in a financial knowledge graph (FKG). The proposed framework is compatible with commonly used instruction-tuning methods. Experiments demonstrate that our framework, coupled with a small-scale language model fine-tuned with instructions, can significantly enhance the logical consistency and quality of the generated analysis texts, outperforming both large-scale language models and other retrieval-augmented baselines.

1 Introduction

Crafting a compelling market analysis report is a complex process that demands careful selection of indicators, extensive financial knowledge, and perceptive reasoning. This intellectually challenging task requires sophisticated analysis and is often time-consuming. Automated generation techniques are urgently needed to streamline this process and reduce manual effort in financial market analysis.

In this paper, we introduce a novel task: Financial Market Analysis Generation (FMAG). The goal of FMAG is to automate the creation of analytical reports using financial market data. The primary challenge lies in synthesizing financial knowledge from extensive market information to produce logically consistent and high-quality analyses.

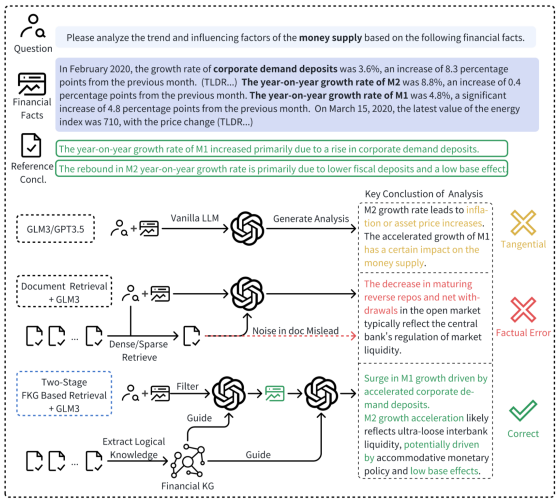

Figure 1: A comparison of FMAG between our method and other baselines.

While large language models (LLMs) have demonstrated remarkable abilities in natural language understanding and generation (Touvron et al., 2023; Wei et al., 2024; Zhu et al., 2023a), they still have limitations in FMAG. These include hallucinations (Asai et al., 2023; Wei et al., 2024), errors in financial knowledge (Kang and Liu, 2024), and insufficient capability to reason about complex financial problems (Reddy et al., 2024), all of which compromise the quality and reliability of generated text, see Fig. 1.

In this work, we propose a two-stage retrieval-augmented generation framework grounded in a financial knowledge graph (FKG), coined *Two-stage FKG-based Retrieval* (**TFR**). The proposed framework consists of three key parts. First, we use LLMs to construct a comprehensive FKG that delineates intricate relationships among financial entities, providing a solid foundation for knowledge retrieval. Second, we devise a clustering-based triple extraction algorithm designed to efficiently retrieve knowledge aligned with given queries and facts from the constructed FKG. Third, we intro-

[*]These authors contributed equally to this work.
[†]Corresponding Author.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1207–1217
November 12-16, 2024

duce a novel two-stage approach for knowledge retrieval and augmentation. In the first stage, the FKG serves as guidance for initial information selection. In the second stage, it facilitates reasoning based on the selected information. In addition, we developed a fine-tuning strategy for smaller models to ensure compatibility with our TFR framework and enable its integration with various LLMs.

Our experiments demonstrate that the proposed framework, even when coupled with a small-scale language model fine-tuned with instructions, can significantly enhance the logical consistency and quality of generated analysis texts, outperforming both large-scale language models and other retrieval-augmented baselines. In summary, our key contributions are as follows:

1. We introduce a novel task, FAMG, which requires reasoning with knowledge based on a substantial amount of input information to generate financial market analysis.

2. We propose a RAG framework grounded in a FKG. The framework consists of a KG construction method using LLMs, a cluster-based method to facilitate the retrieval process, and a two-stage retrieval method.

3. Experiments demonstrate that our framework significantly enhances the logical consistency and quality of the generated analysis texts, outperforming both large-scale language models and other retrieval-augmented baselines.

2 Task Description

In this paper, we introduce a novel task, FMAG, which aims to generate analytical text by reasoning from financial market information, including the values and changes of financial indicators and government financial policy. We consider the task in the format of Question Answering (QA) with an explanation. Specifically, the task can be defined as answering questions through analytical reasoning based on financial facts. We denote an instance of FMAG with three elements:$\{Q, F, A\}$, where Q denotes the user question, F represents financial facts, and A refers to the analysis text, including the analysis process and conclusions. Given Q and F, the progress of FMAG can be formulated as estimating the probability of generating reasoning steps and then deriving the conclusion $P(A|Q, F)$.

Table 1: **The dataset statistics for different splits.** #Avg. Facts means the average number of facts within the instance. #Avg. length means the average length of reference text within the instance.

Split	Instances	# Avg. Facts	# Avg. length
Train	2188	199	105
Test	295	173	97

2.1 Construction of FMAG dataset

To simulate real-world FMKG, we developed a benchmark focused on bond market analysis. This focus streamlines research while representing the complexity of various financial markets. The construction process of the dataset is as follows:

Collection of Expert-Written Analyst Reports We collected 6,000 analysis reports on the Chinese financial market, which included sections analyzing the bond market. Then we segmented the Chinese reports into chunks with a chunk size of 1,400 tokens to facilitate filtering and selecting.

Selection of Analyst Segment The process is done by prompting GPT-4 with examples. First, we extract segments with complete semantic meaning, which refers to text segments containing both factual premises and corresponding conclusions, from each chunk. Second, we select segments that conduct reasoning based on financial facts and are relevant to the bond market. From the selected segments, we extract facts from the analysis text and denote the facts as F_r.

Formulation of Task Instance The target of the formulation process is to get question Q and financial facts F to formulate a task instance. We first prompt GPT-4 to extract the conclusion from the selected analysis text, then prompt GPT-4 to generate questions based on the conclusions to get Q. Finally, we select data related to the bond market of the same date as the report date of analysis text. The data is selected from the financial indicator database AKshare (King, 2019) and CSMAR as supplement facts, which are denoted as F_s. The extracted facts F_r and supplement facts F_s are combined as financial facts F. The summary statistics of the dataset are presented in Table 1.

3 Proposed Framework

We introduce a two-stage FKG-based retrieval-augmented framework shown in Fig. 2. First, we build a FKG via prompting LLMs. Second, we

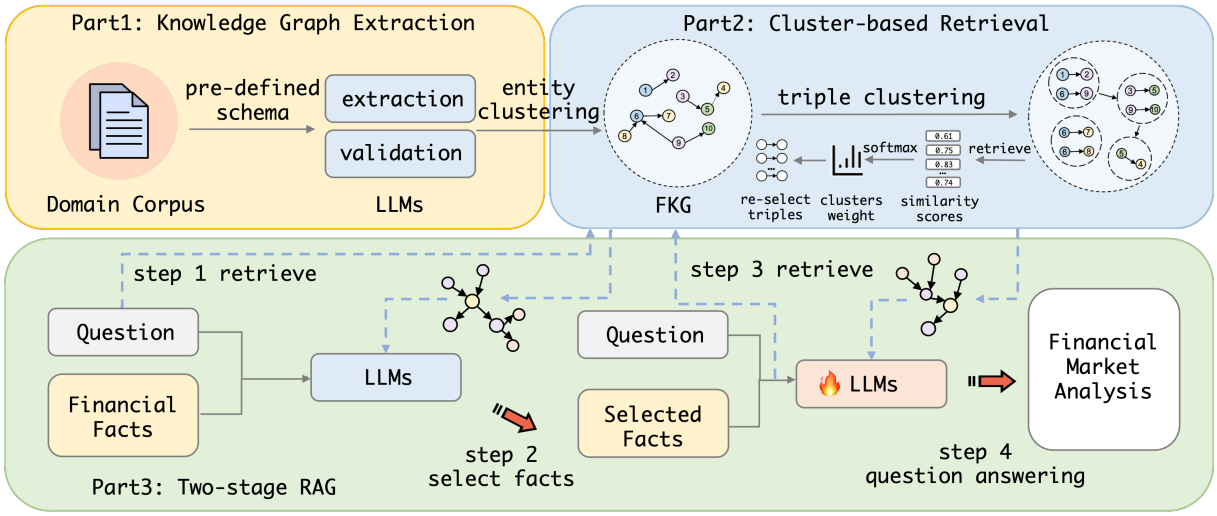

Figure 2: **The overall framework for our Two-stage FKG-based Retrieval (TFR)**. Initially, Knowledge Graph Extraction derives a FKG from the corpus using LLM prompt engineering. Next, in Cluster-based Retrieval, a retrieval method utilizing clustering facilitates the retrieval of triples from the extracted FKG. Lastly, the Two-stage RAG framework employs the FKG for initial information selection and subsequent reasoning to derive financial market analysis.

propose a clusters-based retrieval method to facilitate the retrieval of triples. Thirdly, we propose a two-stage RAG method, in which the KG serves as guidance to conduct initial information selection in the first stage and reasoning in the second stage.

3.1 LLM-based KG extraction

The forms of knowledge entailed in financial analysis texts include sequential, causal, and hypernym-hyponym relations between entities. Therefore, the first step is to extract the logical knowledge from the corpus and construct an FKG.

We decompose the process of knowledge graph construction into three phases, including schema definition, triple extraction, and triple verification. In the first phase, we examine prevalent standards to define a schema, which is the set and definitions of entity and relation categories in the financial domain. In the second phase, we design prompts with examples for each category in the schema. Through few-shot prompting, LLMs identify and extract triplets that meet the target schema from input texts. In the third phase, we prompt LLM to check the validation of extracted triples and modify the error-extracted ones with the original text as a reference. The detail of the extracted knowledge graph can be found in appendix B.

3.2 KG Retrieval from weighted clusters

Due to the free-form expression nature of expert-written financial reports in text corpus, the entities automatically extracted from text tend to be sparsely distributed. A common method is to deploy a clustering algorithm for node and edge clustering. While it is feasible in most general domains. Due to the complexity of financial terminology, some entities may have similar semantics but convey different meanings in practice. Using simple clustering methods may group together these similar yet distinct entities, leading to misleading results. Building on the aforementioned, instead of deploying clustering to connect entities, we introduced a retrieval strategy, enabling efficient retrieval from FKG automatically extracted.

Clustering of Triples We first perform clustering of nodes based on the similarity between their embeddings. To be specific, we utilize the `bge-base-zh` model (Xiao et al., 2024) to encode nodes in FKG. Then we apply the agglomerative clustering algorithm(Müllner, 2011) on the cosine similarity with a distance threshold to group similar nodes. Triples are categorized into clusters where head nodes share one entity cluster and tail nodes share another. Node-based clustering mitigates the impact of relational semantics, yielding entity-centric groupings of triples.

Cluster-based Retrieval The extracted financial knowledge graph contains redundant triples. To address this issue, we propose a retrieval method that considers cluster similarity to ensure diversity and relevance in the retrieved results. By weighting dif-

Table 2: **The results for different models on our benchmark.** GLM4-Score Concl. denotes the consistency score of the generated text and reference conclusion. GLM4-Score Text denotes the consistency score of generated text and reference text. TFR denotes our Two-Stage FKG based retrieval method. The highest score is denoted in **bold**, and the second-highest score is underlined.

Metric	GLM4-Score		BERT Score			RougeL		
Model	Concl.	Text	P	R	F1	P	R	F1
GPT3.5-turbo	2.8625	2.4502	0.6309	0.7341	0.677	0.4672	0.4244	0.3952
GLM3-turbo	2.8247	2.5464	0.6265	0.7351	0.675	0.4057	0.4709	0.3891
GLM3-turbo + BM25 Retrieve	2.9661	2.539	0.6232	0.732	0.6719	0.3322	0.4716	0.3377
GLM3-turbo + Dense Retrieve	3.0761	2.737	0.6336	0.7515	0.6862	0.3678	0.5281	0.382
GLM3-turbo + Triples Retrieve	3.2136	2.9492	0.6371	0.7332	0.6803	0.4333	0.4742	0.4094
GLM3-turbo + TFR	**3.3254**	2.9966	0.6328	0.7267	0.6751	0.3441	0.4728	0.3504
GLM3-6b	2.7424	2.3932	0.6579	0.7331	0.6907	0.3048	0.5162	0.3127
GLM3-6b (SFT w/o FKG)	2.9424	3.4373	**0.8546**	**0.7878**	**0.8178**	0.6184	0.707	0.5911
GLM3-6b (SFT with FKG)	3.0949	3.4712	0.8536	0.7629	0.8034	0.6788	0.5775	0.5708
GLM3-6b (SFT with FKG) + TFR	3.2203	**3.5593**	0.8393	0.7728	0.8023	**0.7438**	**0.6384**	**0.6474**

ferent clusters, this approach maintains relevance while avoiding the concentration of results in a single cluster. We employ the `bge-base-zh` model as our encoder for both query q and KG triples. We aim to retrieve k triples for each query q. The process is as follows: We initially retrieve the top-n triples ($n \gg k$). Given that each triple belongs to a distinct cluster, we calculate the average similarity score for each cluster based on the similarity scores of its constituent triples. We then apply a softmax function to normalize these scores, deriving retrieval weights for different clusters. The weighted score of a particular cluster is multiplied by k to determine the number of triples to be retrieved from that cluster. To ensure we retrieve k triples from different clusters, we round the calculated number of triples for each cluster to the nearest integer and then make minor adjustments.

3.3 Two-stage RAG framework

We divide the reasoning process into two stages: financial facts selection and question answering.

Stage1: Financial Facts Selection Given the inherent complexity and volatility of financial data, it is crucial to navigate through the noise (irrelevant or misleading information) to focus on pertinent facts. The initial and critical step in financial analysis is to carefully identify and select information that is directly relevant to the question at hand. To facilitate this process, we leverage the Domain Knowledge Graph extracted in section 3.1 that encapsulates expert knowledge. First, we retrieve knowledge based on question Q. Then both question Q and retrieved knowledge K_1 are combined as the input for LLM, which is then prompted for financial facts selection. This process can be formalized as:

$$\begin{aligned} K_1 &= \text{Retriever}(Q), \\ F_{\text{select}} &= \text{LLM}(Q, K_1, F, \text{prompt}). \end{aligned} \tag{1}$$

Stage2: Question Answering In the second stage, we first retrieve knowledge based on question Q and selected facts F_{select} from the first stage. We then feed the question Q, selected facts F_{select}, and retrieved knowledge K_2 to LLMs. In this process, LLMs serve as a reasoner to conduct reasoning based on input context. Considering the potential noise introduced by retrieved knowledge, we use the prompting method to prune and eliminate irrelevant retrieved knowledge. The whole process can be denoted as:

$$\begin{aligned} K_2 &= \text{Retriever}(Q, F_{\text{select}}), \\ A &= \text{LLM}(Q, F_{\text{select}}, K_2, \text{prompt}). \end{aligned} \tag{2}$$

3.4 Supervision Fine-tuning with KG retrieval

We can also fine-tune a language model using instruction-following demonstrations to align question-answering based on retrieved knowledge. Adopting the self-instruct approach (Wang et al., 2023), we concatenate financial facts, retrieved triple knowledge, and questions as a prompt, training the model to generate financial analysis text.

Our subsequent ablation experiments revealed that incorporating retrieved KG triples into X not only enhances the model's utilization of the retrieved knowledge but also improves the language model's inherent performance. This improvement was notable compared to training data that solely included factual information.

4 Experimental Setup

4.1 Dataset and Metrics

We use the dataset constructed in section 2.1 to train and evaluate the model. For evaluation, we employ three metrics, including GLM-4-Score, BERTScore (Zhang et al., 2019), and ROUGE-L (Lin, 2004), to access the performance of the models. GLM-4 (GLM et al., 2024) was used to assess the consistency of opinions between the generated text and both the reference conclusion and reference text, scoring each from 0 to 5. A higher GLM-4 score signifies greater consistency between the generation and the reference. BERTScore and RougeL measure semantic similarity between the generated and reference text. We placed greater emphasis on the GLM-4 consistency score with the reference conclusion, as it indicates whether the generated text arrived at correct conclusions based on factual analysis.

4.2 Baselines

Our proposed approach is evaluated against three categories of methods: vanilla LLMs, retrieval-based models, and training-based models.

Vanilla LLMs: We compare our method with various baseline, including vanilla GPT-3.5-turbo, ChatGLM3-turbo (GLM et al., 2024), and ChatGLM3-6b (GLM et al., 2024).

Retrieval-based Models: We consider three retrieval-augmented baselines: BM25 Retriever (Roberts et al., 2020), and Dense Retriever (Lewis et al., 2020a) for document-level retrieval, and Dense Retriever for knowledge triple retrieval. ChatGLM3-turbo serves as the backbone for these retrieval-based methods.

Training-based Models: We also fine-tune ChatGLM3-6b with the constructed training set as a baseline. Detailed descriptions of these baselines are provided in the appendix C.

5 Experimental Results

5.1 Main Results

Table 2 presents comprehensive benchmark results. GLM3-6b (SFT with FKG) + TFR demonstrates superior performance across multiple metrics, achieving the highest scores in GLM4-Score with the reference text and RougeL while maintaining competitive performance in other metrics. This synergistic approach underscores the efficacy of combining supervised fine-tuning with our novel retrieval-augmented generation method. Notably, the RAG-only method (GLM3-turbo + TFR) achieves the highest GLM4-Score with the reference conclusion, indicating its particular strength in improving answer accuracy.

Zero-shot performance of vanilla LLMs yields comparatively lower GLM4-Scores relative to other methods, which can be attributed to their inherent lack of domain-specific knowledge. The performance disparity between zero-shot and fine-tuned models is substantial, with GLM3-6b (SFT w/o triples) outperforming its zero-shot counterpart across all metrics. Notably, the improvements in GLM4-Score with reference text (43.6%), BERT Score F1 (18.4%), and RougeL F1 (89.0%) are significantly larger than the increase in GLM4-Score with reference conclusion (7.3%). This discrepancy suggests that while SFT enhances overall model performance, its impact is more pronounced in aligning the generated text's linguistic style with the reference text rather than improving the model's ability to infer conclusions accurately. This phenomenon underscores the challenge of enhancing a model's reasoning capabilities in this task through fine-tuning alone.

Among retrieval methods, triple retrieval exhibits the most significant improvement in GLM4-Scores compared to its backbone GLM3-turbo and other documents level retrieval models, showing knowledge graph as a more efficient source for retrieval augment in this scenario compared to non-structural documents. Interestingly, while retrieval methods generally enhance GLM4-Scores, particularly for conclusions, they often lead to decreased BERTScore and RougeL metrics, suggesting that RAG alone can improve the model's ability to reason correct conclusions but struggles to align the linguistic style with expert-written texts. To better demonstrate the effectiveness of our method compared to other baselines, we provide a case study in appendix E.

5.2 Ablation Study

We conduct ablation experiments to evaluate the effectiveness of each module in our proposed approach. These experiments involve the systematic variation of key components, including the TFR method and the SFT module, as well as the inclusion or exclusion of retrieved triple knowledge during SFT training. The results, as presented in Fig. 3, reveal several insights.

The addition of our TFR method consistently

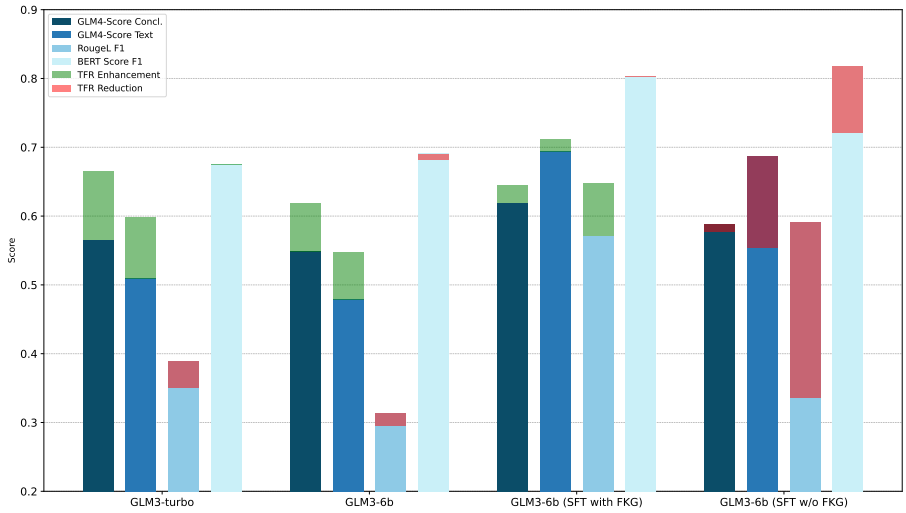

Figure 3: **Comparative analysis of model performance on our benchmark with different components.** Green segments indicate performance improvements achieved through the TFR method, while red segments represent performance decreases relative to the original model.

Table 3: **Main results on our benchmark of different KG size.**

	GLM4-Score		BERT Score	RougeL
KG Size	Concl.	Text	F1	F1
0%	2.9932	2.5768	0.6710	0.3872
20%	3.2508	2.7831	0.6708	0.3280
40%	3.2915	2.8000	0.6734	0.3288
60%	3.3220	2.7864	0.6726	0.3386
80%	3.2847	2.7864	0.6762	0.3512
100%	3.2682	2.8380	0.6792	0.3504

improves the GLM4-Score across most model variants, and the performance boost is more pronounced for more capable models. Fine-tuning significantly enhances model performance, particularly in BERTScore and RougeL metrics. For SFT, incorporating triples not only enhances the ability of smaller LLMs (6B parameters) to utilize retrieved information but also significantly improves the model's inherent capabilities, particularly in terms of conclusion accuracy. Notably, when applying our TFR method, the SFT model trained with triple knowledge exhibits further performance improvements, demonstrating excellent knowledge integration capabilities. In contrast, models without triple integration during SFT show a decline in performance across various metrics when the RAG method is applied.

5.3 Effect of Knowledge Graph Size

This section examines the impact of knowledge graph completeness on our method's performance. We measure completeness by varying the graph size. Size reduction is achieved by randomly removing triples. The experiment utilizes our RAG-only method with GLM3-turbo to isolate the effect of graph size. The results are shown in Table 3. While increasing graph size generally improves performance, the relationship is not strictly linear due to noise in the knowledge graph. Excessive information can introduce more noise, potentially degrading performance.

6 Conclusion

This research introduces a novel retrieval-augmented framework, leveraging a financial knowledge graph to address the limitations of LLMs in generating high-quality financial market analysis reports. The proposed framework, combined with a small-scale language model fine-tuned with instructions, performed significantly better than large-scale language models and other retrieval-augmented baselines. The results demonstrate the potential of our method to enhance the logical consistency and quality of generated financial market analysis, thereby contributing to the automation of premium market analyses.

7 Limitations

In this paper, we propose an efficient framework aimed at enhancing the logical consistency and quality of generated analyses. However, our study does have several limitations. Firstly, our method is built upon the RAG framework, which means its

performance is highly dependent on the quality of the constructed KG. Although our KG is extracted from the corpus through prompt engineering with LLM, it likely contains some noise. To address this issue, we have implemented several strategies to mitigate potential impacts. Specifically, we employed self-validation techniques within LLM to reduce errors in the extraction results. Additionally, we introduced a cluster-based method to minimize redundancy in retrieved triples and utilized prompting techniques to guide LLMs in selecting relevant knowledge before generating answers. Another potential improvement involves applying training methods to facilitate automatic extraction. This work primarily focuses on retrieval-augmented approaches for enhancing LLMs with KGs, leaving room for further advancements in the field.

8 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (U23B2057, 62176185, 62276063), the Natural Science Foundation of Jiangsu Province (BK20221457), and the Ant Group.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *ArXiv*, abs/1908.10063.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46.

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 78–106.

Yanjun Gao, Ruizhe Li, John R. Caskey, Dmitriy Dligach, Timothy A. Miller, Matthew M. Churpek, and Majid Afshar. 2023. Leveraging a medical knowledge graph into large language models for diagnosis prediction. *ArXiv*, abs/2308.14321.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Shibo Hao, Bowen Tan, Kaiwen Tang, Bin Ni, Xiyan Shao, Hengzhe Zhang, Eric Xing, and Zhiting Hu. 2023. BertNet: Harvesting knowledge graphs with arbitrary relations from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5000–5015.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *ArXiv*, abs/2311.11944.

Haoqiang Kang and Xiao-Yang Liu. 2024. Deficiency of large language models in finance: An empirical examination of hallucination. In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Albert King. 2019. Akshare. `https://github.com/akfamily/akshare`.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, volume 33, pages 9459–9474.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 9459–9474.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *ArXiv*, abs/1109.2378.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Lidén, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *ArXiv*, abs/2302.12813.

Maciej P Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15(1):1569.

Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, and Chris Tanner. 2024. Docfinqa: A long-context financial reasoning dataset. *arXiv preprint arXiv:2401.06915*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384.

Guijin Son, Hanearl Jung, Moonjeong Hahm, Keonju Na, and Sol Jin. 2023. Beyond classification: Financial reasoning in state-of-the-art language models. In *Joint Workshop of the 5th Financial Technology and Natural Language Processing (FinNLP) and 2nd Multimodal AI For Financial Forecasting (Muffin) in conjunction with IJCAI 2023*, page 34.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Yilin Wen, Zifeng Wang, and Jimeng Sun. 2023. Mindmap: Knowledge graph prompting sparks graph of thoughts in large language models. *ArXiv*, abs/2308.09729.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *ArXiv*, abs/2303.17564.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2024. Pixiu: a large language model, instruction data and evaluation benchmark for finance. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 33469–33484.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023a. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *arXiv preprint arXiv:2305.13168*.

Yuqi Zhu, Xiaohan Wang, Jing Chen, Shuofei Qiao, Yixin Ou, Yunzhi Yao, Shumin Deng, Huajun Chen, and Ningyu Zhang. 2023b. Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *ArXiv*, abs/2305.13168.

A Related Work

A.1 Financial Natural Language Processing

The utilization of language models in financial NLP is a thriving research area. Some general domain language models have been applied to financial domain models like FinBERT (2019), PIXIU (2024) and BloombergGPT (2023). While increasingly augmenting LLM with domain corpus, existing benchmarks are confined to token or sequence classification tasks. For more challenging tasks that require a multi-step reasoning process, the field of financial reasoning is still largely unexplored. Son et al.(2023) introduced a dataset called sFIOG consisting of synthetic investment thesis samples to evaluate the financial reasoning capabilities of LLMs and proposed a prompting method for the controlled generation of context. Islam et al.(2023) proposed FINANCEBENCH (Islam et al., 2023), an open book financial question answering benchmark required multi-step reasoning. However, the utilization of domain knowledge in financial reasoning tasks is still unexplored.

A.2 Retrieval-augmented LLMs

Retrieval-augmented generation methods(RAG) retrieve relevant information from an external database for the query and incorporate the retrieved knowledge with context as input for generation(Lewis et al., 2020b; Karpukhin et al., 2020; Shi et al., 2024). RAG is efficient for incorporating external knowledge and reducing hallucination in knowledge-intensive tasks (Peng et al., 2023; Baek et al., 2023). Wen et al.(2023)deployed knowledge graph retrieval with exploration methods to prompt LLMs for graph reasoning in the medical domain. Gao, Yanjun et al.(2023) inferred diagnoses from the knowledge graph, the retrieved results are then used to prompt LLMs for final diagnoses. While these methods have reduced hallucination and boosted performance in domain tasks, the retrieval is based on a pre-defined knowledge graph. Retrieval based on domain knowledge graphs with noise information is less explored. In addition, the application of knowledge retrieval in the financial domain is largely unexplored.

A.3 LLM-augmented KG construction

Recent advances in Large Language Models (LLMs) have markedly enhanced Knowledge Graph (KG) construction. Incorporating LLMs into KG development has simplified the automated construction of KGs. Two primary approaches have emerged: direct knowledge extraction from LLMs and leveraging LLMs' natural language understanding for Information extraction. Gao et al.(2023) propose a method to harvest extensive KGs from pre-trained Language Models using minimal input, while West et al.(2022) apply knowledge distillation to extract symbolic KGs from GPT-3. LLMs have also demonstrated proficiency in structured output tasks, including domain-specific entity, relation, and event extraction, with few or no training examples. Recent research by Zhu et al. (2023b) utilizes multiple LLM agents in iterative dialogues for automated KG construction. Other approaches, such as ChatIE(Wei et al., 2023) and ChatExtract(Polak and Morgan, 2024), reframe information extraction as question-answering tasks using ChatGPT and prompt engineering.

B Extracted Domain Knowledge Graph

B.1 Definition of Schema

The entity types in KG include financial indicators, change of financial indicators, comparison between indicators, composition of two indicators, comparison of financial indicators and threshold, market status, and macro-economic control policy. The relationships in KG include *'belong_to'*, *'affect'*, *'indicate'*, *'equal_to'*, *'may_lead_to'*.

B.2 Details of KG

In total, our constructed KG contains 8052 nodes and 5664 triples.

C Implementation of Baselines

BM25 Retriever for document-level retrieval We choose BM25 document retriever methods as a baseline to retrieve the top k documents for each question query. For a fair comparison, we use the analysis texts in the training dataset as the documents, which are the source of extracted KG.

Dense Retriever for document-level retrieval We use a dense retrieval method that is based on text embedding as a baseline. We use `bge-base-zh` model to encode query and documents. The document source is the same as the BM25 method.

Dense Retriever for knowledge triples retrieval To show the efficiency of our proposed retrieval framework compared to direct retrieval on the knowledge graph, we choose the dense retrieval

method for knowledge triple retrieval as a baseline. We use `bge-base-zh` as an encoder to directly encode the query and KG triples and used the similarity scores between them to retrieve the top k most relevant triples as our baseline. We attempted existing KG retrieval methods, which aim to find the shortest KG path between every pair of question entities(Wen et al., 2023; Sun et al., 2024). We tried to apply these methods to our benchmark, but they were almost unable to retrieve any paths on the KG we constructed. Upon analysis, we believe the reason might be that our automatically constructed financial knowledge graph contains numerous redundant nodes with identical meanings but different expressions due to the use of phrasal representations, which reduces the connectivity of the graph. Therefore, we abandoned these methods as baselines.

LLM fine-tuned with training set We choose to fine-tune directly the extracted 2188 training data as a baseline. Specifically, we use the following format of instructional data for fine-tuning: $\{I, F, Y\}$, where I represents the financial question, F represents the financial facts, and Y represents the financial analysis text.

D Implementation Details

For both the baseline method using fine-tuning and supervision fine-tuning with KG retrieval in our proposed method, we employed the LoRA(Hu et al., 2022) method. The training data format for the latter is $\{I, F, T, Y\}$, where T represents the retrieved triples. Each experiment is conducted on one A100, and the same parameters are set for both fine-tuning experiments. Specifically, we set batch size as 1, number of training epochs as 3, LORA rank as 8, learning rate scheduler as cosine, and learning rate as 1e-3.

E Case Study

The case study for our task is shown in Fig. 4. Due to the lengthy nature of the generated contents and the constraints of pages, we only present the results of the top three methods with the best overall performance in the benchmark. Compared to the reference text crafted by experts, the text generated using our proposed RAG framework effectively captures the key points per the reference while incorporating pertinent and accurate information. Utilizing our RAG framework with SFT, the generated text covers relevant conclusions with a language style and length most akin to the example text. Conversely, text generated via the direct fine-tuning method merely reiterates facts, lacking any reasoning or conclusion.

F Ethical Considerations

We propose a framework to reduce hallucination in financial analysis generation by enhancing LLMs with KG. Experiment results show that it is efficient in increasing the logical consistency and quality of generation. However, generated contents face a higher standard of faithfulness in the financial scenarios. Therefore, when applying our research to real-world applications, examination of the faithfulness of generated content is essential.

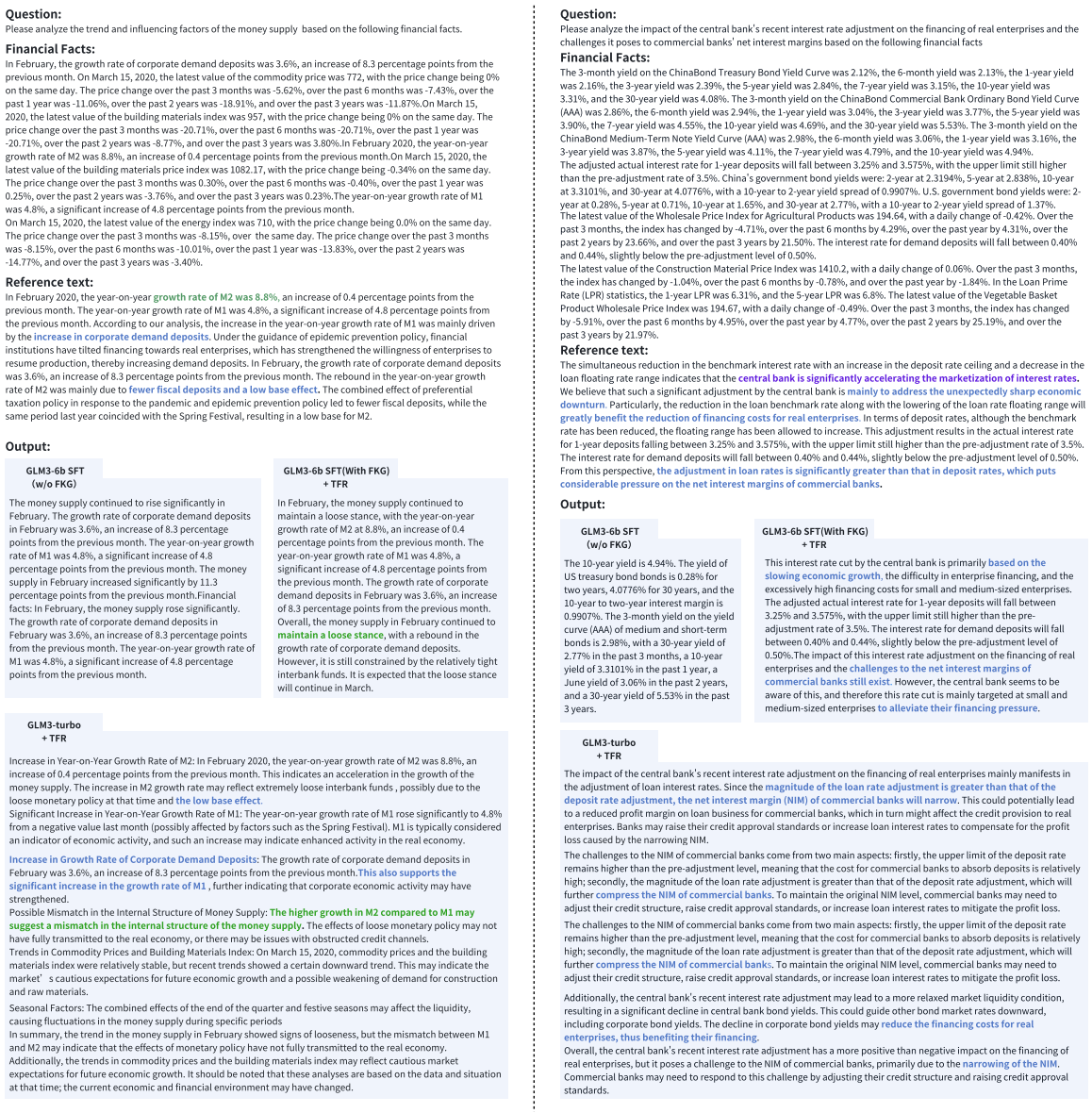

Figure 4: **The case study on the proposed benchmark.** The figure displays our methods' results and the SFT baseline applied to given cases. We translate Chinese reports into English for better understanding. Segments highlighted in bold purple indicate conclusions of the reference text that are not included in the generation. In the generated text, segments highlighted in bold blue indicate conclusions that align with the blue portions in the reference text, while the segments highlighted in green represent inferences drawn from the green portions in the reference text.