FinNLP-2025

The 10th Workshop on Financial Technology and Natural Language Processing

Proceedings of the Workshop

November 9, 2025

Suzhou, China

Preface

Welcome to FinNLP, a forum dedicated to fostering international collaboration and knowledge sharing in applying NLP to the dynamic domain of FinTech. We are especially delighted that the 10th edition of FinNLP coincides with the launch of the ACL Special Interest Group on Economic and Financial Natural Language Processing (SIG-FinTech). With SIG-FinTech, we aim to establish a sustainable presence for this research area within the ACL community. The mission of SIG-FinTech goes beyond advancing NLP research in finance and economics. We also seek to bring together experts from diverse fields to propose important research directions and exchange insights. Each year, we invite speakers from various disciplines to share their perspectives. This year, in addition to distinguished industry experts—Dr. Shi-Xiong (Austin) Zhang and Dr. Sambit Sahu from Capital One—we are honored to host Prof. Saeed Abdullah (Penn State), an expert in Human–Computer Interaction (HCI), and Prof. Andrea Rocci (Università della Svizzera italiana), a linguist, who will share their perspectives on financial information and applications from different disciplinary angles. Another highlight of FinNLP-2025 is the FinEval initiative, where we discuss methods for evaluating generated reports. A central challenge remains how to assess text quality in terms of its usefulness for user decision-making—a task that is still difficult to standardize. We hope that, starting with the 10th FinNLP, this conversation will pave the way toward more robust evaluation frameworks in the era of generative AI.

We are indebted to all the program committee members who dedicated significant time and expertise to provide insightful feedback on submissions and guide the selection process for FinNLP-2025. This year, we received 31 submissions to the main track and accepted 17. The accepted papers cover a broad spectrum of topics, including financial language modeling, evaluation frameworks, document auditing, sentiment analysis, fraud detection, and relation extraction. They also highlight advances in retrieval-augmented generation (RAG), multimodal analysis of charts and tables, and responsible AI practices in finance and law. We look forward to the stimulating discussions at this year's FinNLP and hope you enjoy the workshop.

Chung-Chi Chen, Genta Indra Winata, Stephen Rawls, Anirban Das, Hsin-Hsi Chen, Hiroya Takamura
FinNLP-2025 Organizers

Organizations

General Chairs

- Chung-Chi Chen - AIST, Japan
- Genta Indra Winata - Capital One, US
- Stephen Rawls - Capital One, US
- Anirban Das - Capital One, US
- Hsin-Hsi Chen - National Taiwan University, Taiwan
- Hiroya Takamura - AIST, Japan

Organizing Committee (Advisor)

- Sophia Ananiadou - University of Manchester, UK
- Hen-Hsen Huang - Academia Sinica, Taiwan
- Shi-Xiong Zhang - Capital One, US
- Supriyo Chakraborty - Capital One, US
- Sambit Sahu - Capital One, US

Program Committee (Reviewer)

- Mrinal Ahlawat - Google
- Paulo Alves - Católica Porto Business School
- Yung-Chun Chang - Taipei Medical University
- Emmanuele Chersoni - Aix-Marseille University
- Nelson Correa - Andinum, Inc.
- Giulia D'Agostino - Università della Svizzera Italiana
- Akshar Prabhu Desai - Indian Institute of Technology
- Pablo Duboue - Textualization Software Ltd.
- Himaanshu Gauba - Google
- Yu-Shiang Huang - National Taiwan University
- Kiyoshi Izumi - The University of Tokyo
- Karthik Kushala
- Chit-Kwan Lin - Revelata, Inc.
- Mohammad Luqman - Google
- Chenyang Lyu - MBZUAI
- Hilal Pataci - University of Texas at San Antonio
- Vijai Anand Ramar
- Hiroki Sakaji - Hokkaido University

- Takehiro Takayanagi - The University of Tokyo
- Ke Tian - Rakuten Group, Inc
- Kanji Uchino - FUJITSU RESEARCH OF AMERICA, INC.
- Chuan-Ju Wang - Academia Sinica
- Shih-Hung Wu - Chaoyang University of Technology
- Ajay Yadav - Google LLC
- Sin-Han Yang - Appier
- Annie T.T. Ying - IBM

Shared Task Organizers

- Takehiro Takayanagi - The University of Tokyo, Japan
- Tomas Goldsack - University of Sheffield, UK
- Kiyoshi Izumi - The University of Tokyo, Japan
- Chenghua Lin - University of Manchester, UK
- Hiroya Takamura - AIST, Japan
- Chung-Chi Chen - AIST, Japan

Table of Contents

LLM as a Guide: an Approach for Unsupervised Economic Relation Discovery in Administrative Documents

Thomas Sebbag[1,2], **Solen Quiniou**[1], **Emmanuel Morin**[1],

[1] Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France,
[2] Explore, Carquefou, France,
Correspondence: thomas.sebbag@univ-nantes.fr, solen.quiniou@univ-nantes.fr, emmanuel.morin@univ-nantes.fr

Abstract

Effective relation extraction (RE) from unstructured text is critical, especially when the target relations are unknown. In this case, we can leverage Large Language Models (LLMs) to perform this task. In this paper, we present an LLM as a guide approach for identifying economic relations. This novel methodology, based on sentence clustering and an LLM, is used to identify previously unknown economic relations in French administrative documents. It addresses the challenge of extracting actionable knowledge without predefined relation labels. We evaluate our approach on French and English RE datasets, demonstrating high precision and recall in detecting previously unknown relations. Our results suggest that clustering and LLM-based methods can effectively discover and categorize economic relations, with potential applications to private corpora.

1 Introduction

Effective information extraction poses a significant challenge when the target information is undefined. Identifying the relevance of the information to be extracted is a critical step, particularly in the case of analyzing market dynamics and organizational decisions in the economic domain. As a subfield of the economic domain, the administrative domain is interested in the textual productions of various public organizations in order to analyze the behavior of economic actors from the administration's perspective.

On the one hand, Open Relation Extraction (ORE) aims to transform unstructured text into structured and actionable knowledge by using an unsupervised mechanism to extract predefined relations as presented by Shukla et al. (2025). According to Jiang et al. (2024), traditional RE and ORE methods mainly focus on predefined patterns that refer to a predefined set of relations and entities given during the training step. ORE primarily rely on two datasets from traditional RE: TACRED (Zhang et al., 2017) and Fewrel (Han et al., 2018). These datasets were created from English data from news wires and web text, and the relations present in these datasets are clearly defined. However, to the best of our knowledge, few studies address the problem of identifying unknown relations, and even fewer propose a methodology for exploring a corpus specifically dedicated to this task. Indeed, extracting relations is even more challenging when the target information to be extracted is undefined.

On the other hand, Semantic Typing (ST) attempts to assign tokens or relevant text spans to semantic categories such as relation types, entity types, or event types within a given context (Huang et al., 2022). This task may correspond to the first stage of the RE process. We aim to apply this approach to the administrative domain by constructing a relational schema of the interactions between public administrations and their environment, based on their textual production, to support economic analysis.

In this work, we propose a methodology inspired by Wrzalik et al. (2024) to identify unknown economic relations, in French documents produced by a public administration. Our contributions can be summarized as follows:

- An approach based on sentence clustering and an LLM guiding experts in the process of identifying unknown relations;
- An evaluation of our methodology on two RE datasets (one in French and one in English) to assess its ability to capture general relations.
- The data and associated relations from our administrative corpus[1].

[1] Github repository

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

2 Related Work

2.1 Open Relation Extraction

Information Extraction (IE) extracts structured relationships in the format < *arg1; rel; arg2* > using semantic or syntactic cues to classify sequences (Niklaus et al., 2018). ORE extends this definition to identify new relations from unlabeled open-domain corpora (Wang et al., 2024). While these methods provide a basis, they struggle to capture global context.

Clustering-based approaches (Wang et al., 2022; Li et al., 2022; Zhou et al., 2023) are another method for addressing RE by grouping relations into types through Masked Language Modeling or feature extraction. However, these are sensitive to data quality, dataset size, and relation distribution, making them reliant on the training data, especially for sharp topics.

Generative Relation Extraction (GRE) is an LLM-based approach that comprehends input text and identifies relations in a zero-shot setting, without relying on predefined patterns (Jiang et al., 2024). LLMs in zero-shot and few-shot setups have been shown to perform GRE without fine-tuning (Wadhwa et al., 2023; Li et al., 2023) and have been used to extract economic relations (Ettaleb et al., 2025). However, GRE methods still rely on a predefined set of relations and entities, somewhat similar to traditional RE. Wrzalik et al. (2024) uses a methodology based on statement retrieval and LLM to extract text passages about companies emmisions goals. Inspired by this approach, we propose an approach to discovering unknown relations in a zero-shot setup without predefined relation definition.

2.2 Semantic Typing

Although relation classification aims at categorizing defined relations, it remains a close task to semantic typing, a task focusing on identifying unknown relations according to Thomas et al. (2024). Recent works rely on the typing technique to discover unknown relations, using entity types as a preliminary stage to relation identification on TACRED and French press documents (Lyu and Chen, 2021; Mallart et al., 2021).

3 Datasets

To identify unknown relations between economic actors in a given area and conduct an economic analysis, we use French administrative documents. Potin et al. (2023) and Sebbag et al. (2025) described the textual production of public administrations as a great playground to harvest economic information. They also described these documents as challenging for NLP tasks due to their lack of structure and the potential amount of noise. In this section, we present the three datasets considered.

3.1 Administrative Data

In public administrations, decision documents summarizing actions and debates by authorities are particularly valuable. Discussions with domain experts confirmed that those documents contain key economic relations, offering insights for market understanding.

Following expert recommendations, we selected the French city of Lambesc to initiate our relation typing study. To ensure consistency, we focused on decisions from 2024, a year with an unchanged administrative team. The resulting corpus includes 830 unique sentences and 2,460 named entities (*ORG*, *LOC*, *PER*, in that order; see Appendix A.1). We used internal tools to scrape publicly available documents from Lambesc's website, extracting only sentences to simplify this initial exploratory analysis.

Example of extracted sentence translated to English : *The Town Hall of Lambesc informs the assembly that the commune has applied to the SAFER for the acquisition of the plot of land cadastral section AT n°84 located in Bonrecueil Nord.*

3.2 Other Relation Extraction Datasets

To evaluate our methodology, we also chose two annotated datasets: one close to our domain, containing French economic relations (BizRel[2]), and one containing general English relations (FewRel[3]).

Bizrel (Khaldi et al., 2022) is a multilingual dataset focusing on Business Relation extraction between organizations. It contains 2 007 sentences in French and six relations including 5 economic relations (details in Appendix A.2). Only the French part of the dataset was used in our experiment. To the best of our knowledge, Bizrel is the open source dataset that most closely matches our use case in the context of extracting economic relations.

FewRel (Han et al., 2018), is an English Relation Extraction dataset consisting of 100 general

[2]https://github.com/Geotrend-research/business-relation-dataset

[3]https://github.com/thunlp/FewRel

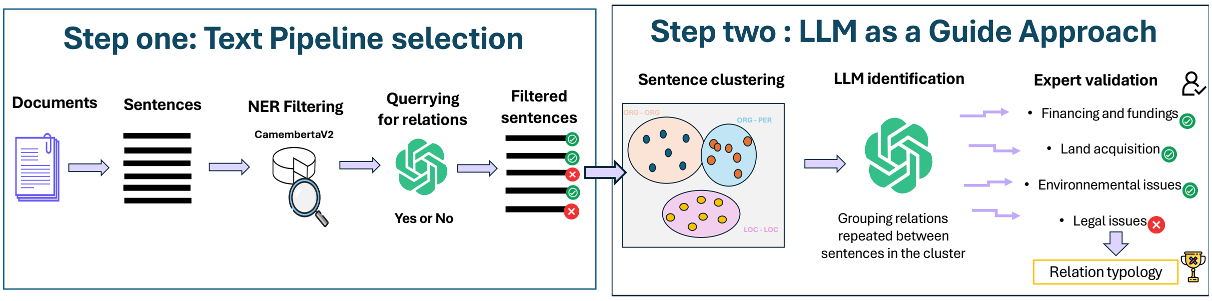


Figure 1: Our two-step pipeline for relation identification, with text selection and LLM-based relation generation.

domain relations from Wikipedia data. We created a random snapshot of 1 611 sentences, from the validation set. This snapshot contains 16 relations described in Appendix A.3. The entity types in this dataset match ours, making it an appropriate resource for evaluating our setup on English data.

4 Relation Exploration Pipeline

In this section, we describe the two steps of our relation identification pipeline, on administrative documents, as shown in Figure 1.

4.1 Step One: Text Pipeline Selection

To identify unknown relations, we go through a phase of selecting sentences that might contain a relation.

The first sub-step consists of segmenting the texts into sentences, based on the hypothesis that sentence-level relations may be more explicit, making automatic relation identification easier. For this text segmentation, we use Langchain[4] module *tiktoken*, as well as Spacy Sentencizer[5] with the *fr_core_news_sm* model.

In a second sub-step, we detect named entities using a CamemBERTaV2 language model (Antoun et al., 2024), fine-tuned on Adminset-NER (Sebbag et al., 2025), an annotated NER dataset consisting exclusively of French administrative documents. This allows us to filter sentences to retain only those that contain at least two named entities

The last sub-step is to use an LLM for a binary task of detecting whether or no a relation is present within the sentence. GPT-4o (OpenAI et al., 2024) was chosen for this task because of its ability to highlight potential relations, as suggested by Ding et al. (2024).

[4]https://python.langchain.com
[5]https://spacy.io/api/sentencizer

4.2 Step Two: LLM-as-a-Guide Approach

Our method proposes an approach based on using an LLM as a guide to identify unknown relations in sentences, after regrouping them into clusters.

For the first sub-step, we created clusters using entity pair templates which could contain a relation.For example, a single cluster will contain all sentences that have both an ORG entity and a LOC entity. We also evaluate this clustering method to a random selection of sentences.

In the next sub-step, we prompted the LLM with a temperature of 0.001 to identify the relation types that are repeated in the sentences of each given cluster, providing the associated texts and entities. (See Appendices A.4 and A.5 for the prompts in French and English).

In the final step, three experts collaboratively evaluated the relations proposed by the LLMs. They serve as Product Owners or Product Managers at a company specialized in information delivery for public procurement applications. Reaching consensus on whether each relation was valid, invalid, or overlapping with a previously identified one was required for validation.

5 Experiments and Results

In this section, we present the experimental setup and the results on the different datasets considered.

Table 1 and 2 present the results, in terms of precision (P) recall (R or R') and f1-score (F1) on relation types.

5.1 Relation Identification on Administrative Data

For our first experiment, we applied entity pair clustering to administrative data (Admin Entity), generating five clusters based on predefined templates. We compared this to random sentence selection (Admin Random), which formed four clusters while maintaining a balanced ratio of repeated

Models	Admin Random		Admin Entity	
	P	R’	P	R’
GPT-4o	88,88%	57.14%	85,71%	85.71%
GPT-4.1	**91,67%**	**78.57%**	**87,50%**	**100.00%**
Llama 3.1 70B	80,00%	28.57%	55,56%	71.43%
Mixtral 8*22B	66,67%	28.57%	60,00%	64.29%
Llama 3.1 8B	40.00%	28.57%	36,84%	50,00%
Mixtral 7B	42.86%	21,43%	54,55%	42,86%

Table 1: Precision and an alternative version of Recall results for relation identification on administrative data, on two methods used to create sentence clusters.

relations. Since relations between entities are not known in advance, precision (P) is calculated based on those judged relevant by domain experts. To address the lack of annotated ground truth, we introduce an alternative recall metric (R’), based on the maximum number of relations, 14 in total, generated by GPT-4.1 and validated by domain experts.

Admin Random As shown in Table 1, GPT models exhibits strong zero-shot performance, with GPT-4.1 achieving 91.66% precision but a recall of 78,57% identifying 11 relations. Only one false positive was noted, where a legal reminder was misclassified as a relation. Llama 3.1 70B and Mixtral 8x22B each produced only four relations, mostly generic, less than half of GPT-4.1’s output, with two false positives for Mixtral 8x22B. Llama 3.1 8B also produced four relations, but created seven false positives, most of which corresponded to inaccurate labels regarding the provided examples. Mistral 7B obtained the lowest results with only generic labels. Only three of the labels were identified correctly, while four were identified as false positives, which explains the low recall.

Admin Entity In this setup, GPT models keep their advance on relation identification. While precision exhibits a slight decrease with a score of 87.45%, recall presents a perfect score, generating the most true relations on topics like *project financing*, *land transfer/acquisition*, and *city planning* (see Appendix A.7), with only 2 identified as false positives. Llama 3.1 70B and Mixtral 8x22B kept struggling in this setup, generating general relations with a high proportion of false positives; however, the models improved the quality of the labels generated, even though some of them were too generic. The smaller models improved their average performance in terms of recall, but Llama 3.1 8B over generated twelve false labels while Mistral 7B generated a significant number of duplicate relations.According to domain experts, the relations generated in this setup were generally finer than before and encapsulated more precise semantic concepts by using entity pairs to create clusters.

In conclusion, experts most appreciated GPT’s ability to identify a wide range of relations and generate clear, meaningful labels, which reflect in our version of the recall score.

5.2 Relation Identification on Annotated Datasets

Datasets	Models	P	R	F1
Bizrel	GPT-4o	83,33%	83,33%	**83,33%**
	GPT-4.1	83,33%	83,33%	**83,33%**
	Llama 3.1 70B	66,67%	66,67%	66,67%
	Mixtral 8*22B	50,00%	66,67%	57,14%
	Llama 3.1 8B	100,00%	66,67%	80,00%
	Mistral 7B	60,00%	50,00%	54,55%
Fewrel	GPT-4o	82,35%	87,50%	**84,85%**
	GPT-4.1	81,25%	81,25%	81.25%
	Llama 3.1 70B	57,14%	25,00%	34,78%
	Mixtral 8*22B	64,29%	52,94%	58,06%
	Llama 3.1 8B	26,32%	31,25%	28,57%
	Mistral 7B	50,00%	18,75%	27,27%

Table 2: Results for relation identification on annotated datasets.

As shown in Table 2, experiments on BizRel and FewRel used random sentence selection to form four clusters. Entity pair clustering was not applicable, as BizRel contains only *ORG* entities and FewRel does not provide entity labels in its validation set. Since both datasets contain annotated relations, we evaluated our protocol using standard metrics precision (P), recall (R), and F1-score (F1).

Results on BizRel This dataset includes six relation types, one of which is *Other*. GPT models achieved an F1 score of 83.33%, correctly identifying all five economic relations. However, they introduced a non-existent type, *Ranking comparison*, and failed to capture the *Other* category. In contrast, Llama 3.1 70B and Mixtral 8x22B struggled to identify accurate relations, often misinterpreting the context. Interestingly, the smaller models outperformed their larger counterparts in terms of precision during this experiment. Llama 3.1 8B benefited from this advantage, achieving a higher F1 score by producing only one false positive. Conversely, Mistral 7B underperformed in terms of F1 score, generating three positive relations and two false positives.

Results on FewRel The FewRel validation snapshot includes 16 relation types. GPT models again achieved the best performance, with GPT-4o reaching an F1 score of 84.85%, correctly identifying 13 out of 16 relations. However, GPT's models consistently struggled with certain types, such as *mother*, *main subject*, and *voice type*. Llama 3.1 70B predicted only seven relations, four of which were correct, resulting in a low recall. Mixtral 8x22B demonstrated a stronger ability to identify relevant relations but frequently produced incorrect labels, reflecting difficulties in capturing contextual nuances. Notably, its outputs included unusually accurate label generation, which could indicate potential data contamination given the model's overall performance across our experiments. However, this remains a hypothesis requiring further investigation. Of the smaller models, Llama 3.1 8B generated five correct relations but produced fourteen false positives, suggesting that it was overwhelmed by the number of relations to identify. Mistral 7B obtained the lowest F1 score, producing only three true positives and duplicating labels across clusters.

GPT models demonstrated strong overall performance across both evaluation datasets but also exhibited consistent weaknesses in recognizing specific relation types. These results suggest potential biases toward certain subjects or difficulty detecting relations that differ greatly from the training data.

6 Conclusion

In this work, we propose an approach based on sentence clustering and an LLM guiding experts in the process of identifying unknown relations in the French administrative domain. We also evaluated our methodology on two annotated datasets for relation extraction in French and English. The validation by industry experts highlights its potential for economic analysis. Furthermore, this method could be applied to private company corpora, as they share similar unstructured frameworks and domain-specific terms. Our experiments mark a first step towards extracting complex relations from administrative documents, with plans to extend it to complex paragraphs and implicit information. Ultimately, we aim to create an annotated corpus for relation extraction, and we hope that our methods will inspire future work in relation identification and information retrieval.

Limitations

It seems important to us to discuss some of the limitations of our experiments:

- Given that FewRel is an open domain dataset obtained from the web, we are concerned about potential data contamination during LLMs training process, which could affect its performance in our experiment. Although we did not evaluate the model directly on the RE task, but rather on its ability to generate relation labels, it is still possible to use its preexisting knowledge base, including this data, to produce more accurate results.
- To effectively evaluate LLMs on our administrative data, we chose to calculate recall based on the maximum number of correct relations generated by GPT-4.1, given that the data are not yet annotated at this stage of the project. This approach introduces a bias in favor of GPT models, which can hinder independent verification. Three domain experts evaluated the relations, and we consider their expertise sufficient to assess the relevance of each relation and base an evaluation score on it.
- Our approach relies on proprietary models, raising concerns about dependency on private companies. Furthermore, these models only partially reveal their inner workings, which restricts our ability to analyze their outputs.

Ethical Concerns

All datasets used in this study, BizRel and FewRel, are publicly available. The French administrative data from the Lambesc website are available under a Creative Commons license under the Attribution-NonCommercial 4.0 International License. This transparency minimizes ethical concerns regarding data acquisition and usage.

In addition, the interpretability and transparency of LLM's decision-making processes are essential. Recognizing the limitations and biases of LLMs, including occasional information inaccuracies, we emphasize the importance of reliability in our evaluation methodology. Furthermore, the integration of LLMs-as-a-guide impacts traditional human roles, requiring careful consideration of the ethical implications of labor displacement. Moreover, the powerful capabilities of LLMs underscore the need for responsible use and measures to prevent misuse, aligning our research with high ethical standards and societal well-being. We carefully reviewed and ensured that the data we used to input each LLM did not contain any offensive information.

The total cost of our experiments is estimated to be 48.86$, according to OpenAI's billing history regarding the usage of GPT models. The total environmental cost, according to the Jean Zay supercomputer documentation is equivalent to 58.872 Wh or 3.07 kg CO2eq based on the carbon intensity of the energy grid mention by BLOOM environmental cost study also made on Jean Zay (Luccioni et al., 2022).

References

Wissam Antoun, Francis Kulumba, Rian Touchent, Éric de la Clergerie, Benoît Sagot, and Djamé Seddah. 2024. Camembert 2.0: A smarter french language model aged to perfection. *Preprint*, arXiv:2411.08868.

Linyi Ding, Jinfeng Xiao, Sizhe Zhou, Chaoqi Yang, and Jiawei Han. 2024. Topic-Oriented Open Relation Extraction with A Priori Seed Generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13834–13845, Miami, Florida, USA. Association for Computational Linguistics.

Mohamed Ettaleb, Véronique Moriceau, Mouna Kamel, and Nathalie Aussenac-Gilles. 2025. The contribution of LLMs to relation extraction in the economic field. In *Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal) - Proceedings of the workshops*, pages 175–183, Abu Dhabi, United Arab Emirates. Association of COmputational Linguistics. Backup Publisher: Chung-Chi Chen and Antonio Moreno-Sandoval and Qianqian Xie and Jimin Huang and Sophia Ananiadou and Hsin-Hsi Chen.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4803–4809, Brussels, Belgium. Association for Computational Linguistics.

James Y. Huang, Bangzheng Li, Jiashu Xu, and Muhao Chen. 2022. Unified Semantic Typing with Meaningful Label Inference. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654, Seattle, United States. Association for Computational Linguistics.

Pengcheng Jiang, Jiacheng Lin, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2024. GenRES: Rethinking evaluation for generative relation extraction in the era of large 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 2820–2837, Mexico City, Mexico. Association for Computational Linguistics.

Hadjer Khaldi, Farah Benamara, Grégoire Siegel, Camille Pradel, and Nathalie Aussenac-Gilles. 2022. How's Business Going Worldwide ? A Multilingual Annotated Corpus for Business Relation Extraction. In *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 3696–3705, Marseille, France. European Language Resources Association (ELRA).

Guozheng Li, Peng Wang, and Wenjun Ke. 2023. Revisiting large language models as zero-shot relation extractors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6877–6892, Singapore. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2022. Open relation and event type discovery with type abstraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6864–6877, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2022. Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. *arXiv preprint*. ArXiv:2211.02001 [cs].

Shengfei Lyu and Huanhuan Chen. 2021. Relation Classification with Entity Type Restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395, Online. Association for Computational Linguistics.

Cyrielle Mallart, Michel Le Nouy, Guillaume Gravier, and Pascale Sébillot. 2021. Active learning for interactive relation extraction in a French newspaper's articles. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 886–894, Held Online. INCOMA Ltd.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and

262 others. 2024. GPT-4 Technical Report. *arXiv preprint*. ArXiv:2303.08774 [cs].

Lucas Potin, Vincent Labatut, Pierre-Henri Morand, and Christine Largeron. 2023. FOPPA: an open database of French public procurement award notices from 2010–2020. *Scientific Data*, 10(1):303. Publisher: Nature Publishing Group.

Thomas Sebbag, Solen Quiniou, Nicolas Stucky, and Emmanuel Morin. 2025. AdminSet and AdminBERT: a Dataset and a Pre-trained Language Model to Explore the Unstructured Maze of French Administrative Documents. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 392–406, Abu Dhabi, UAE. Association for Computational Linguistics.

Neelesh Kumar Shukla, Prabhat Prabhakar, Sakthivel Thangaraj, Sandeep Singh, Weiyi Sun, C Prasanna Venkatesan, and Viji Krishnamurthy. 2025. GraphRAG Analysis for Financial Narrative Summarization and A Framework for Optimizing Domain Adaptation. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 23–34, Abu Dhabi, UAE. Association for Computational Linguistics.

Hugo Thomas, Guillaume Gravier, and Pascale Sébillot. 2024. Recherche de relation à partir d'un seul exemple fondée sur un modèle N-way K-shot : une histoire de distracteurs. In *Actes de la 31ème Conférence sur le Traitement Automatique des Langues Naturelles, volume 1 : articles longs et prises de position*, pages 157–168, Toulouse, France. ATALA and AFPC.

Somin Wadhwa, Silvio Amir, and Byron Wallace. 2023. Revisiting relation extraction in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15566–15589, Toronto, Canada. Association for Computational Linguistics.

Jiaxin Wang, Lingling Zhang, Wee Sun Lee, Yujie Zhong, Liwei Kang, and Jun Liu. 2024. When phrases meet probabilities: Enabling open relation extraction with cooperating large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13130–13147, Bangkok, Thailand. Association for Computational Linguistics.

Jiaxin Wang, Lingling Zhang, Jun Liu, Xi Liang, Yujie Zhong, and Yaqiang Wu. 2022. MatchPrompt: Prompt-based open relation extraction with semantic consistency guided clustering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7875–7888, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marco Wrzalik, Florian Faust, Simon Sieber, and Adrian Ulges. 2024. NetZeroFacts: Two-Stage Emission Information Extraction from Company Reports. In *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing*, pages 70–84, Torino, Italia. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Sizhe Zhou, Suyu Ge, Jiaming Shen, and Jiawei Han. 2023. Corpus-Based Relation Extraction by Identifying and Refining Relation Patterns. In *Machine Learning and Knowledge Discovery in Databases: Research Track*, pages 20–38, Cham. Springer Nature Switzerland.

A Appendix

A.1 Detailed statistics on our administrative corpus

	Sentences	Entities		
		ORG	LOC	PER
Admin Data	830	1 325	738	397

Table 3: Statistics of the administrative dataset.

A.2 BizRel relations

Labels	Details
Investment	An organisation is a subsidiary of another organisation, or an organisation holds (all or part) of the shares of another organisation.
Competition	A competition/rivalry between two organisations providing the same goods or services, or wanting to access the same relatively small market.
Cooperation	A contractual cooperation between two organisations, or when two organisation work together on the same project.
Legal Proceedings	One organisation launches a legal proceedings against another organisation.
Sale-Purchase	One organisation is a client of another, or supplies it with goods or services.
Others	If none of the previously described relations are expressed between the tagged entity pair, or if other types of relations out of this list are expressed, the relation should be OTHERS.

Table 4: Details on the relations in the BizRel dataset.

A.3 FewRel relations

Labels	**Details**
Crosses	Obstacle (body of water, road, etc) which this bridge crosses over or this tunnel goes under
Original language of film or TV show	Language in which a film or a performance work was originally created. Or language of work or name.
Competition class	Official classification by a regulating body under which the subject (events, teams, participants, or equipment) qualifies for inclusion.
Part of	Object of which the subject is a part.
Sport	Sport in which the subject participates or belongs to.
Constellation	The area of the celestial sphere of which the subject is a part.
Position played on team/speciality	Position or specialism of a player on a team.
Located in or next to body of water	Body of water on or next to which a place is located.
Voice type	Person's voice type. expected values: soprano, mezzo-soprano, contralto, countertenor, tenor, baritone, bass (and derivatives).
Follows	Immediately prior item in a series of which the subject is a part.
Spouse	The subject has the object as their spouse (husband, wife, partner, etc.).
Military rank	Military rank achieved by a person, or military rank associated with a position.
Mother	Female parent of the subject.
Member of	Organization or club to which the subject belongs.
Child	Subject has object as biological, foster, and/or adoptive child.
Main subject	Primary topic of a work.

Table 5: Details on the relations in the FewRel dataset.

A.4 [French version] Prompt used for explore relations into administrative documents with LLMs

```
Tu es un assistant IA qui est chargé de faire de l'extraction de relations
dans des batch de textes écrient par des administrations publique française.

Tu reçois en entré plusieurs textes ayant été regroupé dans la même catégorie
ainsi que les entités nommées présentent dans ces textes entre crochet,
tu dois déterminer si ces textes sont reliés par des relations similaires
en indiquant celle-ci et en regroupant les textes par type de relations.
Peux-tu regrouper ces textes par famille de relation
en indiquant la nature de la relation pour chaque groupe ?

Merci de donner des exemples uniquement issus du fichier en txt.
```

A.5 [English version] Prompt used for explore relations into administrative documents with LLMs

```
You're an AI assistant tasked with extracting relationships from batches
of texts written by French public administrations.

You receive as input several texts that have been grouped together in the same category,
as well as the named entities present in these texts between brackets,
and you need to determine whether these texts are linked by similar relationships
by indicating the relationship and grouping the texts by relationship type.
Can you group these texts by relationship family,
indicating the nature of the relationship for each group?

Please give examples from the txt file only.
```

A.6 [French version] Prompt used for explore relations into BizRel with GPT-4o

```
Tu es un assistant IA qui est chargé de faire de l'extraction de relations
dans des batch de textes issus du web français.
Tu reçois en entré plusieurs textes ayant été regroupé dans la même catégorie
ainsi que les entités nommées présentent dans ces textes comprise entre
des balises [E11][E12] et [E21][E22], tu dois déterminer si ces textes
sont reliés par des relations similaire en indiquant celle-ci et en
regroupant les textes par type de relations.

Peux-tu regrouper ces textes par famille de relation en indiquant
la nature de la relation pour chaque groupe ?

Merci de donner des exemples uniquement issus du fichier en txt.
```

A.7 Relations generated from our administrative corpus

Labels	**Details**	**Entity pairs**
Land or infrastructure acquisition	A public administration acquire land or real estate from another organization or person.	ORG - ORG ORG - PER
Subsidies and financing	Concern the grant for a financing asked or proposed by the administration to another organization.	ORG - ORG
Relations between public institutions	Participation in an event, membership, or the desire to create a project with one or more other administrations. Inter-city sharing of administrative staff.	ORG - ORG
Delegation of public services	Public service delegation to an exterior organization.	ORG - ORG
Management of environmental labels	False positive, example didn't match the category.	None
Located in	Mention of a relation between two geographical entities	LOC - LOC
Loan guarantee	A public authority guarantees a loan for an organization to assist with its economic development.	ORG - ORG
Public procurement and contracts	Awarding a public contract or signing a contract with an organisation.	ORG - ORG
Management of municipal and administrative services	Administrative organization, new public services available to the community.	ORG - ORG
Part of an organisation	An elected official could be part of an organization implying one or multiple public structures.	PER - ORG
Public works and infrastructure	Refurbishment and renovation of public buildings. Work on roads and public spaces.	ORG - LOC
Administrative litigation	False positive relative to reminder of the law concerning administrative appeal procedures. Or legal framework for amending agreements and endorsements to municipal contracts.	None
Vote or political position on the community board	Participation in community board votes: Taking a position for or against it, or taking no position.	PER- PER
Position or mandate held within an organization	Elected officials functions or assignment of temporary responsibilities.	ORG - PER
Collaboration, Discussion, or Mention in an Administrative Context	Collaboration, discussion, or mention of individuals in the context of a meeting, debate, or administrative action.	PER - PER
Administrative Decision or Action	An elected official is mention for give a decision about an organization	PER - ORG

Table 6: Labels and details on the relations generated from administrative data, using sentence clustering by entity pair.

Zero-Shot Extraction of Stock Relationship Graphs with LLMs

Hao Zhou[1] **Luis Felipe Costa Sperb**[1,2] **Tiejun Ma**[1]
[1]School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom
[2]Kingston Business School, Kingston University, London, United Kingdom
{hao.zhou, v1lcost5, tiejun.ma}@ed.ac.uk
f.costasperb@kingston.ac.uk

Abstract

Stock return prediction using Graph Neural Networks (GNNs) is often hindered by flawed graph structures. Existing models typically rely on rigid, predefined static graphs based on industry classifications or knowledge bases, which fail to capture the nuanced and complex business relationships between companies. To address this limitation, we pioneer the use of Large Language Models (LLMs) for zero-shot extraction of stock relationship graphs. By prompting an LLM, we extract its prior knowledge to construct a multi-relational static graph that captures fundamental corporate relationships. This method eliminates the reliance on simplistic, predefined industry classifications or knowledge base. To our knowledge, this is the first work to leverage zero-shot LLM graph generation for financial modeling, providing a more meaningful structural backbone for GNN-based prediction tasks.

1 Introduction

Stock return prediction is a crucial technique for profitable stock investment, and recent studies have begun to incorporate stock relationships as additional information for forecasting. To explore such information, graph neural networks (GNNs), a powerful paradigm for modeling inter-stock dependencies, are being applied. However, the predictive power of GNNs is often constrained by inadequate graph construction strategies. Current approaches that use static graphs rely on predefined relationships (e.g., industry sectors) that cannot capture evolving business relationships. In reality, stocks are not independent and can be influenced by complex connections beyond simple sector groupings; for instance, competitive or supply-chain relationships create dependencies that predefined classifications miss. These rigid graphs fail to capture meaningful underlying relationships, limiting the GNN's potential.

To address this significant gap in graph construction, we introduce a novel method for creating a static relationship graph to serve as the market's "structural backbone". By prompting a Large Language Model (LLM), we extract fundamental business relationships that reflect stable, long-term interconnections between companies, including but not limited to sector connections, competitive relationships, and supply chain dependencies. Our approach moves beyond the traditional, structurally-defined graphs used in prior research.

To sum up, our core contribution is that we pioneer the use of LLMs for zero-shot extraction of stock relationship graphs that capture multifaceted business relationships and long-term structural interconnections between companies, eliminating reliance on predefined industry classifications. To the best of our knowledge, this is the first work to prompt LLMs for this purpose in the financial domain.

2 Related Work

Graph Neural Networks in Finance Patel et al. (2024) identified the common pattern and segregated this task into three different modules: Graph Construction Module, Historical Information Encoder and Relational Module. Early studies typically rely on predefined stock relationships, such as industry-sector (Sawhney et al., 2021), consumer-supplier (Chen and Robert, 2022), and shareholding patterns (Wang et al., 2023), etc. Some works also construct static correlation graphs based on historical stock price (Li et al., 2021; Yin et al., 2021), though they are more widely used in build dynamic graphs due to their fast-changing nature. For instance, Cheng and Li (2021) infer the latent stock relation from the sequential embedding at each timestep. Since the static graphs and dynamic graphs model the stock relationships from different views, researchers has began to explore the combination of them. For example, Wang et al. (2022) use a static graph which is predefined based on domain knowledge and a latent dynamic graph which is learned end-to-end. The output feature vectors from the two separate graph convolutions are summed together to create a single fused representation.

LLMs in Finance Recent researches begin to explore the potential of using information extracted by LLMs to construct and analyze knowledge graphs in the financial sector. Notably, Trajanoska et al. (2023) used LLMs to generate knowledge graphs from ESG (Environmental, Social, and Governance) reports, creating node-edge-node triples to represent relationships between entities, including companies. Similarly, Cheng et al. (2022) developed a Semantic-Entity Interaction Module with LLMs and CRF to construct financial knowledge graphs from brokerage reports, demonstrating the potential of zero-shot techniques for relationship extraction without manual rule-setting. However, these works all rely on additional external textual information, while we aim to extract the prior knowledge within the LLMs to build company relationship graphs.

3 Methodology

3.1 Framework Overview

In our framework, we first employ an LLM to perform zero-shot extraction of structured company relationship graphs without any textual input or external data sources, thereby allowing us to directly probe the LLM's prior knowledge about inter-company relationships. The extracted graph encodes multiple types of relations, including supply chain dependencies, competitive dynamics, and strategic partnerships, represented as multi-relational edges. The initial node features are constructed from historical stock price data. To model the structural and semantic information embedded in these graphs, we adopt two representative graph neural network (GNN) architectures: the Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) and the Relational Graph Attention Network (RGAT) (Busbridge et al., 2019). The learned node representations are subsequently used for stock ranking prediction. In summary, the framework enables us to evaluate the efficacy of the LLM-extracted relationship graphs by comparing their predictive performance against models trained on other predefined relationship graphs.

3.2 LLM-prompted Static Graph Construction Module

The foundation of our model is a static, multi-relational graph, G_S, designed to capture the multifaceted, long-term economic ties between companies. These fundamental relationships, such as supply chains and competitive positions, provide a structural backbone that is less susceptible to the daily noise of market news. Instead of relying on manually curated databases, which can be incomplete or outdated, we introduce a novel methodology to construct this graph by leveraging a Large Language Model (LLM) as a zero-shot knowledge extractor.

The first step is to systematically query the LLM to identify relationships between every pair of companies (s_i, s_j) in our stock universe S. To ensure the LLM provides structured and relevant output, we employ carefully designed prompt engineering.

Based on the S&P Global Business Relationship Dataset [1], we define a comprehensive **multi-relational taxonomy**, $\mathcal{R}$, that covers key economic interactions: is_Customer_of, is_Supplier_of, is_Distributor_of, is_Competitor_of, is_Peer_of, is_Investor_of, is_Invested_by, is_Subsidiary_of, is_Parent_of, is_Cross_owned_with, is_Joint-Venture_partner_of, is_R&D_partner_of, is_Marketing_partner_of, is_Strategic_partner_of, is_Licensor_of, is_Licensee_of, is_Franchisor_of, is_Franchisee_of, is_Creditor_of, is_Borrower_of, is_Acquirer _of, is_Target_of_acquisition, is_Merger_partner_with, has_Interlocking_directors_with.

For each pair of companies, we use a structured prompt that forces the LLM to classify their primary relationship into one of these predefined categories and to provide a confidence score. An example prompt is shown in Figure 1.

However, LLM-generated graphs can be sparse or noisy for certain relation types. To ensure that each relational graph used by our model possesses a meaningful level of connectivity, we apply a **connectivity-based pruning** step. Specifically, for each relation type r, we calculate the total number of edges in its initial graph, given by $\|A'_r\|_1 = \sum_{i,j} A'_r[i,j]$. If this edge count falls below a predefined connectivity threshold κ (set to

[1] https://www.marketplace.spglobal.com/en/datasets/business-relationships-(5)

```
You are a financial analyst with expertise
in the US market. For each of the following
relationships, please help me find five
companies from the list of SP500
constituents which have that relationship
with the source company and sort them from
high to low by relevance. It's fine if you
can't find enough related companies. Please
make sure the relationship is existing and
real. The companies are represented by
ticker symbol.

Your response should be in the json format
without explanation: {{{Source_company}:
{{Relation_1: [company_1,
company_2,...]}}}}.

Source company: {ticker}
Relationships: {Relation_1, Relation_2,...}
```

Figure 1: The structured prompt used to query the LLM for zero-shot relationship extraction between company pairs.

200 in our experiments), we deem the relation type too sparse to be reliable and discard it entirely.

This pruning step yields a final, refined set of adjacency matrices $\{A_r\}_{r\in\mathcal{R}_{final}}$, where $\mathcal{R}_{final} \subseteq \mathcal{R}$. Collectively, the set of nodes S and these filtered adjacency matrices constitute our static multi-relational graph, $G_S = (S, \{A_r\}_{r\in\mathcal{R}_{final}})$, which serves as a stable and robust input to the dual-component GNN encoder.

3.3 Graph Neural Networks

We employ two representative multi-relational graph neural networks to model the extracted company relationship graphs: the Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) and the Relational Graph Attention Network (RGAT) (Busbridge et al., 2019). RGCN extends standard graph convolutions to multi-relational settings by applying relation-specific transformations with parameter-efficient basis decomposition, enabling effective aggregation of neighborhood information across different relation types. RGAT, in contrast, incorporates relation-aware attention mechanisms that adaptively weight the influence of neighboring nodes, allowing the model to focus on more informative relations. Both architectures operate on initial node features derived from historical price data, and their learned representations are used for stock ranking prediction.

RGAT The Relational Graph Attention Network (RGAT) (Busbridge et al., 2019) generalizes the conventional graph attention mechanism (Veličković et al., 2018) to accommodate multi-relational graphs by introducing relation-specific transformations and attention computations. For each relation type r, a relation-specific linear transformation $\mathbf{W}^r$ is applied to the input node features, producing

$$\mathbf{h}_i^r = \mathbf{W}^r \mathbf{x}_i. \tag{1}$$

Multi-head attention coefficients are then computed as

$$e_{ij}^{(r,h)} = \text{LeakyReLU}\left(\mathbf{a}^{(r,h)\top}\left[\mathbf{h}_i^r \,\|\, \mathbf{h}_j^r\right]\right), \tag{2}$$

where $\mathbf{a}^{(r,h)}$ denotes a learnable attention vector associated with relation r and attention head h, and $\|$ represents vector concatenation. These coefficients are normalized via the softmax function:

$$\alpha_{ij}^{(r,h)} = \frac{\exp\left(e_{ij}^{(r,h)}\right)}{\sum_{k\in\mathcal{N}(i)} \exp\left(e_{ik}^{(r,h)}\right)}. \tag{3}$$

The relation-specific aggregated representation is then obtained as

$$\mathbf{h}_i'^{(r)} = \sum_{j\in\mathcal{N}(i)} \alpha_{ij}^{(r,h)} \mathbf{h}_j^r. \tag{4}$$

Finally, the outputs across all relations are combined through an aggregation function (e.g., sum, mean, max, or learned attention-based weighting) to yield the final node representation:

$$\mathbf{h}_i' = \text{AGG}\left(\mathbf{h}_i'^{(r)}\right)_{r=1}^{R}. \tag{5}$$

RGCN The Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2017) extends the standard graph convolution operation to multi-relational graphs by incorporating relation-specific transformations and a basis decomposition scheme for parameter efficiency. For each node i and relation type r, the model applies degree-based normalization with

$$c_{i,r} = \frac{1}{|\mathcal{N}_i^r|}, \tag{6}$$

where $\mathcal{N}_i^r$ denotes the set of neighbors of node i connected via relation r. The relation-specific transformation matrices $\mathbf{W}^r$ are parameterized using a basis decomposition:

$$\mathbf{W}^r = \sum_{b=1}^{B} a_{rb} \mathbf{V}_b, \tag{7}$$

where $\{\mathbf{V}_b\}_{b=1}^{B}$ are shared basis matrices and a_{rb} are learned relation-specific coefficients. This reduces the number of parameters from $R \times d_{\text{in}} \times d_{\text{out}}$ to $B \times d_{\text{in}} \times d_{\text{out}} + R \times B$. The forward propagation rule combines self-loop and neighbor messages as

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{h}_i^{(l)} + \sum_{r=1}^{R}\sum_{j\in\mathcal{N}_i^r} c_{i,r}\mathbf{W}_r^{(l)}\mathbf{h}_j^{(l)}\right), \tag{8}$$

where $\mathbf{W}_0^{(l)}$ handles self-connections and $\sigma(\cdot)$ is a non-linear activation function. This formulation enables efficient learning over knowledge graphs while preserving relation-specific inductive biases through the decomposed weight matrices.

3.4 Training Objective

Following previous works (Feng et al., 2019; Li et al., 2024), we formulate the next-day stock return prediction as a *learning to rank* problem. We optimize a hybrid objective combining regression and ranking terms:

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \phi \cdot \mathcal{L}_{\text{rank}}, \tag{9}$$

where ϕ is a hyperparameter controlling the contribution of ranking supervision. We set it to be 0.5 here. The regression loss is defined as:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i^{t+1} - r_i^{t+1}\right)^2, \tag{10}$$

where N is the number of stocks, $\hat{y}_i^{t+1}$ is the predicted score, and r_i^{t+1} is the ground truth return.

The ranking loss is formulated as:

$$\mathcal{L}_{\text{rank}} = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\max\Big(0, (\hat{y}_i^{t+1} - \hat{y}_j^{t+1}) \cdot (r_j^{t+1} - r_i^{t+1})\Big), \tag{11}$$

which penalizes cases where the predicted ordering contradicts the ground truth ordering. For implementation, we compute all pairwise differences in predictions and ground truth, multiply them element-wise, and apply a ReLU to retain only positive ranking violations.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate on S&P 500 constituents using daily OHLCV data enriched with technical indicators (MA, RSI, MACD). Our chronological split spans training (2012/07-2022/06), validation (2022/07-2023/09), and testing (2023/10-2024/12). We filter out the companies which do not have full trading records during this period, resulting in 452 tickers in total. Critically, our static graph uses GPT-4o-mini (knowledge cutoff: October 2023) [2], with testing beginning October 2023 - ensuring true out-of-sample evaluation where LLM-extracted relationships are tested on genuinely unseen future data. See Appendix A for detailed feature descriptions and data splits.

Evaluation Metrics We evaluate our model using three complementary metric categories: ranking quality using Mean Reciprocal Rank (MRR), prediction accuracy with Mean Squared Error (MSE), and trading performance with the cumulative Investment Return Ratio (IRR) and Sharpe Ratio (SR). Specifically, We construct two long-only strategies by selecting the top-1 and top-10 stocks based on predicted rankings. At the start of the test period, we invest one unit of capital in each strategy and compute their cumulative profit and Sharpe ratio over the evaluation horizon.

Baselines We compare the proposed method against two commonly used predefined stock relationship graphs: (1) Wikidata-based relationships extracted from a structured knowledge base, and (2) the GICS industry-sector classification reflecting economic sector groupings. These baselines enable evaluation of the efficacy of the LLM-extracted relationship graph relative to established graph constructions. We follow the same process as Feng et al. (2019), the details are in Appendix B.

4.2 Relation Extraction Analysis

Before evaluating the performance on downstream tasks, it is crucial to verify that the LLM-extracted company relationship graph is structurally richer and more diverse than existing alternatives. We quantitatively compare the number and distribution of relations against two commonly used predefined graphs (GICS industry-sector hierarchy and Wikidata corporate relations). As shown in Table 2, compared to the baselines, our LLM-extracted graph exhibits clear advantages in relational diversity, coverage, and structural realism. It contains 13 distinct relation types, far exceeding the 4 in both Wikidata and GICS, enabling richer semantic mod-

[2] https://platform.openai.com/docs/models/gpt-4o-mini

Relations	**MSE** ($\times 10^{-4}$) $\downarrow$	**MRR** $\uparrow$	**IRR (1)** $\uparrow$	**SR (1)** $\uparrow$	**IRR (10)** $\uparrow$	**SR (10)** $\uparrow$
RGAT						
Wikidata	3.269	0.016	0.177	0.408	0.206	0.901
GICS	5.311	0.016	0.273	0.615	0.237	1.060
Ours (LLM)	**3.196**	**0.035**	**0.421**	**0.835**	**0.262**	**1.185**
RGCN						
Wikidata	3.190	0.026	0.254	0.558	0.258	1.038
GICS	3.190	0.022	0.600	1.011	0.301	1.090
Ours (LLM)	**3.189**	**0.027**	**0.820**	**1.176**	**0.350**	**1.246**

Table 1: Performance comparison of RGCN and RGAT models on different relationship graphs. We report the average performance across 40 runs. "(1)" and "(10)" mean the top-1 and top-10 strategy, respectively.

Graph	R	$\lvert E\rvert$	Nodes	Sym.
Wikidata	4	4,723	358	✓
GICS	4	19,732	452	✓
Ours(LLM)	13	12,800	452	✗

Table 2: Basic statistics of the relationship graphs. R denotes the number of distinct relation types, $|E|$ denotes the total number of edges, "Nodes" denotes the number of unique companies covered, and "Sym." indicates whether the graph is symmetric.

eling. While its total number of edges (12,800) is lower than GICS (20,636), it is substantially denser than Wikidata (4,723), striking a balance between diversity and connectivity. In terms of coverage, it spans 452 companies, matching GICS and surpassing Wikidata's 358, ensuring applicability to the full stock universe. For both Wikidata and our LLM-generated relations, we filtered out relation types with fewer than 200 edges to maintain graph connectivity, ensuring that the comparison focuses on meaningful and well-connected relations. Importantly, our graph incorporates directed edges, capturing asymmetric corporate relationships (e.g., supplier–customer) that symmetric baselines cannot represent, thereby offering a more realistic and informative foundation for downstream stock ranking tasks.

4.3 Stock Ranking Performance

As shown in Table 1, the LLM-extracted relations consistently outperform Wikidata and GICS across both RGAT and RGCN backbones. In terms of prediction accuracy, our method achieves the lowest MSE in all cases, showing that LLM-extracted relations better capture stock dynamics. The gains are even clearer in ranking quality: MRR nearly doubles under RGAT (0.035 vs. 0.016) and remains the strongest under RGCN, indicating that our approach identifies top-performing stocks more effectively.

The improvements translate into substantial trading benefits. For both top-1 and top-10 strategies, our relations deliver markedly higher cumulative returns and Sharpe ratios, with the RGCN backbone achieving over 1.0 in top-1 Sharpe ratio (1.176), demonstrating strong risk-adjusted profitability. These consistent gains across two distinct architectures highlight that the advantage comes from the richer, directional relations themselves rather than a specific model choice, validating our claim that LLM-extracted structures bridge the gap between generic knowledge graphs and actionable financial insights.

5 Conclusions & Future Work

This work demonstrates that high-quality stock relationship graphs extracted from large language models can significantly enhance stock ranking accuracy and trading performance compared to widely used knowledge sources such as Wikidata and GICS. By leveraging LLMs' implicit knowledge of corporate relationships, we achieve 37% higher investment returns and improved Sharpe ratios across multiple evaluation strategies. To the best of our knowledge, we are the first to use zero-shot LLM extraction for financial graphs, opening a new research direction - using LLMs as knowledge bases for finance. Future work will explore temporal dynamics to capture evolving corporate relationships while maintaining stability, and investigate multi-modal integration combining LLM knowledge with alternative data sources and market sentiment.

References

Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. 2019. Relational graph attention networks. *Preprint*, arXiv:1904.05811.

Qinkai Chen and Christian-Yann Robert. 2022. Multivariate realized volatility forecasting with graph neural network. In *Proceedings of the Third ACM International Conference on AI in Finance*, ICAIF '22, page 156–164, New York, NY, USA. Association for Computing Machinery.

Rui Cheng and Qing Li. 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):55–62.

Zehua Cheng, Lianlong Wu, Thomas Lukasiewicz, Emanuel Sallinger, Georg Gottlob, Georg Gottlob, André Hernich, Clemens Kupke, Thomas Lukasiewicz, Marcelo Arenas, and 1 others. 2022. Democratizing financial knowledge graph construction by mining massive brokerage research reports. In *Proceedings of the Workshops of the {EDBT/ICDT} 2022 Joint Conference, Edinburgh, UK, March 29, 2022*, volume 846, pages 453–473. CEUR-WS. org.

Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Trans. Inf. Syst.*, 37(2).

Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. 2024. Master: market-guided stock transformer for stock price forecasting. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2021. Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Manali Patel, Krupa Jariwala, and Chiranjoy Chattopadhyay. 2024. A systematic review on graph neural network-based methods for stock market forecasting. *ACM Comput. Surv.*, 57(2).

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):497–504.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. *Preprint*, arXiv:1703.06103.

Milena Trajanoska, Riste Stojanov, and Dimitar Trajanov. 2023. Enhancing knowledge graph construction using large language models. *Preprint*, arXiv:2305.04676.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *Preprint*, arXiv:1710.10903.

Changhai Wang, Hui Liang, Bo Wang, Xiaoxu Cui, and Yuwei Xu. 2022. Mg-conv: A spatiotemporal multi-graph convolutional neural network for stock market index trend prediction. *Computers and Electrical Engineering*, 103:108285.

Heyuan Wang, Tengjiao Wang, Shun Li, and Shijie Guan. 2023. Hatr-i: Hierarchical adaptive temporal relational interaction for stock trend prediction. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6988–7002.

Xingkun Yin, Da Yan, Abdullateef Almudaifer, Sibo Yan, and Yang Zhou. 2021. Forecasting stock prices using stock correlation graph: A graph convolutional network approach. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

A Dataset Details

For each S&P 500 constituent, we collect daily Open, High, Low, Close prices and Volume (OHLCV)[3]. We compute standard technical indicators including Moving Averages (MA), Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD). These raw values are normalized to form time-series feature vectors $\mathbf{x}_i^t$ that initialize node representations in our model.

Table 3 details our chronological split, designed to simulate realistic live trading conditions:

Split	**Period**	**Days**
Train	2012/07 - 2022/06	2517
Validation	2022/07 - 2023/09	313
Test	2023/10 - 2024/12	315

Table 3: Chronological split of the dataset for training, validation, and testing.

The validation set is used for hyperparameter tuning and early stopping. By aligning our test period start with GPT-4o-mini's knowledge cutoff[4], we ensure the fundamental relationships extracted

[3] https://paperswithbacktest.com/datasets/stocks-daily-price

[4] https://platform.openai.com/docs/models/gpt-4o-mini

by the LLM are evaluated on their ability to generalize to future, unseen market data, providing a rigorous assessment of predictive power.

B Baseline Details

GICS The Global Industry Classification Standard, jointly developed by MSCI and Standard & Poor's, is a widely adopted taxonomy for categorizing companies into a four-tier hierarchical structure consisting of sectors, industry groups, industries, and sub-industries. This system facilitates consistent classification and comparison of firms based on their primary business activities, enabling investors and analysts to construct sector-based investment strategies and benchmark performance. In our experiments, we utilize GICS four-level relationships as one of the baseline graphs. The statistics of each level are shown in Table 4.

Level	$\lvert E \rvert$	Categories
Sector	10,841	11
Industry Group	5,323	25
Industry	2,347	67
Sub-Industry	1,221	119

Table 4: Statistics of GICS relationships at different levels.

Wikidata Wikidata is a collaboratively edited knowledge base that provides structured information across a wide range of domains, including corporate entities and their interconnections. In our work, we extract company–company relationships from Wikidata by meta-paths defined by Feng et al. (2019). To ensure the connectivity and robustness of the resulting graph, we filter out relation types whose total number of connections is below 200, retaining only sufficiently frequent relations for analysis.

Meta-path	$\lvert E \rvert$
Industry - Industry	1,944
Member of - Member of	1,220
Owned by - Owned by	1,040
Product or material produced -	519

Table 5: Statistics of Wikidata relationships at different meta-paths.

Enhancing Financial RAG with Agentic AI and Multi-HyDE: A Novel Approach to Knowledge Retrieval and Hallucination Reduction

Akshay Govind Srinivasan[*1], **Ryan Jacob George**[*1], **Jayden Koshy Joe**[*1], **Hrushikesh Kant**[*1], **Harshith M R**[*1], **Sachin Sundar**[*1], **Sudharshan Suresh**[*1], **Rahul Vimalkanth**[*1], **Vijayavallabh**[*1]

[1]Indian Institute of Technology Madras

Abstract

Accurate and reliable knowledge retrieval is vital for financial question-answering, where continually updated data sources and complex, high-stakes contexts demand precision. Traditional retrieval systems rely on a single database and retriever, but financial applications require more sophisticated approaches to handle intricate regulatory filings, market analyses, and extensive multi-year reports. We introduce a framework for financial Retrieval Augmented Generation (RAG) that leverages agentic AI and the Multi-HyDE system, an approach that generates multiple, nonequivalent queries to boost the effectiveness and coverage of retrieval from large, structured financial corpora. Our pipeline is optimized for token efficiency and multi-step financial reasoning, and we demonstrate that their combination improves accuracy by 11.2% and reduces hallucinations by 15%. Our method is evaluated on standard financial QA benchmarks, showing that integrating domain-specific retrieval mechanisms such as Multi-HyDE with robust toolsets, including keyword and table-based retrieval, significantly enhances both the accuracy and reliability of answers. This research not only delivers a modular, adaptable retrieval framework for finance but also highlights the importance of structured agent workflows and multi-perspective retrieval for trustworthy deployment of AI in high-stakes financial applications.

1 Introduction

Large Language Models (LLMs) such as GPT-4 (OpenAI et al., 2024), LLaMA (Touvron et al., 2023), and PaLM (Chowdhery et al., 2022) have significantly advanced natural language processing, demonstrating strong capabilities in contextual reasoning and few-shot learning. These models are increasingly applied in high-stakes domains, including healthcare diagnostics (Singhal et al., 2023), legal document analysis (Henderson et al., 2023), and financial services (Wu et al., 2023; Li et al., 2023). Their ability to process and generate domain-specific, human-like responses offers clear potential benefits.

However, a persistent limitation of LLMs is *hallucination* - the generation of factually incorrect or fabricated content presented as truth (Ji et al., 2023; Huang et al., 2023). This limitation poses significant risks in domains where factual accuracy is paramount. In domains such as finance, where decisions must be based on accurate and verifiable data, hallucinations can lead to significant monetary losses, reputational harm, and regulatory violations.

Retrieval-Augmented Generation (RAG) frameworks (Lewis et al., 2020; Guu et al., 2020) address this issue by grounding LLM outputs in external knowledge sources. Conventional RAG pipelines use a retriever to fetch relevant document chunks from a database based on semantic similarity between vector embeddings (Karpukhin et al., 2020; Xiong et al., 2020). Improvements in retrieval have come from better embedding methods (Reimers and Gurevych, 2019; Gao et al., 2021), hybrid dense-sparse strategies, and hierarchical retrieval (Khattab and Zaharia, 2020; Zhang et al., 2022).

One particularly effective method for improving retrieval is Hypothetical Document Embeddings (HyDE) (Gao et al., 2023), where an LLM first generates a synthetic "hypothetical" answer to a query, embeds it, and then retrieves real documents most similar to that synthetic answer. This approach improves alignment between queries and relevant passages, especially in cases where the original query is underspecified or phrased differently than the source content.

Recent work in *Agentic RAG* (Schick et al., 2023; Qin et al., 2023; Yao et al., 2022; Liu et al., 2023) extends the static "retrieve-then-generate" pipeline

[*]These authors contributed equally to this work

into a dynamic decision-making process. Here, the LLM acts as an orchestrator, capable of decomposing complex queries, selecting appropriate tools or retrieval strategies, performing multi-hop searches, and verifying intermediate results before generating a final answer. Such systems have shown particular promise in domains requiring multi-step reasoning and evidence verification, making them well-suited for financial question answering, where queries may range from straightforward fact lookups to multi-document analyses (Wang et al., 2025).

Financial QA systems must process vast repositories of unstructured data, including annual reports, regulatory filings, earnings call transcripts, and market analyses (Wu et al., 2023; Li et al., 2023). The retrieval strategy must be both accurate and efficient, as inadequate retrieval can lead to irrelevant or misleading context being passed to the LLM. This is especially problematic for multi-hop queries, where context mismanagement or excessive token usage can degrade performance despite the availability of long-context models. Methods that involve processing information in the data stores into structures like graphs result in increased upfront token costs, albeit with better performance. To address these challenges in financial question answering, we present the following contributions:

- **Multi-HyDE**: A retrieval mechanism that utilizes multi-perspective hypothetical documents bringing an improvement in retrieval accuracy without an increase in token costs over HyDE (Gao et al., 2023)
- A combination of dense and sparse retrieval strategies to maintain performance on vector stores with over 500,000 tokens.
- An Agentic system that is capable of handling both straightforward queries and ones requiring planning, multi-hop retrieval, tool calling and verification.

Details of our system have been discussed in detail in Section 3. Details about the evaluation set up have been discussed in Section 4.

2 Related works

2.1 Retrieval Methods

The efficacy of Retrieval-Augmented Generation (RAG) systems fundamentally depends on the quality of their retrieval component (Lewis et al., 2020; Guu et al., 2020). Traditional RAG implementations employ semantic similarity search over vector databases, but this approach often suffers from a semantic mismatch between concise queries and the verbose, context-rich nature of source documents (LangChain, 2023). To address this, recent research has focused on enhancements in three main categories: pre-retrieval query transformations, hybrid retrieval strategies, and post-retrieval processing.

Pre-retrieval Query Transformation Pre-retrieval Query Transformation bridges the semantic gap through sophisticated query manipulation. A seminal advancement is Hypothetical Document Embeddings (HyDE), which uses a language model to generate a "pseudo-document" representing an ideal answer. The embedding of this richer document is then used for retrieval, shifting the paradigm from a query-to-document to a more effective answer-to-answer similarity search (Gao et al., 2023). Parallel to this, multi-query strategies improve recall by generating several variations of a user's query to capture different facets of the information need (LangChain, 2023). However, generating merely similar queries can sometimes degrade precision (Eibich et al., 2024). Recent advances include DMQR-RAG (Diverse Multi-Query Rewriting) (Li et al., 2024), which operates at different information granularity levels, and MUGI (Multi-Text Generation Integration) (Zhang et al., 2024), a training-free approach that generates multiple pseudo-references to enhance both sparse and dense retrieval. While these approaches improve retrieval, they fundamentally rely on query similarity rather than the complementary diversity we propose.

Hybrid Retrieval Strategies Hybrid Retrieval Strategies combine sparse and dense methods to leverage both keyword matching and semantic similarity. Dense retrieval excels at capturing semantic connections but can struggle with exact term matching, while sparse methods like BM25 provide precise keyword matching. In the context of large, structured financial reports, methods relying on vector similarity alone often fail to retrieve all relevant information and struggle to disambiguate semantically similar sections that differ only in critical numerical or temporal details. Our framework explicitly integrates Multi-HyDE with BM25 in a unified pipeline optimized for these documents, improving coverage and disambiguation.

Post-retrieval Processing Post-retrieval Processing has evolved beyond simple re-ranking to incorporate sophisticated correction mechanisms. For instance, CRAG introduces a retrieval evaluator that assesses document quality and triggers corrective actions, like web searches, when quality is insufficient (Yan et al., 2024). Self-RAG trains language models to adaptively retrieve passages and self-critique through generated reflection tokens (Asai et al., 2023). MAIN-RAG proposes a multi-agent filtering framework where agents collaboratively score retrieved documents (Chang et al., 2024). While promising, these systems introduce computational overhead to fix retrieval issues. Our approach therefore also emphasizes improving retrieval quality from the outset to reduce the need for extensive correction.

Our Multi-HyDE generates multiple non-equivalent but contextually related queries. Unlike methods that create semantically similar queries, our approach creates distinct but complementary information needs—for instance, generating separate queries about a company's fraud investigations and its criminal cases that might be answered within the same document context.

2.2 Agentic RAG

The static retrieve and generate workflow of traditional RAG is insufficient for complex queries that require multi-step reasoning and dynamic information gathering. This has spurred the development of Agentic RAG, which embeds autonomous agents into RAG pipelines to create dynamic problem-solving systems.

Finite State Machine Approaches Finite State Machine Approaches structure agentic workflows through formal state management. StateFlow models language model workflows as finite state machines, distinguishing between "process grounding" via states and "sub-task solving" through actions (Wu et al., 2024). This approach has achieved 13-28% higher success rates than ReAct on benchmarks while reducing costs by 3-5×. Our work extends this paradigm. In contrast to prior work applying state management primarily to retrieval and generation, we extend it to govern all tool calls issued by the language model, enabling coherent reasoning across multiple modalities.

Multi-Agent Architectures Multi-Agent Architectures coordinate specialized agents for complex tasks. MAIN-RAG exemplifies this with its multi-agent filtering system (Chang et al., 2024). However, such multi-agent systems can suffer from increased complexity and failure points.

2.3 RAG in Finance

Financial RAG systems face unique challenges due to complexity, precision, and regulation. These include handling 100+ page multi-year reports, disambiguating semantically similar sections, and managing numerical precision where subtle differences have significant implications. Specialized Financial Platforms have emerged to address these challenges.

Specialized Financial Platforms FinRobot provides a four-layer architecture with Financial AI Agents and Multi-source Foundation Models (Yang et al., 2024). While comprehensive, it lacks the specialized retrieval innovations for financial document disambiguation that our Multi-HyDE approach directly addresses.

FinSage focuses on regulatory compliance through a multi-aspect RAG framework, achieving 92.51% recall and a 24.06% accuracy improvement over baselines (Wang et al., 2025). However, FinSage relies on standard HyDE rather than our multi-perspective approach and uses curated questions instead of a comprehensive benchmark evaluation.

Financial Knowledge Graph Integration Financial Knowledge Graph Integration handles complex relationships through structured representations. While promising, knowledge graph approaches require significant upfront processing costs and may not adapt well to rapidly changing financial information. Our approach offers greater flexibility and lower preprocessing overhead while achieving comparable performance through retrieval optimization.

Evaluation Challenges Evaluation challenges in finance are complicated by the need for numerical precision. FinanceBench reveals that GPT-4-Turbo with retrieval systems incorrectly answers or refuses 81% of its questions (Islam et al., 2023). ConvFinQA highlights challenges in conversational queries requiring extensive calculations (Chen et al., 2022). These issues suggest that many existing systems may report inflated performance due to flawed evaluation methodologies. Our emphasis on human evaluation provides more accurate assessments for high-stakes applications. Our framework's modular design and reliability-focused architecture directly address enterprise deployment concerns often overlooked in academic research, demonstrating that retrieval optimization may provide greater returns than developing domain-specific language models alone.

In summary, existing RAG systems face key challenges including retrieval issues with semantic ambiguity in complex financial texts, limited capacity for multi-step reasoning and calculations, and inefficiencies due to complex architectures and flawed evaluations. Our framework addresses these by using Multi-HyDE with hybrid BM25 to improve retrieval accuracy and disambiguation, integrating an agentic tool usage system governed by unified state management for advanced reasoning, and reducing overhead by avoiding heavy knowledge graphs while relying on human evaluation for realistic performance assessment. This approach enhances retrieval reliability, reasoning capabilities, and system efficiency for financial RAG applications.

3 Methodology

To address the challenges outlined in Sections 1 and 2, we propose a retrieval-augmented generation (RAG) pipeline with the following key components:

- **Multi-HyDE:** A multi-hypothesis document expansion module that generates several hypothetical documents based on diverse variants of the input query. These documents are then used to retrieve semantically relevant content from the vector store.
- **Keyword-based Retrieval:** An auxiliary keyword-based retriever (e.g., BM25) designed to enhance retrieval performance for structured data such as tables, as well as for semantically similar documents (e.g., annual reports across different years).
- **Agentic Pipeline:** A multi-stage reasoning and retrieval process comprising:
    1. *Query Clarification:* The system first seeks to clarify the user's question, either through direct interaction with the user or by leveraging web search.
    2. *Initial Retrieval:* The clarified query is used to perform retrieval from the vector store using the components described above.
    3. *Iterative Refinement:* If the retrieved content is unsatisfactory, the system formulates a retrieval plan. This includes the ability to perform multi-hop retrievals, invoke external tools, and decompose the query into sub-queries.
    4. *Final Response:* Once the retrieved evidence is deemed sufficient, the system synthesizes and delivers the final answer to the user.

This integrated design allows the pipeline to combine the semantic strengths of vector-based retrieval with the precision of keyword-based methods, while also enabling dynamic reasoning for complex, multi-step information needs.

3.1 Multiple Hypothetical Dynamic Embeddings (Multi-HyDE)

For our main retrieval tool, we employ a combination of multi-query based retrieval (Eibich et al., 2024) and HyDE (Gao et al., 2023), which we call *Multi-HyDE*, along with BM25 based retrieval for tables and a re-ranker.

HyDE Gao et al. (2023) employ a generator g to create multiple hypothetical documents from a query q and retrieves real documents d_i from the dataset $\mathcal{D}$ that are similar to the hypothetical ones. N documents are sampled from g. An embedding model f is used to generate "hypothetical document embeddings" $\hat{v}$ for a query q as depicted in Equation 1.

$$\hat{v} = \frac{1}{N} \sum_{\hat{d}_i \sim g(q)} f(\hat{d}_i) \tag{1}$$

Multi-HyDE Multi-query approaches usually generate similar queries to the user's, but this has been shown to reduce retrieval precision (Eibich et al., 2024). Our approach instead uses an LLM g_q to generate queries $[q_1, q_2, ..., q_N]$ that may have answers present in the same context, following which it generates a hypothetical document for each query. These queries may take the form of similar queries, related queries with distinct meanings (such as including a query on fraud by a company A and a query on criminal cases by company A) or it may result in query decomposition. To the best of our knowledge, this particular approach has not been tried before. An embedding model f is used to generate "hypothetical document embeddings"

$\hat{v}_i \in \mathbb{R}^{\hat{d}_{embed}}$, as depicted in Algorithm 1. Our retriever h retrieves k_1 documents from $\mathcal{D}$, and we further use a reranker to select the top k_2 documents.

Algorithm 1 Multi-HyDE Retrieval

Require: query q, database $\mathcal{D}$, query and document generators g_q, g, embedding model f, retriever h, reranker r, hyperparameters N, k_1, k_2

$[q_1, \ldots, q_N] \leftarrow g_q(q)$
for each q_i in $[q_1, \ldots, q_N]$ **do**
  $\hat{v}_i \leftarrow f(g(q_i))$
  $S_i \leftarrow h(\hat{v}_i)$
end for
$d_{total} \leftarrow concat(S_1, S_2, ..., S_N)$
$d_{final} \leftarrow r(d_{total})$
return d_{final}

3.2 Agentic RAG

To address both simple and complex multi-hop queries, we employ an agentic system (Figure 1) equipped with several tools, including `edgar_tool`, `Alpha Vantage Exchange Rate`, `web_search`, and a Python calculator, as well as a retriever based on Multi-HyDE. Additional tools are listed in Appendix D.

The query processing begins with direct retrieval using Multi-HyDE, ensuring the system remains grounded in explicitly-included sources. Retrieved documents are then passed to the LLM Agent for reasoning and synthesis. If these documents are insufficient to fully answer the query, the LLM dynamically invokes available tools.

For improved performance, the LLM not only generates tool calls but also produces intermediate reasoning steps, user-facing responses, decomposed sub-queries, and a structured execution plan, inspired by Hao et al. (2023); Radhakrishnan et al. (2023); Zhou et al. (2023); Wang et al. (2023); Girhepuje et al. (2024). The full prompt is given in Appendix B. Queries are broken down into atomic steps, with each step resolved using the most suitable tool from the current toolset. The LLM evaluates intermediate results at each stage, adapting the plan when necessary to ensure accuracy and grounding.

This design supports highly dynamic workflows: tools can be added or removed on demand, enabling integration of custom data sources, access to live information, and execution of complex sequential reasoning processes. While standard RAG also grounds responses in retrieved documents, it typically relies on a single retrieval step, leaving the model prone to filling gaps with its latent knowledge if the evidence is incomplete. In contrast, Agentic RAG decomposes queries into atomic steps, validates intermediate results, and dynamically invokes additional tools or retrievals as needed. This iterative, evidence-driven process strengthens fidelity to verifiable sources, reduces hallucination, and produces more reliable answers across diverse and complex query types.

Algorithm 2 Agentic RAG System

Require: query q, database $\mathcal{D}$, set of tools T, LLM agent A

function PROCESS_QUERY$(q, \mathcal{D}, T, A)$
  $d_{initial} \leftarrow$ Multi-HyDE$(q, \mathcal{D})$
  LLM Agent history $H \leftarrow [q, d_{initial}]$
  loop
    A analyzes H to determine if the query can be answered
    if A determines an answer exists **then**
      Generate final answer from H
      return Final answer
    else
      A generates a sub-query q_{sub} and selects a tool $t \in T$
      $tool\_output \leftarrow t(q_{sub})$
      $H \leftarrow$ concat$(H, tool_output)$ ▷ Add tool's output to the LLM's history
    end if
  end loop
end function

4 Experimental setup

We ran our experiments using subsets of datasets (selection of subset is described in Appendix E) due to limited resources. We employ GPT-4o mini and the Mini-LM reranker for running the pipeline. Additional implementation details are included in Appendix G.

4.1 Evaluation datasets

We use a subset of questions from the FinanceBench (Islam et al., 2023) and ConvFinQA (Chen et al., 2022) datasets. From FinanceBench, we have selected from 150 human-annotated examples provided. These examples include evidence designated as ground truth context, with additional

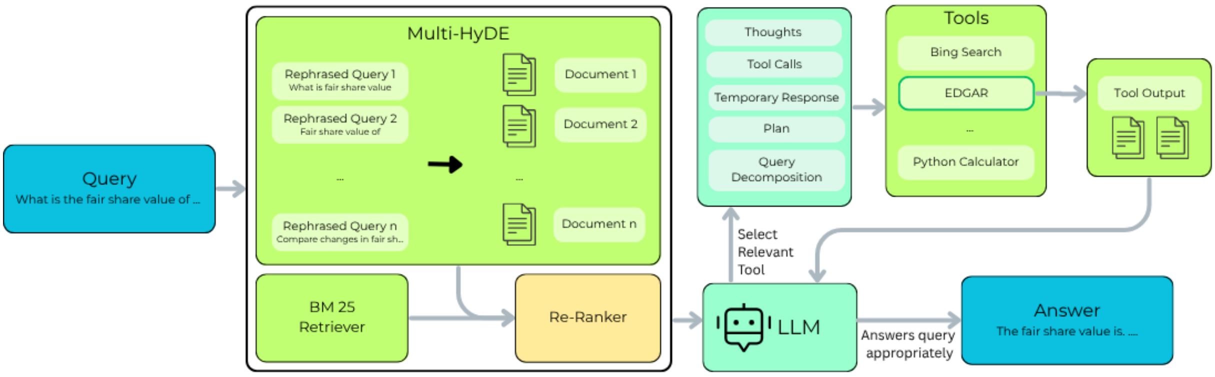


Figure 1: Our proposed agentic framework for financial question-answering.

justification considered when necessary. Appendix E provides details on the subsets used. **Furthermore, we ensure that the entire PDF document is added to the vector store in contrast to the ConvFinQA and FinanceBench datasets, which only pass the evidence pages.** We believe this better simulates real-world scenarios. This restricts us from comparing other retrieval methods where only evidence pages are passed as context. We also include a subset of filings and questions from financial-qa-10K[1] for a comparative study. An example of a question-answer pair is provided in Appendix A.

4.2 Metrics

We evaluate experiments and optimizations using **ROUGE,**[2] **Cosine Similarity**, as well as metrics from RAGAS (Es et al., 2023) and human evaluation. We use RAGAS with GPT-4o mini to calculate **Factual Correctness** (similar to the F1 Score) and **Faithfulness**. During human evaluation, accuracy and reliability are measured. The metrics are defined in Appendix F.1.

5 Experimental results and analyses

Method	Accuracy (%)	Reliability (%)
Multi-HyDE	34.4	37.91
Final Pipeline	**45.6**	**52.91**

Table 1: Human evaluation on subset of ConvFinQA and FinanceBench.

[1] https://huggingface.co/datasets/virattt/financial-qa-10K

[2] Low ROUGE scores in some experiments are attributed to the fact that the ground truth answers in the dataset consisted of only a single number, whereas large models explained their approaches.

Performance against other methods We provide a comparison of our pipeline against a representative method for retrieval optimization (HyDE), graph based knowledge organization (LightRAG) (Guo et al., 2024) and post-retrieval corrective measures (CRAG) (Yan et al., 2024). We include scores for Multi-HyDE with access to tools against these baselines.

Our results show improvement across all measures except Cosine Similarity: we achieve significant improvements in Recall, Facutal accuracy and Faithfulness (See table 2, 3) while having the same token costs involved as HyDE (since both generate the same number of hypothetical documents for a given user query) and avoid the upfront costs associated with graph based methods to create the graph.

Our approach supports dynamic vector stores - documents can be added or removed from the vector store without incurring additional costs, graph based approaches where removing information from the graph would incur some costs.

The results show the advantages of Multi-HyDE in the financial domain. We attribute the improved performance to the fact that financial reports across multiple years could have semantically similar content - pairing a dense retrieval method that identifies relevant information from an increased variety of potential sources and a sparse keyword based retriever to identify structured information improves overall performance by being able to handle more cases than any individual method.

Reliability Considerations: In verifying the LLM as a judge procedure utilized by RAGAS, we observe that in numerical examples, the LLM judge might provide incorrect evaluations (see Appendix F.2). Further, cases where the wrong answer is provided confidently has greater chance of ad-

Method	Cosine Similarity	Recall	Factual Correctness	Faithfulness	ROUGE score*
Multi-HyDE	0.6269	**0.3547**	**0.3849**	**0.8404**	**0.0594**
HyDE	0.7660	0.1154	0.2890	0.8290	0.0498
CRAG	0.7939	0.1556	0.0855	0.2521	0.0443
LightRAG	**0.7999**	0.0000	0.2434	0.4629	0.1632

Table 2: Evaluation Metrics for Different Methods on subset of **ConvFinQA + FinanceBench**.

Method	Cosine Similarity	Recall	Factual Correctness	Faithfulness	ROUGE score
Multi-HyDE	0.8976	0.8170	0.5205	**0.9352**	0.**4871**
HyDE	0.8883	0.6885	**0.5585**	0.8463	0.3726
CRAG	**0.9347**	**0.8500**	0.4708	0.7774	0.4290
LightRAG	0.7308	0.0000	0.0368	0.4629	0.3412

Table 3: Evaluation Metrics for Different Methods on subset on questions from financial-qa-10K .

verse impact that the system admitting to not having the exact answer. To confirm the performance of our proposed pipeline in light of the above challenges, we conduct a human evaluation of the responses with metrics reliabilty (fraction of confidently given answers which are correct rather than hallucinations) and accuracy (fraction of correct answers). Detailed definitions are provided in F.1.

Ablation study: In Table 4, we show that Multi-HyDE outperforms regular HyDE. We also perform a comparison between 2 rerankers `ms-marco-MiniLM-L-6-v2` (Cross Encoder) and `bge-reranker-v2-m3` (BGE) from huggingface. Though BGE is more performant, it is significantly more resource-intensive and slower. We also show that hybrid retrieval with BM25 clearly outperforms dense retrieval methods for long-document financial data. Tool calling does not improve accuracy, however it provides resiliency when some types of relevant data are not provided.

6 Future work

Agents and fine-tuning Small Language Models finetuned using parameter efficient techniques like LoRA(Hu et al., 2021) to be used as individual agents instead of relying on large closed source models, especially for tasks like query re-writing or hypothetical document generation, particularly to suit the language and format used in financial reports.

Better metrics for financial RAG Currently, LLM-based evaluation often incorrectly evaluates responses, especially when an answer is primarily numeric. Different evaluation systems may help improve this. In addition, a more comprehensive evaluation on complete datasets could be undertaken given more resources.

7 Conclusion

This research presents a novel approach to financial question answering, addressing key challenges in hallucination reduction and accurate information retrieval from complex financial documents. Our framework introduces Multi-HyDE, an extension of Hypothetical Document Embeddings that leverages multiple non-equivalent queries to enhance retrieval effectiveness. When combined with BM25 for tables and appropriate rerankers, Multi-HyDE demonstrates superior performance in capturing relevant information from financial corpora. Additionally, we developed and evaluated an agentic pipeline offering improved performance, capable of handling both simple queries, and ones requiring complex multi-hop retrieval and reasoning.

Our evaluation highlights the importance of specialized retrieval techniques for domain-specific applications and underscores the limitations of current LLM-based assessment metrics in financial contexts. Human evaluation proved crucial for accurately measuring performance, revealing substantial improvements with our ensembled approach. The modular design of our framework facilitates adaptation to other domains requiring precise information extraction. By addressing fundamental challenges in financial RAG systems, our work contributes to building more trustworthy AI systems for high-stakes applications where factual accuracy

Method	Cosine Similarity	Recall	Factual Correctness	Faithfulness	ROUGE score
1	0.8883	0.6885	0.5585	0.8463	0.3726
2	0.8932	0.7464	0.5539	0.8837	0.3575
3	0.8935	**0.8484**	**0.5868**	0.8768	0.3996
4	0.8976	0.8170	0.5205	**0.9352**	0.**4871**
5	**0.9119**	0.8033	0.5172	0.8298	0.4628
6	0.8935	**0.8484**	0.5867	0.8767	0.3996

Table 4: Effect of BM25, rerankers and tools on recall. (with financial-qa 10k dataset)

1. HyDE
2. Multi-HyDE + Cross Encoder Reranker
3. Multi-HyDE + BM25 + Cross Encoder Reranker
4. Multi-HyDE + BM25 + BGE Reranker
5. Multi-HyDE + BM25 + BGE Reranker without tools
6. Multi-HyDE + BM25 + Cross Encoder Reranker without tools

is paramount. Future research directions include fine-tuning models for financial contexts and developing more nuanced evaluation metrics.

Limitations

Due to resource constraints, our evaluation is conducted on a relatively small dataset, which may limit the generalizability of the results.

Although our approach demonstrates improvements over existing baselines, its practical deployment is still challenged by the presence of hallucinations in more complex and ambiguous datasets. Consequently, the system currently requires human oversight and verification to ensure reliability and factual consistency.

Acknowledgements

The authors thank InterIIT Tech Meet 13.0 and Pathway for proposing the problem statement and facilitating access to task materials and clarifications during the competition.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *Preprint*, arXiv:2310.11511.

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, and Na Zou. 2024. MAIN-RAG: Multi-Agent Filtering Retrieval-Augmented Generation. *Preprint*, arXiv:2501.00332.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. *Preprint*, arXiv:2210.03849.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Matouš Eibich, Shivay Nagpal, and Alexander Fred-Ojala. 2024. ARAGOG: Advanced RAG Output Grading. *Preprint*, arXiv:2404.01037.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. RAGAS: Automated Evaluation of Retrieval Augmented Generation. *Preprint*, arXiv:2309.15217.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Sahil Girhepuje, Siva Sankar Sajeev, Purvam Jain, Arya Sikder, Adithya Rama Varma, Ryan George, Akshay Govind Srinivasan, Mahendra Kurup, Ashmit Sinha, and Sudip Mondal. 2024. RE-GAINS & EnChAnT: Intelligent Tool Manipulation Systems For Enhanced Query Responses. *Preprint*, arXiv:2401.15724.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. LightRAG: Simple and Fast Retrieval-Augmented Generation. *Preprint*, arXiv:2410.05779.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv preprint*. ArXiv:2002.08909 [cs].

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with Language Model is Planning with World Model. *Preprint*, arXiv:2305.14992.

Peter Henderson, Koustuv Sinha, Nicolas Angelard-Gontier, Nan Rosemary Ke, Geneviève Fried, Ryan Lowe, and Joelle Pineau. 2023. Foundation models for legal reasoning. *arXiv preprint arXiv:2307.03557*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Yue Huang, Xiaohan Sun, Yao Xiong, Zhicheng Dou, Guoliang Zhang, and Jian Yuan. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. FinanceBench: A New Benchmark for Financial Question Answering. *Preprint*, arXiv:2311.11944.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

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. *arXiv preprint arXiv:2004.04906*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

LangChain. 2023. Query Transformations.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Zhicong Li, Jiahao Wang, Zhishu Jiang, Hangyu Mao, Zhongxia Chen, Jiazhen Du, Yuanxing Zhang, Fuzheng Zhang, Di Zhang, and Yong Liu. 2024. Dmqr-rag: Diverse multi-query rewriting for rag. *Preprint*, arXiv:2411.13154.

Zhuangzhuang Li, Hanyi Wang, Zhengqing Chen, and Xia Chen. 2023. Finbert: A pre-trained financial language representation model for financial text mining. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*.

Yujia Liu, Yuwei Xie, Chunyuan Chen, Sylvia Wang, Yuxin Yuan, Yang Liu, Xiang Hu, Songyang Wang, Tianyu Qiao, Lingyu Pan, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yujia Qin, Shengding Deng, Furui Xu, Shiwei Chen, Yankai Lin, Weilin Sun, Meng Bu, Peng Li, Shulin Zhou, Chao Yang, and 1 others. 2023. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*.

Ansh Radhakrishnan, Karina Nguyen, Anna Chen, Carol Chen, Carson Denison, Danny Hernandez, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Sam McCandlish, Sheer El Showk, Tamera Lanham, Tim Maxwell, Venkatesa Chandrasekaran, and 5 others. 2023. Question decomposition improves the faithfulness of model-generated reasoning. *Preprint*, arXiv:2307.11768.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Venkataraman, Gabriel Maginnis, Arun Nori, and 1 others. 2023. Large language models in medicine. *Nature Medicine*, 29(8):1998–2012.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *Preprint*, arXiv:2305.04091.

Xinyu Wang, Jijun Chi, Zhenghan Tai, Tung Sum Thomas Kwok, Muzhi Li, Zhuhong Li, Hailin He, Yuchen Hua, Peng Lu, Suyuchen Wang, Yihong Wu, Jerry Huang, Jingrui Tian, Fengran Mo, Yufei Cui, and Ling Zhou. 2025. Finsage: A multi-aspect rag system for financial filings question answering. *Preprint*, arXiv:2504.14493.

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 preprint arXiv:2303.17564*.

Yiran Wu, Tianwei Yue, Shaokun Zhang, Chi Wang, and Qingyun Wu. 2024. StateFlow: Enhancing LLM Task-Solving through State-Driven Workflows. *Preprint*, arXiv:2403.11322.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective Retrieval Augmented Generation. *Preprint*, arXiv:2401.15884.

Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, and Christina Dan Wang. 2024. FinRobot: An Open-Source AI Agent Platform for Financial Applications using Large Language Models. *Preprint*, arXiv:2405.14767.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Barret Zhang, Jonathon Shlens, and Jeff Dean. 2022. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024. Exploring the best practices of query expansion with large language models. *Preprint*, arXiv:2401.06311.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

A Question-answer example

```
{
  "question": "For American Water
  ↪  Works, what was the rate of
  ↪  growth from 2013 to 2014 in the
  ↪  fair value per share"
  "answer" : "",
  "context": "~13.3%. Page 81,
  ↪  Table[The
  weighted-average assumptions used in
  ↪  the
  Monte Carlo simulation and the
  ↪  weighted-average
  grant date fair values of RSUs
  ↪  granted
  for the years ended December 31]

  [45.45 -40.13]/40.13 = 13.3%

  AMERICAN WATER WORKS COMPANY, INC.",
  "ticker": "AWK",
  "filed on": "31 December 2015"
}
```

B Meta-Plan JSON Instructions

```
{
  "thought": "...",  # Thought process
  ↪  and reasoning of the bot for the
  ↪  current step
  "tool_calls": [{"name": "...",
  ↪  "args": {...}},
  {"name": "...", "args": {...}},
  ↪  ...], # List of tools to be
  ↪  called along with the
  ↪  appropriate arguments.
  "audio": "...", # Respond
  ↪  comprehensively to the query in
  ↪  a verbose way and output in
  ↪  formatted markdown string
  "plan": "...",
  # The overall plan for calling
  ↪  various tools and answering the
  ↪  query. This needs to be updated
  ↪  dynamically based on the
  ↪  retrieved information from tool
  ↪  calls.
  "queries":
  ↪  [{"query":"...","answer":"..."},
  {"query":"...","answer":"..."}]
```

```
    # The decomposed sub-queries.
    ↪  Intially all the answers are
    ↪  empty strings, as answers from
    ↪  tool calls come in, update them
    ↪  accordingly
}
```

C Retrieval Challenges

In the financial domain, retrieval methods that rely solely on vector similarity often fail to distinguish between passages that are semantically similar but differ in critical numerical details or temporal references. These distinctions, although subtle, are essential for producing accurate and trustworthy responses when answering questions about structured financial reports.

Consider the following example from our evaluation set:

```
"query": "For American Water Works, what
↪  was the growth in allowance for
↪  other funds used during construction
↪  from 2013 to 2014?"

"retrieved_reference_1": "In 2014, we
↪  spent $3.6 million, including $0.8
↪  million funded by research grants...
↪  (discussion on research and
↪  development spending)
↪  [awk_2015_10K.pdf]"

"retrieved_reference_2": "Cash flows
↪  used in investing activities
↪  increased in 2014 compared to 2013
↪  primarily due to an increase in our
↪  capital expenditures... (details on
↪  capital expenditures)
↪  [awk_2015_10K.pdf]"

"retrieved_reference_3": "Amortization
↪  of contributions in aid of
↪  construction was $23,913, $22,363,
↪  and $20,979 for the years ended
↪  December 31, 2014, 2013, and 2012...
↪  (amortization details)
↪  [awk_2015_10K.pdf]"

"retrieved_reference_4": "Such grants
↪  reduce the cost of research and
↪  allow collaboration with leading
↪  national and international
↪  researchers... (discussion on
↪  research grants and collaboration)
↪  [awk_2017_10K.pdf]"

"retrieved_reference_5": "Amortization
↪  of contributions in aid of
↪  construction was $27, $26, and $24
↪  for the years ended December 31,
↪  2016, 2015, and 2014... (further
↪  amortization details)
↪  [awk_2017_10K.pdf]"
```

As shown above, SEC filings from different years (e.g., 2015 vs. 2017) often include passages with similar or even nearly identical phrasing. However, for financial question-answering, distinctions such as the reporting year or specific numerical values are vital for correctness. Standard dense retrieval models tend to conflate these passages due to their semantic resemblance, leading to unreliable results.

To mitigate this issue, we incorporate BM25 alongside dense vector retrieval. This hybrid approach ensures that keyword and phrase-level matches (e.g., exact years, financial figures, or domain-specific terminology) complement semantic similarity, resulting in more precise and contextually appropriate retrievals.

D Tools

Our agentic pipeline has various tools to fetch data from various data sources apart from the retrieved context. The tools are divided into different types based on their use cases give below. Having more than one tools provide redundancy in case one or more tools fail.

1. **Web-search:** The web search tool provides real-time access to the web search queries providing access to news, web pages, and more which might not be there in the retrieved context. **SERP API**, **Bing Web Search**, and **DuckDuckGo Web Search** are the tools used by the agent to obtain the data from a web search.

2. **Financial Data API:** This is a collection of tools that provide real-time as well as histori-

cal data about the prices of stocks, securities, and cryptocurrencies. **Yahoo Finance**, **Alpha Vantage**, **EDGAR Tool**(Electronic Data Gathering, Analysis, and Retrieval system) and **Financial Modelling Prep** tool providing real time financial data from various exchanges.

3. **Mathematical tools:** The **WolframAlpha API** and **Python Calculator** are the tools incorporated to provide the the data processing ability to the agent. WolframAlpha takes in the mathematical questions in natural language and provides us with the answer whereas python calculator can be used to help the agent with more menial calculations.

E Dataset

Owing to the inconsistent evaluation results often observed in LLM-based methods and limited computational resources, we conduct our experiments on a focused subset of the FinanceBench and ConvFinQA datasets. Specifically, we select reports with the highest density of associated questions to ensure the relevance and informativeness of our evaluation. The selected subset comprises SEC 10-K filings from the following companies:

- **American Water Works**: 2015, 2017, 2018
- **AMD**: 2022
- **American Express**: 2022
- **Boeing**: 2022

F Evaluation

F.1 Definitions of metrics

RAGAS defines metrics by comparing the facts in a model's answer to those in the retrieved context or ground truth. The *Faithfulness Score* is RAG-specific, measuring the proportion of claims in the answer that are supported by the retrieved context. *Factual Correctness*, based on the F1 score, can be applied to any model.
Faithfulness:

$$\text{Faithfulness} = \frac{\text{Supported claims}}{\text{Total claims in answer}} \tag{2}$$

Factual Correctness: Let

$$\begin{aligned}
\text{TP} &= \#\{\text{claims in answer present in reference}\}\\
\text{FP} &= \#\{\text{claims in answer not in reference}\}\\
\text{FN} &= \#\{\text{claims in reference not in answer}\}
\end{aligned}$$

Then

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

$$\text{F1} = \frac{2 \circ \text{Precision} \circ \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

For human evalution, we define accuracy (A) as the fraction of correct answers[3] and reliability (R) as the fraction of answers where the LLM was confident about its answer being correct. This is related to the hallucination rate (the fraction of incorrect answers, denoted HR) as

$$HR = \frac{A}{R} - A, \text{where} \tag{6}$$

$$\begin{aligned}
U &= \text{Number of times LLM refused to answer}\\
N &= \text{total number of answers}
\end{aligned}$$

In Table 1, we rely on human evaluation to show that the Agentic pipeline performs better than Multi-HyDE with tools.

F.2 Examples in which RAGAS fails

In addition to errors in the LLM evaluation used by RAGAS, we note that the evaluation often fails when more precise numbers are given in the generated answer than available in the ground truth. This is partly due to ambiguities in the dataset, but also because, for instance, it often penalizes the model for minor inaccuracies or rounding, and at other times it gives a high score despite an outright refusal to answer.

Listing 1: RAGAS incorrectly gives a factual correctness of 0 though the response is simply more precise than the reference.

```
Question: For American Water
    Works, what was the rate of
    growth from 2013 to 2014 in
    the fair value per share
Response: ```json
{
  "tool_calls": []
}
```
```

[3]In cases when a question requires multiple independent answers, we assign the score as the fraction of correct answers for that question
```

```
Based on the provided text, the
    fair value per share for
    American Water Works was $37
    .40 in 2012, $40.13 in 2013,
    and $45.45 in 2014 (Document
    8).  Therefore, the growth
    rate from 2013 to 2014 is:

($45.45 - $40.13) / $40.13 =
    0.1325 or 13.25%



Reference: 13.30%
Semantic Similarity:
    0.5918684811439618
Factual Correctness: 0.0
Faithfulness: 0.6666666666666666
```

Listing 2: RAGAS incorrectly scores the LLM's refusal to answer very highly

```
Question: For American Water
    Works, what was the rate of
    growth from 2013 to 2014 in
    the fair value per share
Response: The provided text does
    not contain information on the
     fair value per share of
    American Water Works for 2013
    and 2014, nor does it provide
    the means to calculate a
    growth rate.  Therefore, I
    cannot answer your question
    using the given documents.  To
     answer this question,
    additional information is
    needed.  I would need to
    access financial statements or
     other reliable sources
    containing this data.

Reference: 13.30%
Semantic Similarity:
    0.5736175582900219
Factual Correctness: 0.91
Faithfulness: 1.0
```

G Additional Implementation Details

PDF Parsing We use a customised PDF parsing system (Fig. 2) using the **Docling** library to extract and structure data from complex documents. It handles text, tables, and images, exporting tables in HTML format. Further, it utilizes recursive chunk splitting as the text chunking strategy for context preservation.

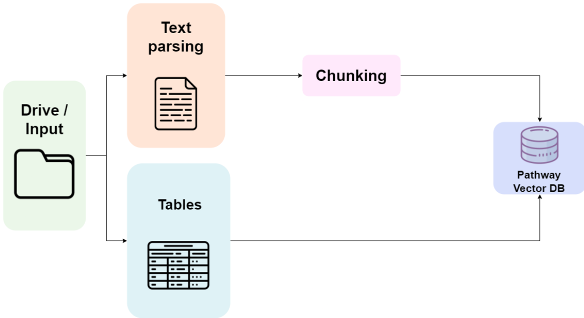


Figure 2: PDF-Parser System: Documents can be added to a Google Drive, which dynamically updates the Vector Store. Text is parsed using our parsing pipeline, chunked recursively before populating the Vector Store.

Retrieval We employ **HNSW** for indexing and in addition use **BM25** for retrieving tables. Our Vector Store is implemented using Pathway [4]. Row and column aggregation is also performed on tables. Keeping modularity in focus, retrieval methods are represented as tools, alongside others like web search and calculator. The **Multi-HyDE** retriever, selects the top $K_1 = 10$ chunks, while the BM25 retriever fetches the top $K_2 = 15$ chunks.

Reranking A re-ranker[5] is employed to pick the top $K = 8$ relevant chunks. This was determined after evaluating performance on various values of K, as shown in Table 5.

Top K Value	Accuracy (%)
1	57.5
2	75.3
8	79.6
10	**80.1**

Table 5: Accuracy for different values of 10 K retrieved documents.

[4] https://github.com/pathwaycom/pathway

[5] https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-6-v2 or https://huggingface.co/yxzwayne/bge-reranker-v2-m3, specified in our experiments

H Other Ablations

Tables below depict other ablations performed as a part of our experimentation and analysis.

Method	**Precision**	**Recall**	**Accuracy**
Naive-RAG	0.912	0.592	0.616
HyDE	TBD	TBD	TBD
Multi-HyDE	0.932	0.625	0.721

Table 6: Comparison of Naive-RAG, HyDE, and Multi-HyDE on a subset of financial-qa-10K dataset

Method	**In-Tokens**	**Out-Tokens**	**Time Spent**
Naive-RAG	-	-	0.199s
HyDE	133.5	428.2	9.344s
Multi-HyDE	193.6	421.4	9.121s

Table 7: Resource usage comparison on Financial-qa-10K Dataset.

Metric	**Method 1**	**Method 2**
Cosine Similarity	**0.5981**	0.5765
Recall	0.2462	**0.2910**
Factual Correctness	0.1738	**0.2291**
Faithfulness	0.8103	**0.8754**
ROUGE score	0.0346	**0.0349**

Table 8: Evaluation metrics comparing different parsers, showing the improvement of Docling over Open Parse (with ConvFinQA and FinBench dataset subsets)

Method 1: Multi-HyDE + Re-ranker + Open Parse (Llama-70b)
Method 2: Multi-HyDE + Re-ranker + Docling Parser (Llama-70b)

StockGenChaR: A Study on the Evaluation of Large Vision-Language Models on Stock Chart Captioning

Le Qiu and Emmanuele Chersoni
Department of Language Science and Technology
The Hong Kong Polytechnic University
lani.qiu@connect.polyu.hk, emmanuele.chersoni@polyu.edu.hk

Abstract

Technical analysis in finance, which aims at forecasting price movements in the future by analyzing past market data, relies on the insights that can be gained from the interpretation of stock charts; therefore, non-expert investors could greatly benefit from AI tools that can assist with the captioning of such charts.

In our work, we introduce a new dataset *StockGenChaR* to evaluate large vision-language models in image captioning with stock charts. The purpose of the proposed task is to generate informative descriptions of the depicted charts and help to read the sentiment of the market regarding specific stocks, thus providing useful information for investors[1].

1 Introduction

In finance, technical analysis is the discipline that aims to predict future price trends based on historical data (Ponsi, 2016; Edwards et al., 2018). Analysts usually anticipate price directions by inspecting *stock charts*, where they are represented in the form of wave patterns (see Figure 1). Such patterns allow traders to identify potential entry or exit positions and make informed investment decisions.

Reading those charts, however, requires specific financial knowledge and expertise, and it can be challenging for non-expert traders to extract useful insights from merely visual cues. Therefore, we would like to propose framing this problem as an image captioning task where, given an image, an automatic system has to produce a description of its content (Vinyals et al., 2015). In our view, automating the manual interpretation process and producing descriptive texts for the charts would open up a shortcut to understanding the market dynamics for various groups of users, including experienced traders, novices, and individuals seeking support for their fast-paced trading activities.

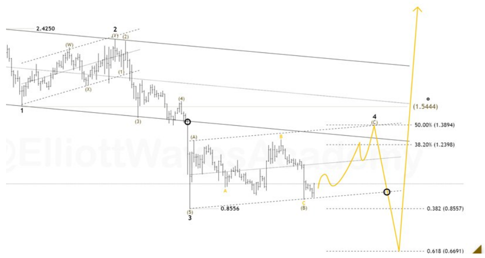


Figure 1: Future outlook for ARBUSD on the daily time frame based on Elliott-wave Theory (Elliott Waves Academy, 2024). The yellow line indicates the predictive movements of the stock.

Previous evaluation work on image captioning largely revolved around general-domain data, with relatively limited coverage of specialized domains [2]; and it typically aimed at summarizing the image content with just 1-2 concise sentences (Bernardi et al., 2016), which might not be ideal for the goals of technical analysis and financial decision-making. The image captioning setting needs to be adapted for stock chart reading: given an annotated chart image I, a system should generate a multi-sentence description C that provides a holistic narrative of the chart, covering the past movements and predictive trends, and ideally with trading advice. The generated text C is expected to be accurate and informative to lead to a well-grounded conclusion for the audience. Additionally, the text should remain as concise and comprehensible as possible, in order to be easily understandable even by less experienced traders.

To this purpose, we introduce *StockGenChaR*, a new dataset for the re-formulated stock-chart captioning task. To establish baseline performance

[1] The data and code will be made available on https://github.com/Laniqiu/GenChaR

[2] A summary of evaluation datasets for Image Captioning can be found in Table 4 in the Appendix.

levels, we tested some representative LVLMs (i.e., LLMs with visual capabilities (Li et al., 2023b)) by using metrics that focus on different aspects of the generated texts, including n-gram overlap, semantic similarity, sentiment alignment, and accuracy of metadata information.

2 Related Work

Prior work around stock charts focused on utilizing the numerical data for goals such as financial return prediction and portfolio optimization (Hu et al., 2018; Kusuma et al., 2019; Ho and Huang, 2021; Norasaed and Siriborvornratanakul, 2024), while other studies made use of the graphical component in image or pattern recognition tasks (e.g., Velay and Daniel, 2018; Zheng et al., 2021). However, to our knowledge, the task of stock chart captioning has received limited attention so far.

The most recent approach to the image captioning problem consists of the vision-language pre-training approach (VLP). VLP models are pre-trained on a large amount of image-text pairs, and then fine-tuned for downstream tasks (Gan et al., 2022; Chen et al., 2023). Popular VLP models that can be applied to image captioning tasks include, for example, SimVLM (Li et al., 2019), OSCAR (Li et al., 2020b) and CLIP (Radford et al., 2021). Large Vision Language Models (LVLMs) can be considered as enhanced and ready-to-use versions of VLP models: in recent research work, models such as GPT-4 Vision (OpenAI, 2023), Gemini (Gemini Team Google, 2024), BLIP-2 (Li et al., 2023a) and LLaVa (Liu et al., 2024) proved their ability of successfully carrying out several multimodal tasks, including image captioning and visual reasoning (Li et al., 2023b; Zhang et al., 2024).

There have been examples of customized LVLMs that have exhibited some chart reasoning abilities. For example, Liu et al. (2023) developed MMCA, a MultiModal Chart Assistant achieving state-of-the-art performance on several chart question answering benchmarks; they also introduced a new and more challenging benchmark with nine different tasks evaluating reasoning capabilities over charts and plots, and reported that even the most sophisticated LVLMs have important limitations in interpreting charts. The works of Bhatia et al. (2024) and Xie et al. (2024) both introduced large instruction datasets for tuning LVLMs for the financial domain, together with two models, FinTral and FinLLaVA, that excel in solving tasks related to the interpretation of financial tables and charts.

Although such works challenge models in visual question answering on charts, we believe that framing stock chart understanding as an image captioning task would be closer to the needs of investors and practitioner in the financial industry, as image captioning could provide trend interpretations beyond the constrained setting of question-based benchmarks. To our knowledge, this type of task is not covered by any of the existing benchmarks for financial chart understanding.

In our study, we aim at filling this gap by building a new dataset for stock chart captioning, *StockGenChaR*. We will also present a systematic evaluation of the most commonly used LVLMs on the new benchmark.

3 Dataset Creation

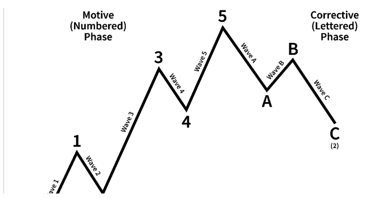


Figure 2: An EWP cycle.[3]

3.1 Sample Collection

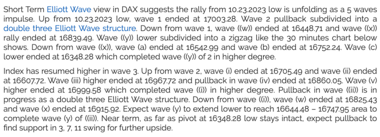
We identified an ideal source for this chart captioning task: the *ElliottWave-Forecast* website [4]. ElliotWave-Forecast is a worldwide top-notch technical analysis company, providing a wide range of coverage across about 80 markets, including Forex, Commodities, World Indices, and U.S. stocks & ETFs (ElliottWave-Forecast, 2024). The analyst team uses Elliott Wave Principle (EWP) as a major tool for chart analysis and offers forecasting and instructive guidance to its clients. EWP is a popular technical analysis approach: it is based on the belief that market prices have a tendency to move infinitely in a cycle (see Figure 2) in all time frames, exhibiting repetitive wave patterns (Poser, 2003). EWP provides the theoretical foundations for chart analysis and for the automatic completion of chart patterns within a specific timescale. The use of EWP also makes it easier to understand the charts and the captions, as it annotates the waves with

[3] Source: https://www.investopedia.com/terms/e/elliottwavetheory.asp

[4] https://elliottwave-forecast.com

the so-called *degrees* (the alphabetical or numeral indices along the wave patterns in Figure 2).

(a) Article



Short Term Elliott Wave view in DAX suggests the rally from 10.23.2023 low is unfolding as a 5 waves impulse. Up from 10.23.2023 low, wave 1 ended at 17003.28. Wave 2 pullback subdivided into a double three Elliott Wave structure. Down from wave 1, wave ((w)) ended at 16448.71 and wave ((x)) rally ended at 16839.49. Wave ((y)) lower subdivided into a zigzag like the 30 minutes chart below shows. Down from wave ((x)), wave (a) ended at 16542.99 and wave (b) ended at 16752.24. Wave (c) lower ended at 16348.28 which completed wave ((y)) of 2 in higher degree.

Index has resumed higher in wave 3. Up from wave 2, wave (i) ended at 16705.49 and wave (ii) ended at 16607.72. Wave (iii) higher ended at 16967.72 and pullback in wave (iv) ended at 16860.05. Wave (v) higher ended at 16999.58 which completed wave ((i)) in higher degree. Pullback in wave ((ii)) is in progress as a double three Elliott Wave structure. Down from wave ((i)), wave (w) ended at 16825.43 and wave (x) ended at 16915.92. Expect wave (y) to extend lower to reach 16644.48 – 16747.95 area to complete wave (y) of ((ii)). Near term, as far as pivot at 16348.28 low stays intact, expect pullback to find support in 3, 7, 11 swing for further upside.

(b) Textual content

(c) Chart image

Figure 3: A sample article from ElliottWave-Forecast in Fig. 3a, with the image and text displayed separately (EWFHendra, 2024). In the image captioning task, Figure 3c is the image input *I*, and Figure 3b the *gold* caption *C*.

The analysts publish their analysis (in English) together with chart images (see Figure 3a). For our dataset, we downloaded the articles released on the website by February, 2024. Each article was split into images and texts. We removed images that are not target charts, and texts and mark-up that are unnecessary, such as authorship, HTML tags, advertisements, and so on. So far, we only kept the samples containing one single stock chart (some articles may include two or more), to ensure a collection of one-to-one rather than one-to-many image-caption pairs. In addition, samples in which the text body is too long (> 400 words) or too short (<100 words) have been excluded, according to the statistics on text length. The remaining stock chart-text pairs are our final dataset items, for a total of 1972 chart-caption pairs.

3.2 Chart Annotation

As shown in Figure 3c, stock charts map the price on the Y-axis against the time on the X-axis, and they typically come with several annotations. We categorized such annotations in four main types: Degree, Time, Price, and Add-on. *Degree* refers, roughly speaking, to a price movement; *Add-on* includes the information that is additionally applied to the charts, such as reminder messages and titles, while *Time* and *Price* are self-explanatory. We also annotated the endpoints of predictive patterns (categorized as *Point*) for further studies on automatic pattern completion[5]. The taxonomy and descriptions of the annotations are presented in Table 1.

[5]We did not annotate the endpoints for historical patterns, because they can be automatically generated with price data.

Table 1: A taxonomy of charting annotations. Here OHLC is used as a general term for *OHLC*, *Adj.* and *Volume* data. OHLC stands for Opening, Highest, Lowest, and Closing prices of a financial instrument during a timeframe, while *Adj.* is the adjusted closing price accounting for corporate actions, and *Volume* refers to the transaction amount.

Category	Description
Degree	EWP degrees
Time	X-axis ticks, timestamp, time markers
Price	Y-axis ticks, OHLC, price markers
Add-on	Supplementary indicators, annotations, and watermarks, etc.
Point	Endpoints of the predictive patterns

4 Evaluation with LVLMs

4.1 Model Choice

We ran evaluations with five recent general-purpose LVLMs that have showcased impressive capabilities in image captioning and visual question answering tasks (Li et al., 2023b; Zhang et al., 2024), including GPT-4V (OpenAI, 2023), mPLUG-Owl2 (Ye et al., 2023), LLaVA (Liu et al., 2024), InstructBLIP (Dai et al., 2024) and Gemini (Gemini Team Google, 2024). Each LVLM was prompted with the instruction below to produce candidate captions, and evaluated in a zero-shot setting. [6]

- Instruction: *Based on the chart image, generate a text around 100 to 400 words, describing the historical price movements and predictions and concluding the opinion of the chartist towards the stock trends.*

4.2 Evaluation Metrics

4.2.1 Text Similarity Metrics

Regarding evaluation metrics, we have considered the most popular ones, which are mainly based on n-gram overlapping, including BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005) and CIDEr (Vedantam et al., 2015). We also included two semantic-oriented metrics that are based on contextualized

[6]We had a pilot study with a few samples and some candidate instructions. The presented instruction achieved good performance overall. We observed that the few-shot learning setting is not applicable given the nature of the presented task, and thus we did not conduct the few-shot experiments

embeddings: BERTScore (Zhang et al., 2020), and paragraph-level cosine similarity, denoted as COS_F. After a pilot study, we selected the OpenAI embedding model as the best one to measure COS_F (see section A in the Appendix for more details).

4.2.2 Fine-grained Examination

Additionally, observing that the metrics above measure textual similarity in general, we also attempted to have a closer examination of the generated captions. To this purpose, we further analyzed the results in terms of *SA* and *IoU*.

IoU (Intersection over Union). We borrowed this metric from the field of object detection to measure how much important information has been included in the candidates relative to the references. In object detection, it measures the overlap over the union of predicted and ground-truth objects (Everingham et al., 2010). Different from its original formulation, an adaption has been made to fit into the settings of our work. In object detection, not all objects present in an image are annotated as ground truth. Also, a predicted object is only considered correct if it matches a labeled ground truth object.

In contrast, we assume that all metadata items present in the image are potentially valid and relevant for narration. Therefore, a candidate text should be rewarded whenever it correctly mentions an item from the image's full annotation — even if that item is not found in the reference text. Such *extra* information is still accurate and should neither be penalized nor neglected. Meanwhile, we attempt to bridge between the candidate and the reference. For this, we measure how the candidate text covers the objects compared to the reference text, while using the annotations as a *background* that has all possible metadata items. Let O_a be all metadata appeared in the image, and O_c and O_r be the metadata that is mentioned in the candidate and reference texts respectively. For each candidate-reference text pair (c, r), we first count how many objects they mention that also appear in O_a:

$$Hit_c = |O_c \cap O_a|, Hit_r = |O_r \cap O_a|$$

Then the relative coverage of c to r can be a ratio:

$$\frac{Hit_c}{|O_a|} / \frac{Hit_r}{|O_a|}$$

To further account for the impact of text length and large values, we redefine the IoU formula as:

formulate IoU as:

$$IoU = \log\left(1 + \frac{\frac{|O_c \cap O_a|}{L_c}}{\frac{|O_r \cap O_a|}{L_r} + \epsilon}\right), \tag{1}$$

where L_c and Lr is the text length of c and r respectively, ϵ is a constant value to prevent division by zero and a logarithm could smooth the results.[7]

The dataset contains three categories of metadata i.e., the *metadata*, including *Degree*, *Time* and *Price*. For each sample, the IoU score is an aggregated result across these three categories. The IoU value would always be non-negative. A higher value may suggest that a more fine-grained description has been given, with more important information or meta information has been referred to. Conversely, lower scores may indicate under-description, omission of key elements.

SA (Sentiment Alignment). The reference text states the historical movement of price, then outlines a prediction for future trends, indicating the existence of opinions, i.e., sentiment. An appropriate candidate is expected to express a similar sentiment in general. Accordingly, sentiment analysis can be performed on the reference and candidate texts to evaluate whether their sentiments were aligned. The assessment of sentiment alignment is presented as a text classification problem on three polarity categories, including *positive*, *negative* and *neutral*. The SA score of each reference-candidate pair is formulated as below:

$$\mathrm{SA}(x, \hat{x}) = \begin{cases} 1, & \text{if } x = \hat{x} \\ 0, & \text{otherwise} \end{cases},$$

where x and $\hat{x}$ denote the reference and candidate text, and y and $\hat{y}$ denote their respective sentiments, which can be obtained using a BERT model fine-tuned for financial sentiment analysis [8], 1 indicates a correct sentiment alignment between x and $\hat{x}$ and 0 a mismatch.

Importantly, we don't perform sentiment analysis over the entire text. What is valued the most is the portion that contains opinions towards the future, which are typically located in the final part of the text. Therefore, our analysis only focus on the last a few sentences of a text. Given that this is an open-ended generation task and sentence-level alignment between reference and candidate

[7] ϵ takes the value of 1×10^{-6} during calculation.

[8] https://huggingface.co/ahmedrachid/FinancialBERT-Sentiment-Analysis

texts seems infeasible, we measure aggregated sentiments: The sentiment of each text segment is computed from the predicted labels of its individual sentences through a weighted voting approach, in which sentences closer to the end are assigned higher weights.

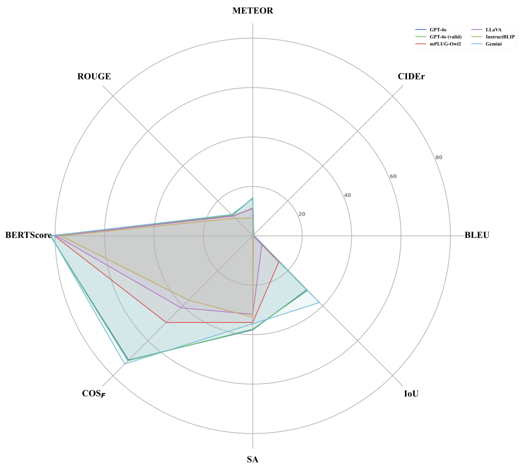


Figure 4: Radar chart of all evaluation scores. All values are presented on a 100% scale.

5 Results and Discussion

Table 2 presents the zero-shot evaluation results.

The instant observation is that all the LVLMs, despite their excellence in image understanding tasks, have achieved very low values for the evaluation metrics based on n-gram overlap (roughly the upper part of Table 2). The radar chart in Figure 4 presents the extremely unbalanced distribution between these common metrics and those introduced by us. Standard evaluation metrics for image captioning tasks — BLEU, ROUGE, CIDE_r and METEOR — consistently report low scores for all models, even for the top ones like Gemini and GPT-4V. This is actually not unexpected: given the domain specificity and the length of the reference texts, it would have been surprising - and possibly an index of data contamination - to observe high values for such metrics.

For SA and IoU, GPT-4V and Gemini outperform the other models. It should be noticed that Gemini scores are much better on the IoU metric, meaning that its responses are more likely to contain more of the correct chart metadata, and has better coverage. In general, LVLMs achieve higher scores in BERTScore and COS_F, showing that even if they do not use the same words (i.e. low n-gram overlap), the texts generated by some of the models have high conceptual similarity with the reference. On the other hand, the SA scores, reflecting the alignment between model outputs and opinions of the analysts, have only around 40% in the best models. Higher values would be desirable, as this metric might be the most relevant in affecting the stock buying decisions of the investors and we would like LVLMs to be as much aligned as possible with human experts' insights, in order for them to be deployed in real-world applications.

Examples are provided in Table 6 (in the Appendix). From them, we can find that the reference text and model output appear to differ in narration. In this sense, the low scores are quite understandable, as these metrics in nature rely on lexical overlapping. While these methods may perform well on regular datasets, where texts are in short forms and show limited possibilities for paraphrasing, their effectiveness can be predictably compromised in this context which presents a open-ended long text generation task. The extremely low values of BLEU or CIDEr suggest they could be inappropriate for the evaluation of this task. Nonetheless, despite their variance in wording and constructions, some model outputs — such as those from Gemini and GPT-4V as listed — remain faithful to the given chart and provide meaningful interpretations for readers. Therefore, relying solely on traditional metrics would seem unfair. In this sense, the use of alternative metrics — BERTScore and COS_F seem inevitable. BERTScore, according to its formulation (please refer to Zhang et al. (2020) for details), measures lexical similarity using contextualized embeddings from Transformers, which is more flexible and effective compared to its precedents. COS_F, on the other hand, encodes the entire text as vectors and then measures semantic similarity in a more blunt manner.

Additionally, we observe that the reference texts which are produced by professional analysts not only uncover the future price trends, but also try to disclose the full picture of the movement, i.e., to provide detailed context and justification for their description or analysis, making it more reliable. During the process, EWP markers (i.e., the numerical and alphabetical annotations around the waves) are frequently referred to, which in some way enhances the accessibility of the text. In contrast, the performance of LLMs varies significantly. Strong models like Gemini and GPT-4V appear to be aware of the EWP theory and able to recognize the visual elements such as markers and labels on

Table 2: A summary of the evaluation results on the dataset (1972 instances). Several images are rejected by GPT-4V, due to sensitivity concerns or other reasons such as image quality, according to the feedback from the model. The *Valid* column contains scores for valid captions only (1915 instances). Out of 1972 GPT-4V responses, 57 are found invalid. 4-gram scores are reported for BLEU and CIDEr, and F-score for BERTScore and for SA. The best scores are indicated in **bold** among all samples or underlined among valid samples. For all metrics, scores are reported in percentage (%) and rounded half up to two decimal precision and higher values indicate better performance.

	GPT-4V		mPLUG-Owl2	LLaVA	InstructBLIP	Gemini
	Overall	Valid				
BLEU	.65	.66	.44	.49	.25	**0.73**
CIDEr	.58	.60	.26	.24	.05	**0.62**
METEOR	**15.32**	15.36	11.05	11.24	7.14	15.05
ROUGE$_L$	11.74	11.78	11.39	11.28	10.07	**12.20**
BERTScore	**81.91**	81.94	80.22	79.51	77.63	81.71
COS$_F$	71.10	71.46	49.56	41.16	36.77	**73.32**
SA	**38.04**	37.74	35.04	31.73	32.89	35.58
IoU	30.88	31.31	14.94	5.19	.43	**38.15**

the chart. In contrast, weak models could even fail in recognition. For example, the red-highlighted sentence in Table 6 indicates that the LLAVA model could not recognize the time frame of the chart. From its generated textual description, nor can we find traces that the model understands the EWP theory or master the skills of giving suggestions for investment. Text-similarity metrics such as BERTScore, ROUGE, may fail to capture these subtle distinction in their measurement, and that is the reason that IoU is introduced in the evaluation.

Table 3: A breakdown of the IoU scores. The reported values (in %) represent the metadata coverage within different categories. For instance, suppose n Time markers are included in the full annotation and m found in the reference text, the *Time* coverage of the reference is then calculated as $\frac{m}{n}$.

	Degree	Time	Price
Reference	26.83	20.09	4.65
GPT-4V	3.72	7.07	5.83
GPT-4V (valid)	3.69	7.16	5.86
mPLUG-Owl2	.26	4.79	.22
LLaVA	.01	1.58	.05
InstructBLIP	.00[9]	.13	.01
Gemini	5.47	12.01	5.79

Besides from the overall scores presented in Table 2, Table 3 provides more statistics around IoU. These objective and quantitative results support our observation that human chartists tend to favor including metadata especially wave degrees, while LLMs are less attentive in their generation.

Regarding SA, we reported its F1-scores in Table 2. The confusion matrices of sentiment alignment are revealed in Figure 6 in the Appendix. A majority of errors occur between the *neutral* class and the others. Although the sentiment analysis model could be to blame, we noticed that subtle mismatches between the narrative tones could have contributed to the low scores. For instance, as shown in Table 5 (in Appendix), the reference text expresses a positive outlook towards the stock, while Gemini, although agrees, adopts a more cautious and restrained tone.

Based on these findings, it is evident that the some LLMs could have posed the capability of stock chart interpretation and demonstrate a certain level of financial domain knowledge, even in cases where their overall performance is not satisfying.

In the future, we believe that the performance can be further improved via a more systematic search for optimal prompt instructions, and possibly by introducing customized architectures for the task.

6 Conclusion

In this work, we have introduced StockGenChaR, a dataset for stock chart captioning, and we have reported the preliminary work on the captioning task, including the creation of the benchmark and a preliminary evaluation with some popular LVLMs. By first exploring these LVLMs, we hope to find out their capacity in this stock chart captioning

[9]The actual value is above zero, but it is displayed as zero due to standard rounding.

without additional fine-tuning, and also to identify proper evaluation methods for the task.

The current findings suggest that these LVLMs could have possessed limited capabilities of stock chart captioning. However, for practical deployment in financial scenarios, task-specific fine-tuning is still required. Also, given the sensitive nature of financial data, directly using open-sourced models may pose risks related to confidentiality or information breach. Also, despite our efforts on evaluation metrics, automatic evaluation metrics alone may be insufficient to capture all aspects of the text quality, especially in our setting, where the task involves generating long-form texts intended for human readers, including amateurs to finance and stocking. In this sense, further explorations with customized approaches and the inclusion of human evaluation seem necessary. On the bright side, we have seen that these LVLMs, close-sourced or open-sourced, are equipped with the visual recognition and textual generation ability. It is therefore safe to assume that the performance on the presented task can be further improved via a more systematic search for optimal prompt instructions, and possibly by introducing customized architectures.

Limitations

Our work has some limitations that have to be acknowledged. First, the paper presents only preliminary evaluation results with general-purpose LVLMs. For future work, we plan to experiment with more customized LVLM architectures (e.g., those mentioned in Section 2) to further push the boundaries of model performance on the proposed chart captioning task.

Also, the present work only adopts automatic metrics for evaluation. From Table 2, we can find that metrics that are based on word overlapping such as BLEU and CIDEr, are insufficient in evaluating long texts; the two embedding-based methods, BERTScore and COS_F, despite capturing semantic similarity, demand further examination because they could fail to measure the user *accessibility* of the generated texts, as the task requires. In the future, we plan to explore more sophisticated prompting strategies to explicitly target accessibility and other *desiderata* aspects of the generated captions, and to include human annotators for evaluating the generated texts.

Acknowledgements

The authors acknowledge the support from the project "Analyzing the semantics of Transformers representations for financial natural language processing" (ZVYU), funded by the Faculty of Humanities of the Hong Kong Polytechnic University (PolyU-UGC). We would also like to thank our reviewers for the constructive feedback.

Ethical Considerations

The models and data used in this study are intended for research purposes only. No personally identifiable information or sensitive content is involved. Therefore, we believe the ethical risk of this work is minimal.

We utilized several models in this study, each governed by its individual license — LLaVA is released under the Apache License 2.0, mPLUG-Owl2 under the MIT License, and InstructBLIP under the CC BY-NC 4.0 License. For other models and the data we collected without publicly specified licenses, we used them in accordance with the terms of service or usage guidelines by their original provider, where available. Additionally, ChatGPT was employed as a writing assistant under API terms for translation and grammar checking purposes.

The annotators were recruited online from Mainland China. All had undergraduate or master's degrees in computer-related disciplines and were employed in relevant industries. Only individuals who passed a qualification test were selected to participate in the annotation task. All annotators provided informed consent, received compensation on a per-annotation basis in accordance with local labor standards, and retained the right to withdraw from the study at any time.

This study has been approved by the Institutional Review Board (IRB) from the Department of Language Science and Technology of the Hong Kong Polytechnic University.

Acknowledgement of Data Usage

The data used in this research was obtained from Elliott Wave Forecast with the necessary permission for usage. The provider has explicitly granted consent, ensuring compliance with relevant legal, ethical, and regulatory requirements. We affirm that the data will be handled responsibly and utilized strictly within the agreed scope.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. 2019. Nocaps: Novel Object Captioning at Scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Raffaella Bernardi, Ruket Cakici, Desmond Elliott, Aykut Erdem, Erkut Erdem, Nazli Ikizler-Cinbis, Frank Keller, Adrian Muscat, and Barbara Plank. 2016. Automatic Description Generation from Images: A Survey of Models, Datasets, and Evaluation Measures. *Journal of Artificial Intelligence Research*, 55:409–442.

Gagan Bhatia, El Moatez Billah Nagoudi, Hasan Cavusoglu, and Muhammad Abdul-Mageed. 2024. FinTral: A Family of GPT-4 Level Multimodal Financial Large Language Models. *arXiv preprint arXiv:2402.10986*.

Ali Furkan Biten, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. 2019. Good News, Everyone! Context Driven Entity-aware Captioning for News Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Fei-Long Chen, Du-Zhen Zhang, Ming-Lun Han, Xiu-Yi Chen, Jing Shi, Shuang Xu, and Bo Xu. 2023. VLP: A Survey on Vision-language Pre-training. *Machine Intelligence Research*, 20(1):38–56.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. 2024. Instructblip: Towards General-purpose Vision-language Models with Instruction Tuning. *Advances in Neural Information Processing Systems*, 36.

Robert D Edwards, John Magee, and WH Charles Bassetti. 2018. *Technical Analysis of Stock Trends*. CRC Press.

Elliott Waves Academy. 2024. Future Outlook for the Digital Currency ARBUSD on the Daily Time Frame. Online; accessed on 13 June, 2024 from Twitter.

ElliottWave-Forecast. 2024. About Us. `https://elliottwave-forecast.com/about-us/`. Accessed: 04-Feb-2024.

Kawin Ethayarajh. 2019. How Contextual Are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of EMNLP*.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88:303–338.

EWFHendra. 2024. DAX Near Term Support Area. `https://elliottwave-forecast.com/news/dax-near-term-support-area/`. Accessed: 05-Feb-2024.

Zhaoxin Feng, Jianfei Ma, Emmanuele Chersoni, Xiaojing Zhao, and Xiaoyi Bao. 2025. Learning to Look at the Other Side: A Semantic Probing Study of Word Embeddings in LLMs with Enabled Bidirectional Attention. In *Proceedings of ACL*.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating Attractive Visual Captions with Styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, and 1 others. 2022. Vision-language Pre-training: Basics, Recent Advances, and Future Trends. *Foundations and Trends in Computer Graphics and Vision*, 14(3–4):163–352.

Gemini Team Google. 2024. Gemini: A Family of Highly Capable Multimodal Models. *arXiv preprint arXiv:2023arXiv231211805G*.

Michael Grubinger, Paul Clough, Henning Müller, and Thomas Deselaers. 2006. The IAPR TC-12 Benchmark: A New Evaluation Resource for Visual Information Systems. In *International Workshop OntoImage*, volume 2.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. 2020. Captioning Images Taken by People Who Are Blind. In *Proceeding of the European Conference of Computer Vision (ECCV)*.

Trang-Thi Ho and Yennun Huang. 2021. Stock Price Movement Prediction Using Sentiment Analysis and CandleStick Chart Representation. *Sensors*, 21(23):7957.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research*, 47:853–899.

Guosheng Hu, Yuxin Hu, Kai Yang, Zehao Yu, Flood Sung, Zhihong Zhang, Fei Xie, Jianguo Liu, Neil Robertson, Timpathy Hospedales, and 1 others. 2018. Deep Stock Representation Learning: From Candlestick Charts to Investment Decisions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2706–2710. IEEE.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, and 1 others. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 123:32–73.

Rosdyana Mangir Irawan Kusuma, Trang-Thi Ho, Wei-Chun Kao, Yu-Yen Ou, and Kai-Lung Hua. 2019. Using Deep Learning Neural Networks and Candlestick Chart Representation to Predict Stock Market. *arXiv preprint arXiv:1903.12258*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of EMNLP*.

Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. 2019. Entangled Transformer for Image Captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8928–8937.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping Language-image Pre-Training with Frozen Image Encoders and Large Language Models. In *Proceedings of the International Conference on Machine Learning*, pages 19730–19742. PMLR.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, and 1 others. 2020b. Oscar: Object-semantics Aligned Pre-training for Vision-language Tasks. In *Computer Vision–ECCV*, pages 121–137. Springer.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating Object Hallucination in Large Vision-language Models. *arXiv preprint arXiv:2305.10355*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in COntext. In *Proceedings of Computer Vision (ECCV)*, pages 740–755. Springer.

Fuxiao Liu, Xiaoyang Wang, Wenlin Yao, Jianshu Chen, Kaiqiang Song, Sangwoo Cho, Yaser Yacoob, and Dong Yu. 2023. MMC: Advancing Multimodal Chart Understanding with Large-scale Instruction Tuning. *arXiv preprint arXiv:2311.10774*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual Instruction Tuning. *Advances in Neural Information Processing Systems*, 36.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated Flower Classification over a Large Number of Classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE.

Witawat Norasaed and Thitirat Siriborvornratanakul. 2024. Market Movement Prediction Using Chart Patterns and Attention Mechanism. *Discover Analytics*, 2(1):1.

OpenAI. 2023. GPT-4V(ision) System Card. *Published on: openai.com*.

Vicente Ordonez, Girish Kulkarni, and Tamara Berg. 2011. Im2text: Describing Images Using 1 Million Captioned Photographs. *Advances in Meural Information Processing Systems*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*.

Ed Ponsi. 2016. *Technical Analysis and Chart Interpretations: A Comprehensive Guide to Understanding Established Trading Tactics for Ultimate Profit*. John Wiley & Sons.

Steven W Poser. 2003. *Applying Elliot Wave Theory Profitably*, volume 169. John Wiley & Sons.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763. PMLR.

Arnau Ramisa, Fei Yan, Francesc Moreno-Noguer, and Krystian Mikolajczyk. 2017. Breakingnews: Article Annotation by Image and Text Processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1072–1085.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting Image Annotations Using Amazon's Mechanical Turk. In *Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset for Automatic Image Captioning. In *Proceedings of ACL*.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. Textcaps: A Dataset for Image Captioning with Reading Comprehension. In *Proceedings of the European Conference of Computer Vision (ECCV)*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based Image Description Evaluation. In *Proceedings of the IEEE cConference on Computer Vision and Pattern Recognition*.

Marc Velay and Fabrice Daniel. 2018. Stock Chart Pattern Recognition with Deep Learning. *arXiv preprint arXiv:1808.00418*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD Birds 200. *California Institute of Technology, Technical Report*.

Qianqian Xie, Dong Li, Mengxi Xiao, Zihao Jiang, Ruoyu Xiang, Xiao Zhang, Zhengyu Chen, Yueru He, Weiguang Han, Yuzhe Yang, and 1 others. 2024. Open-FinLLMs: Open Multimodal Large Language Models for Financial Applications. *arXiv preprint arXiv:2408.11878*.

Xuewen Yang, Heming Zhang, Di Jin, Yingru Liu, Chi-Hao Wu, Jianchao Tan, Dongliang Xie, Jue Wang, and Xin Wang. 2020. Fashion Captioning: Towards Generating Accurate Descriptions with Semantic Rewards. In *Proceedings of the European Conference of Computer Vision (ECCV)*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. mplug-owl2: Revolutionizing Multi-modal Large Language Model with Modality Collaboration. *arXiv preprint arXiv:2311.04257*.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From Image Descriptions to Visual Denotations: New Similarity Metrics for Semantic Inference over Event Descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. 2024. MM-LLMs: Recent Advances in Multimodal Large Language Models. *arXiv preprint arXiv:2401.13601*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *Proceedings of ICLR*.

Yuechu Zheng, Yain-Whar Si, and Raymond Wong. 2021. Feature Extraction for Chart Pattern Classification in Financial Time Series. *Knowledge and Information Systems*, 63(7):1807–1848.

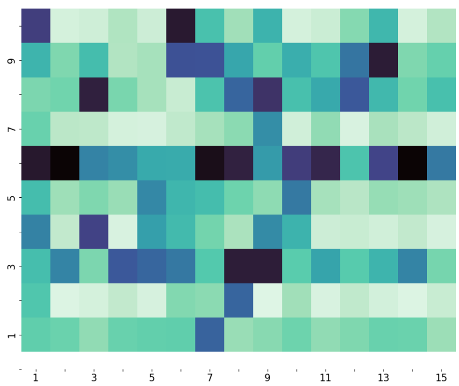
A Embedding Model Selection

Following Li et al. (2020a), we designed a similarity experiment to find the ideal embedding model:

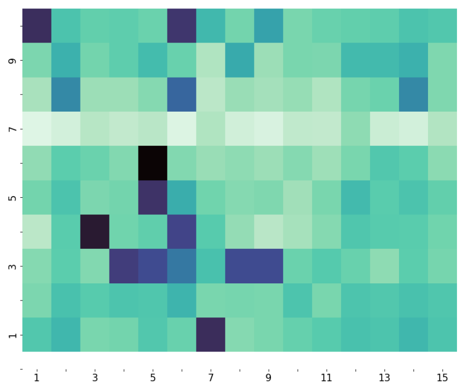
- First, we collected generated texts from GPT-4V using randomly-picked samples and designed instructions, and then created gold label similarities between pairs of gold truth captions and generated texts (The gold label similarity is actually a pseudo similarity. Currently, we haven't conducted a standardized human evaluation of the generated texts. But GPT-4V sometimes produces invalid responses due to sensitivity concerns or other reasons, so we take the extreme values by assigning a similarity score of 0 between the caption and invalid response pair, and a score of 1 for other cases.);
- We obtained text embeddings from each candidate embedding model;
- We computed COS_F scores between each paired embeddings as the model-wise predicted similarities, and then calculate the Spearman's correlation coefficients between them. We experimented with BERT, RoBERTA and the OpenAI embedding models, and reported the COS_F and Spearman scores in Figure 5. An ideal model is supposed to exhibit strong correlation with the gold labels. Additionally, it should be able to differentiate between different instructions, meaning that the predicted similarities should vary upon the provided instructions. Based on these criteria, the OpenAI embedding model outperformed the others. Besides, considering its larger token window (1536 tokens), we decided to select the OpenAI embedding model as the embedding source for the COS_F metric.

B Supplementary Tables and Figures

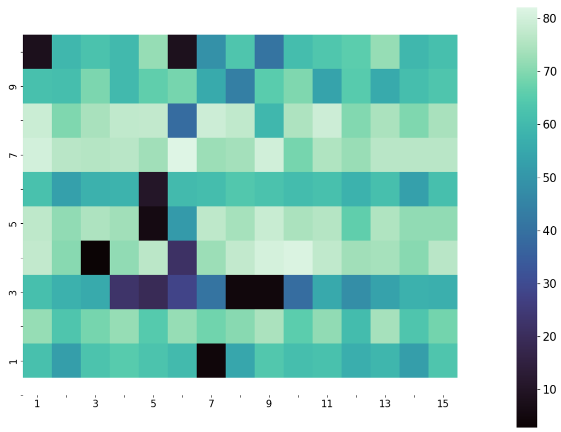
[10] The number of captions per image.



(a) BERT
r_s. = 27.48
Avg. = 90.97



(b) RoBERTa
r_s. = 44.82
Avg. = 99.74

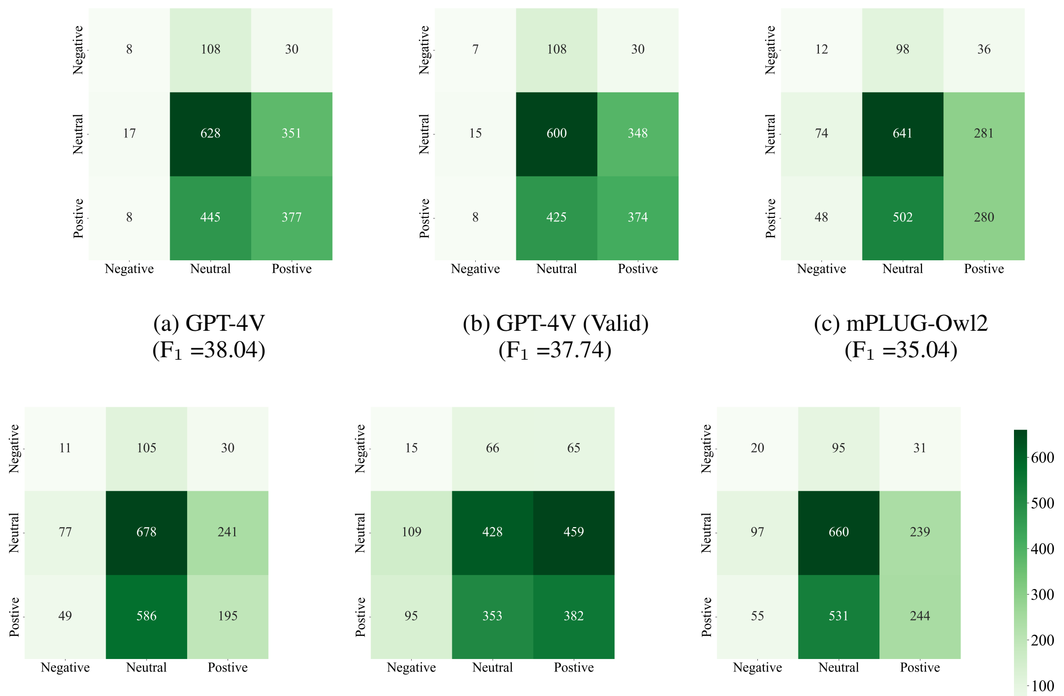


(c) OpenAI model
r_s. = 47.23
Avg. = 61.79

Figure 5: COS$_F$ scores are presented in heatmaps, where the X-labels indicate different instructions and Y-labels different samples. The Spearman's correlation coefficients are denoted as r_s., with p-value $< 0.1\%$ in all cases, while Avg. indicates the average COS$_F$ score (notice that the OpenAI model is the one with the highest correlation while having at the same time a lower average cosine score. The other two models seem to have a high level of anisotropy of the vector space, cf. Ethayarajh (2019); Feng et al. (2025), assigning similarities close to 1 to most text pairs). Regarding model versions, we chose *bert-base-uncased* for BERT, *rflike berta-base* for RoBERTa, and *text-embedding-3-small* for the OpenAI model. All scores are reported in percentage.

Dataset	Domain	Total	Caps.[10]	Source
Conceptual Caption (Sharma et al., 2018)	Generic	3.3M	5	Web
MS COCO (Lin et al., 2014)	Generic	328K	5	Web
Flickr30k (Young et al., 2014)	Generic (people, animals)	31K	5	Flickr.com
Flick 8K (Hodosh et al., 2013)	Generic	8K	1-5	Flickr.com
FlickrStyle10K (Gan et al., 2017)	Generic	10K	2	Flickr.com
SBU Captions (Ordonez et al., 2011)	Generic	1M	5	Web
Visual Genome (Krishna et al., 2017)	Generic	108K		Web
VizWiz Captions (Gurari et al., 2020)	Assistive	39K	5	VizWiz APP
CUB-200 (Welinder et al., 2010)	Birds	12K	10	Web
Oxford-102 (Nilsback and Zisserman, 2008)	Flowers	8K	10	Web
Fashion Captions (Yang et al., 2020)	Fashion	52K	5	Web
BreakingNews (Ramisa et al., 2017)	News(sports, arts, etc.)	100K	5	Web
GoodNews (Biten et al., 2019)	News	466K	1	New York Times
SentiCap (Mathew et al., 2022)	Generic	3.2K	6	MS COCO
TextCaps (Sidorov et al., 2020)	OCR	28.4K	5-6	Web
nocaps (Agrawal et al., 2019)	Generic	15.1K	11	Web
IAPR TC-12 (Grubinger et al., 2006)	Generic	20K	1-5	Viventura
PASCAL 1K (Rashtchian et al., 2010)	Generic (people, animals)	1K	5	PASCAL VOC

Table 4: A general summary of the benchmark datasets for image captioning



(d) LLaVA
(F_1 =31.73)

(e) InstructBLIP
(F_1 =32.89)

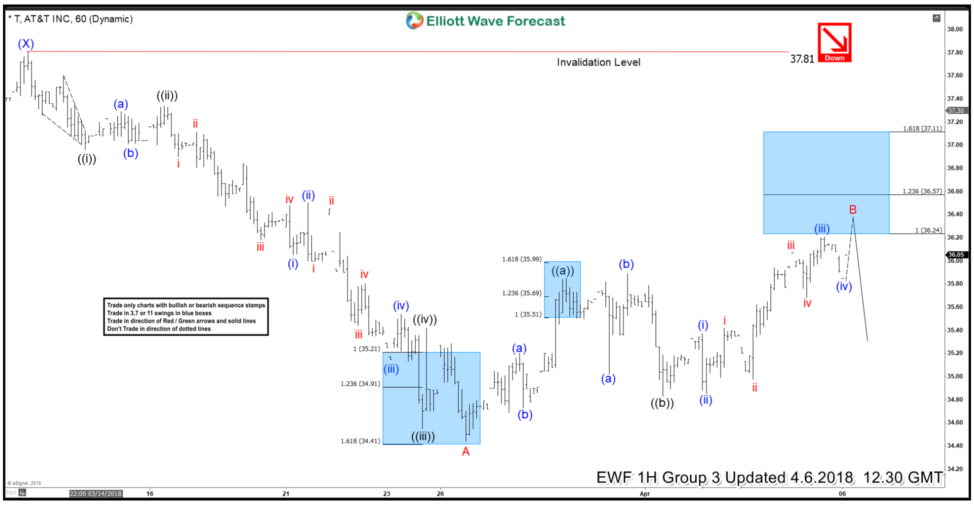
(f) Gemini
(F_1 =35.58)

Figure 6: Confusion matrices of SA. Macro F_1 scores of the models are presented in the brackets, reported on a 100% scale.

Table 5: An example of the false neutral case from Gemini (i.e., a positive sample is classified as a neutral). Only the last a few sentences are presented.

Reference: "... The internal structure of wave Z is unfolding as a double three structure where wave (w) of Z ended at \$53.79 and wave (x) of Z has ended at \$52.08. Near term, the rally from wave (x) low ended at \$54.37 as 5 waves, and CL_F is expected to do a wave b pullback to correct the rally from \$52.08 low in 3, 7, or 11 swing before the rally resumes towards \$55.93 – \$56.83. We don't like selling the proposed pullback and expect buyers to appear again once wave b pullback is complete in 3, 7, or 11 swing as far as pivot at \$52.07 remains intact in the first degree. If pivot at \$52.07 fails, then CL_F has ended cycle from \$49.95 low and will do a larger correction and still expected to turn higher while second degree pivot at \$49.95 stays intact."

Gemini: ... The identified wave structure combined with critical support levels provides a framework for possible upward momentum. Traders and investors are advised to watch for a rebound from the defined support levels, which could signify a continuation of the bullish trend established earlier in the cycle. This analysis reinforces a strategic approach, encouraging trades in alignment with the identified Elliott wave patterns and Fibonacci levels for optimal risk management.



Reference:	AT&T ticker symbol: ($T) short-term AT&T Elliott Wave view suggests that the bounce to 313 high (37.81) ended Intermediate wave (X) bounce. Below from there, intermediate Y leg lower remains in progress as a Zigzag correction. Where initial decline unfolded in 5 waves Elliott Wave Impulse Sequence with extension in Minor wave A. Which can be seen in the Chart below, with the internal distribution of each leg consisted of 5 waves structure thus it favored being impulse. Now let's take a quick look at the short-term price action in the stock from 313 peak (37.81). The decline from there can be seen in 5 waves impulse sequence, where internals of each leg is consist of 5 waves structure. Where Minute wave ((i)) ended at 36.97, Minute wave ((ii)) ended at 37.34 and Minute wave ((iii)) ended at 34.55. Then from there, Minute wave ((iv)) ended at 35.29 and Minute wave ((v)) of A ended at 34.44 low. Up from there, the stock is correcting the 313 cycle (37.81) in Minor wave B bounce. And expected to find seller's in 3, 7 or 11 swings as far as the pivot from 313 peak (37.81) stays intact. So far the bounce looks to be in 3 swings as a Flat structure, where Minute wave ((a)) ended at 35.89, Minute wave ((b)) ended at 34.84 and Minute wave ((c)) of B remains in progress towards 36.24-36.57 100%-123.6% Fibonacci extension area of ((a))-((b)). Before finding seller's again for another extension lower ideally provided the pivot from 37.81 high remains intact. We don't like buying it and intraday traders can look to sell it from an above-mentioned 100%-123.6% ext area for 3 wave reaction lower at least. Keep in mind that the market is dynamic and the FTSE Elliott Wave view could change in the meantime...
Gemini:	This Elliott Wave Forecast chart for AT&T INC (60-minute timeframe) depicts historical price movements and projects future trends using Elliott Wave principles. The chart shows a complex pattern, marked by distinct impulsive and corrective waves labeled with Roman and lowercase letters respectively. The analysis identifies two completed corrective patterns labeled 'A' and 'B'. 'A' shows a corrective three-wave structure, followed by an impulsive five-wave structure labeled 'B'. The 'B' wave appears to have reached a potential termination point, suggesting a significant price reversal is imminent. A dashed line indicates a potential invalidation level at approximately 37.81. If the price breaks above this level, the bearish projection would be invalidated. The chart's creator has drawn blue boxes around specific wave structures, suggesting these areas represent significant price swings suitable for trading. Red and green arrows indicate the predicted direction of price movement; dotted lines, however, should be avoided for trading signals. The analyst recommends trading only charts displaying clear bullish or bearish sequence stamps and focusing on 3, 7, or 11 swings within the blue box areas. The chartist's overall opinion is bearish. The structure of wave 'B' and its potential termination point suggest that a downward trend is likely to follow. The invalidation level at 37.81 provides a clear benchmark for assessing the validity of this bearish prediction. A break above this level would signal a significant shift in the underlying price trend and invalidate the current bearish outlook. The analyst emphasizes the importance of adhering to the specified trading rules, particularly avoiding trades counter to the direction of the solid and directional arrows.
GPT-4V:	The chart displaying the historical price movements of AT&T Inc. (Ticker: T) utilizes the Elliott Wave theory to forecast future trends based on observed patterns in the price movements. The analysis reveals the stock has experienced significant volatility, characterized by both upward and downward movements delineated by specific wave formations. Initially, the stock faced a bearish trend, denoted by the downward movements labeled as waves (i), (ii), and (iii). This initial decline appears to have established a low point around 34.41, suggesting potential support levels. Subsequently, the price exhibited a corrective wave structure wherein it managed to regain some value, as indicated by the formations tagged as ((iii)), ((iv)), and subsequent retracements represented by (a) and (b). The chart depicts key Fibonacci retracement levels, illustrating potential areas of resistance and support. For instance, the (1) level at 35.51 indicates a notable area where the price experienced fluctuations, while the 1.236 and 1.618 levels offer insights into possible price targets for the next wave movements. The areas shaded in blue emphasize critical zones for potential trading decisions, guiding market participants on optimal entry and exit points. Moving forward, the chartist's...
LLAVA:	The chart image displays a historical price movement of a stock, with a red line indicating a downward trend. The stock has been experiencing a decline since the beginning of the year, with a notable drop in January. The chartist's opinion on the stock trends is that the stock is likely to continue its downward trajectory, as the current trend is not showing any signs of reversal. The chart also features a forecast for the stock's future performance, with a red line indicating a potential decline. The chartist believes that the stock's current downward trend is likely to continue, and the forecast suggests that the stock may reach a lower low in the near future. The chartist's conclusion is that the stock is not a good investment at the moment, as the current trend is not favorable for long-term investors. The stock's historical price movements and the forecast indicate that it may continue to decline, making it a risky investment option.

Table 6: We selected a chart image and presented it with its reference description and several model output for comparison. The chart presents a 1-hour short term movement of AT&T ticker.

FinEval-KR: A Financial Domain Evaluation Framework for Large Language Models' Knowledge and Reasoning

Shaoyu Dou*
Ant Group

Yutian Shen*†, Mofan Chen*†
Zixuan Wang† and Jiajie Xu†
Shanghai University of Finance and Economics

Qi Guo, Kailai Shao
Chao Chen and Haixiang Hu
Ant Group

Haibo Shi
Min Min and Liwen Zhang‡
Shanghai University of Finance and Economics

Abstract

Large Language Models (LLMs) demonstrate significant potential but face challenges in complex financial reasoning tasks requiring both domain knowledge and sophisticated reasoning. Current evaluation benchmarks often fall short by not decoupling these capabilities indicators from single task performance and lack root cause analysis for task failure. To address this, we introduce FinEval-KR, a novel evaluation framework for decoupling and quantifying LLMs' knowledge and reasoning abilities independently, proposing distinct knowledge score and reasoning score metrics. Inspired by cognitive science, we further propose a cognitive score based on Bloom's taxonomy to analyze capabilities in reasoning tasks across different cognitive levels. We also release a new open-source **Chinese** financial reasoning dataset covering 22 subfields to support reproducible research and further advancements in financial reasoning. Our experimental results reveal that LLM reasoning ability and higher-order cognitive ability are the core factors influencing reasoning accuracy. We also specifically find that even top models still face a bottleneck with knowledge application. Furthermore, our analysis shows that specialized financial LLMs generally lag behind the top general large models across multiple metrics.

1 Introduction

In recent years, rapid LLM development has led the transformation of artificial intelligence. LLMs demonstrate strong natural language processing capabilities and inspire application innovations across various fields, including scientific research (Zhang et al., 2024), financial services (Nie et al., 2024b), content creation (Betker et al., 2023), etc. However, achieving satisfactory performance for complex tasks, such as financial decision making, proves difficult when relying solely on knowledge introduced during training or instructions (Wang and Brorsson, 2025; Liu et al., 2025; Wang et al., 2024). Reasoning ability, that is, the ability of logical deduction, problem solving, and abstract thought, stands as a core hallmark of advanced intelligence and becomes crucial for assessing LLM intelligibility and applicability (Wang and Song, 2024; Li et al., 2024b; Valmeekam et al., 2023). Therefore, it is critical to accurately assess the knowledge and reasoning abilities of LLMs. This helps to understand the shortcomings of the model to support targeted optimization.

Although several LLM evaluation benchmarks have been proposed, they still have several limitations in evaluating reasoning capabilities.

Insufficient capability decoupling. Mainstream benchmarks typically evaluate a model's capabilities based on its performance across various tasks. Their performance represents either the model's knowledge capacity (Nie et al., 2024a; Liu and Jin, 2024) or its reasoning ability (Saparov and He, 2023; Geva et al., 2021). However, our experiments demonstrate that LLM performance in reasoning tasks is influenced by both knowledge and reasoning ability. Therefore, it is essential to decouple and quantify them separately to achieve a more accurate capability characterization.

Lack of root cause analysis. Current reasoning evaluation frameworks mostly focus on the correctness of the reasoning processes and results, while neglecting the diagnosis of the erroneous result. Specifically, there is not yet an effective way to distinguish whether a reasoning failure stems from knowledge gaps, such as unclear concept comprehension, or from flaws in the reasoning processes, like missing reasoning steps. This limits the potential for specifically optimizing the model.

Neglect of the cognitive science perspective in financial LLM benchmark. While some benchmarks

*Equal contribution.
†This work was completed during the internship at Ant Group.
‡Corresponding author.

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

have begun to assess reasoning ability, they generally lack a design grounded in cognitive science–a critical omission for the financial domain. Unlike the deductive reasoning of law or the diagnostic processes of medicine, the essence of financial decision-making is a *quantitative game against future uncertainty*. This requires a spectrum of higher-order cognitive abilities that extend far beyond simple knowledge application (Zhang et al., 2025). For instance, evaluating the impact of a central bank's interest rate policy requires not only *applying* knowledge of rate-to-exchange dynamics but also *evaluating* the complex interplay between market sentiment and strategic policy expectations. This distinction underscores the need for a framework like Bloom's Taxonomy as an essential diagnostic tool to pinpoint the specific cognitive deficiencies that hinder advanced financial reasoning in LLMs.

The above limitations motivate us to explore the following questions:

Q1: How do knowledge and reasoning ability jointly determine the performance of LLMs in domain reasoning tasks?

Q2: Can we develop an evaluation framework that decouples and independently quantifies knowledge and reasoning ability from task performance?

We begin by empirically verifying the fundamental role of knowledge in domain-specific reasoning through a preliminary experiment in the finance sector. Building on these initial findings, we make the following contributions:

- **A novel evaluation framework**. We propose a novel LLM evaluation framework that disentangles the assessment of domain knowledge and reasoning ability from task performance metrics. This allows us to introduce two distinct metrics: a *Knowledge Score* and a *Reasoning Score*. Furthermore, inspired by cognitive science, we posit that complex reasoning relies on a hierarchy of cognitive abilities. This motivates our third metric, the *Cognitive Score*, which leverages Bloom's Taxonomy to provide a fine-grained analysis of the cognitive processes employed by LLMs during reasoning.

- **A new open-source Chinese-language dataset for financial reasoning**[1]. The released dataset encompasses 22 key financial subfields, its primary contribution lies in its multi-layered annotations, where each sample includes knowledge point labels, step-by-step reasoning chains, and the required cognitive skills. As a comprehensive and deeply annotated resource, it is designed to serve as a specialized benchmark to advance research in both financial and cognitive reasoning.

- **Evaluation results of mainstream LLMs**. Based on the proposed framework and dataset, we conduct a comprehensive evaluation of the current mainstream LLMs, which verifies the effectiveness of the proposed evaluation methodology and yields a series of insightful conclusions (see Section 5.3 and Appendix I for details).

[1] https://github.com/SUFE-AIFLM-Lab/FinEval-KR

2 Related Work

Recent studies highlight a paradigm shift in LLM evaluation, moving from task-specific benchmarks to capability-based assessments of core competencies like knowledge and reasoning (Cao et al., 2025). Financial LLM benchmarks have followed a similar trajectory, evolving through three main phases. Early benchmarks adapted general NLP tasks to the financial domain (e.g., FinGPT (Wang et al., 2023), CFBenchmark (Lei et al., 2023)). As tasks grew more complex, the focus shifted to specialized knowledge evaluation (e.g., FinEval (Zhang et al., 2023), FinTruthQA (Xu et al., 2024)). Contemporary benchmarks now incorporate complex decision-making tasks requiring integrative reasoning, such as market analysis and risk assessment (e.g., InvestorBench (Li et al., 2024a), FinBen (Xie et al., 2024)).

Despite this progress, financial reasoning poses unique challenges that current evaluation methods struggle to address. It demands (1) comprehension of complex, multi-modal data (e.g., time-series and unstructured text), (2) deep domain-specific knowledge, and (3) advanced computational and deductive skills. This complexity creates a critical limitation in existing benchmarks: overall performance metrics entangle knowledge mastery with reasoning ability, making it impossible to diagnose the true source of a model's failure. This fundamental challenge motivates our primary contribution: a decoupled assessment framework. Furthermore, these unique demands directly guided the design of our financial reasoning dataset (see Section 4.1).

Beyond the issue of entangled evaluation, a second key limitation exists: the lack of fine-grained cognitive analysis. While cognitive science perspectives offer crucial insights for LLM optimization (Huber and Niklaus, 2025; Adams, 2015), they are largely overlooked in financial benchmarks. This gap hinders targeted model improvement, as engineers cannot pinpoint specific cognitive deficiencies (e.g., analysis vs. evaluation) to address during fine-tuning. Our work fills this gap by introducing a cognitive evaluation dimension, enabling a more diagnostic approach to model development.

3 Preliminary Experiment

To further illustrate our research motivation, we design a set of preliminary experiments focusing on a simple reasoning task in the financial domain – a financial calculation problem that requires only a single formula. For the dataset and prompts used in this experiments, please refer to Appendix A.

3.1 Experiment Settings

In order to correctly solve such problems, LLMs need to complete the following three sub-tasks: (1) Recall the calculation formula for the variable to be solved. (2) Identify the variable names and their values from the problem statement. (3) Substitute the variable values into the formula and calculate the target variable. According to Bloom's Taxonomy, these three sub-tasks correspond to remembering, understanding and applying/analyzing respectively. Obviously, the model can only answer the questions correctly if all these subtasks are done correctly.

We design the following three comparative experiments to evaluate the impact of knowledge on reasoning task: *Experiment 1* (E1). The LLM is directly prompted with the original question and asked to complete the entire reasoning process independently. *Experiment 2* (E2). Based on experiment 1, inject variables and their values into the prompt. *Experiment 3* (E3). Further provide formulas for calculation based on experiment 2.

To ensure the fairness, we add an equal amount of irrelevant information as distractors to the control groups, while providing the key knowledge in the experimental groups. The experiments are first conducted on the Qwen2.5-7B_Instruct, and the generalizability of the findings is verified using GPT-4o. The experimental results are presented in Table 1.

Model/Settings	E1	E2	E3
Qwen2.5-7B_Instruct	58.0%	72.4%	85.9%
GPT-4o	64.5%	80.5%	92.5%

Table 1: Cumulative correctness rate of reasoning task in three experimental settings.

3.2 Results Analysis

Experiment 3 to 1 can be regarded as knowledge stripping experiments. As key knowledge is progressively removed, the problem-solving rate decreases significantly, suggesting that the lack of knowledge is often the root cause of reasoning failure in reasoning tasks. This leads to the following conclusion.

In complex domain reasoning tasks, knowledge is necessary for successful reasoning.

On the contrary, Experiments 1 to 3 can be regarded as knowledge enhancement experiments, and the results show that the reasoning success rate of LLM is significantly improved after the introduction of key knowledge in the prompts. However, even knowledge is sufficiently injected into the prompt, GPT-4o still persists with an error rate of about 7.5%, suggesting that reasoning ability may become a performance bottleneck for such tasks. Combined with the previous conclusion, we infer that:

Knowledge is a necessary but not sufficient condition for successful reasoning.

It is noteworthy that both models exhibit significant knowledge dependence in all experimental settings, while the performance gap between them persists. This performance discrepancy suggests that there may be significant differences in the knowledge and reasoning capabilities of different models. Therefore, the decoupled evaluation framework can help us identify more clearly the shortcomings of the models in terms of knowledge and reasoning capabilities.

From a cognitive science perspective, knowledge stripping experiments revealed the damaging effects of lower-order cognitive deficits on higher-order reasoning, as evidenced by a 27.94% and 28% decrease in reasoning accuracy in qwen2.5-7b and GPT-4o, respectively. This change also supports the progressive dependence between cognitive levels. Furthermore, the performance difference between the two models in the same experiment settings suggests that there is a significant gap between them, at least in the remembering layer.

4 FinEval-KR

In this section, we present a **Fin**ancial domain **Eval**uation framework for assessing **K**nowledge and **R**easoning abilities (**FinEval-KR**). We first describe the methodology used to construct the evaluation dataset. Next, we introduce a multi-stage evaluation framework that performs root cause analysis of reasoning errors via knowledge-augmented question answering, enabling a decoupled assessment of a model's knowledge mastery and reasoning ability. Finally, we define a series of evaluation metrics: a *knowledge score* based on domain knowledge coverage; a *reasoning score* and a *cognitive score* which is based on Bloom's taxonomy. This approach improves the interpretability of LLM evaluations in financial scenarios and provides clear directions for targeted model improvement.

4.1 Benchmark Dataset Construction

The FinEval-KR benchmark dataset is constructed through four steps: data collection and processing, automated question generation, answer generation, and dataset annotation. This framework ensures comprehensive coverage of financial knowledge domains while maintaining academic rigor. All prompt templates for the dataset generation are detailed in Appendix B.

Corpus Collection To ensure the dataset is both authoritative and relevant, we selected nine canonical textbooks from major financial disciplines. These sources provide a comprehensive and up-to-date overview of modern finance. This process yielded a financial corpus totaling 8,460 pages. A detailed list of the textbooks and the rationale for their selection is available in Appendix C.

Question Generation We generate financial problems from the obtained corpus using a two-stage process. First, we use a custom-designed prompt to instruct OpenAI o1, a state-of-the-art model at the time of our research that was specially enhanced for reasoning capabilities, to create a computational question based on a given text segment. The prompt's design ensures the question meets predefined standards (see Figure 6 in the appendix for details). Second, we subject each candidate question to a three-step automated validation: it is checked for logical coherence, consistency with the source material, and overall quality. Questions that fail any check are discarded, ensuring the high fidelity of the final dataset. The details of validation please refer to Appendix D.

Answer Generation We generate and validate the ground-truth answers using a structured three-stage pipeline: (1) Unconstrained generation: We first prompt OpenAI o1 to solve each problem, guided by the prompt shown in Figure 7. Crucially, we impose no initial format constraints on the output, a strategy designed to capture the model's most natural and diverse problem-solving pathways. (2) Standardized formatting: The resulting raw solutions are then systematically parsed and reformatted according to a predefined template to ensure consistent and uniform presentation across the dataset. (3) Rigorous validation: Finally, each formatted answer undergoes a triple-validation protocol, which mirrors the question validation process (see Appendix D for details).

Dataset Annotation For each question, we use OpenAI o1 guided by the prompt in Figure 8 to identify all requisite knowledge points. These include, but are not limited to, core financial concepts, regulatory frameworks and mathematical operations.

For each step in the answer, we annotate the corresponding cognitive level using Bloom's Taxonomy. To mitigate the inherent subjectivity of this task, we developed a constrained, keyword-driven methodology. We guide the OpenAI o1 using a predefined set of keywords strongly associated with each cognitive level, derived from Anderson and Krathwohl (2001) (see Figure 9 for prompt template). This process maps each reasoning step to one or more levels: Remembering, Understanding, Applying, Analyzing, and Evaluating. We intentionally exclude the Creating level. This decision ensures objectivity, as our dataset comprises problems with determinate solutions, a principle that aligns with standardized financial certification exams.

Dataset Characteristics and Exemplary Samples The final evaluation set contains a total of 9,782 question-answer pairs and their associated annotations. The questions and answers in the dataset are verified by human experts, and a sample check showed that the accuracy of the dataset is above 90%. Further details on the human annotators, the quality control mechanisms, and the validation process are provided in the Appendix J. An exemplary sample from the constructed dataset is given in Figure 1 on a yellow background. (see Figure 10 for the complete example). The complete statistical characterization of the dataset is detailed in Appendix E.

4.2 Evaluation Framework

As shown in Figure 1, the FinEval-KR framework comprises three evaluation stages. First, a model attempts a problem, and an LLM-as-a-judge identifies any errors, extracting the specific knowledge points needed for a correct solution. In the second stage, we provide the model with this missing knowledge and have it re-answer the question. The final stage performs a comparative analysis for attributing the initial error to knowledge deficiency or reasoning ability deficiency. This mechanism allows for the independent quantification of a model's knowledge and reasoning capabilities.

4.2.1 Stage 1: Question Answering

Unconstrained Solution Generation In this initial stage, the model under evaluation generates a solution without any format constraints. This design is crucial for preventing judgment errors based on superficial format-matching. It ensures, for example, that a logically sound reasoning path is not unfairly penalized simply for deviating from the step-order of the reference answer. The prompt for this stage is detailed in Appendix F, Figure 15.

Structured Judgment and Error Analysis Next, we employ an LLM-as-a-judge to analyze the generated solution against a reference answer. To ensure the judgment is objective and reproducible, the judge is guided by a highly structured, Chain-of-Thought (CoT) prompt that enforces a rigorous step-by-step analysis (see Figure 14). Additionally, we address the potential bias challenge of the judge model in Appendix G.

If an error is detected, the judge's output, termed the *review result*, pinpoints the first incorrect step, identifies its root cause, and lists the corresponding knowledge deficiencies (an example is shown in Figure 2). If the final answer is correct, we consider the entire reasoning process valid. This assumption is grounded in the novelty and multi-step complexity of our dataset, which makes correct answers via guessing or exploiting artifacts (the "Clever Hans" effect) highly improbable.

4.2.2 Stage 2: Knowledge-augmented Answering

If the evaluated model makes a error in Stage 1, the framework proceeds to this second stage. Here, we provide the model with the exact knowledge points that the judge identified as missing in Stage 1. This knowledge is integrated into a new prompt (see Appendix F, Figure 16), instructing the model to re-attempt the problem. The core purpose of this stage is to isolate the reasoning variable. By explicitly providing the necessary knowledge – a prerequisite for correct reasoning as validated in our preliminary study – we can now assess if the model can reason correctly when its knowledge gaps are filled. The judge then re-evaluates the new solution using the same protocol as in Stage 1. The outcome of this assessment is termed the *augmented review result*.

4.2.3 Stage 3: Error Diagnosis

The final stage performs a comparative analysis between the outcomes of Stage 1 and Stage 2 to determine the root cause of the initial error. Our approach is grounded in the principle that *LLM's preference to external information reveals its internal knowledge gaps* (Wu et al., 2024). Specifically, the judge model compares the *review result* with the *augmented review result*, following the principles below to determine the root cause of reasoning errors:

- *Knowledge Deficiency*. If the augmented review result shows that the model reasons correctly in Stage 2, or the erroneous step occurs later than in Stage 1. This indicates that the evaluated model preferred the augmented knowledge in the second stage. This proves that the initial error is caused by knowledge deficiency.

- *Reasoning Ability Deficiency*. If the evaluated model still makes a reasoning error in Stage 2, and the erroneous step is consistent with Stage 1, this indicates that the evaluated model still preferred its internal prior knowledge. This proves that the initial error is rooted in poor reasoning ability.

Through this attribution method, our framework successfully decouples a model's knowledge and reasoning abilities from its overall task performance.

4.2.4 Evaluation Metrics

We propose three core metrics: knowledge score, reasoning score, and cognitive score, while also retaining accuracy to measure the model's overall task performance.

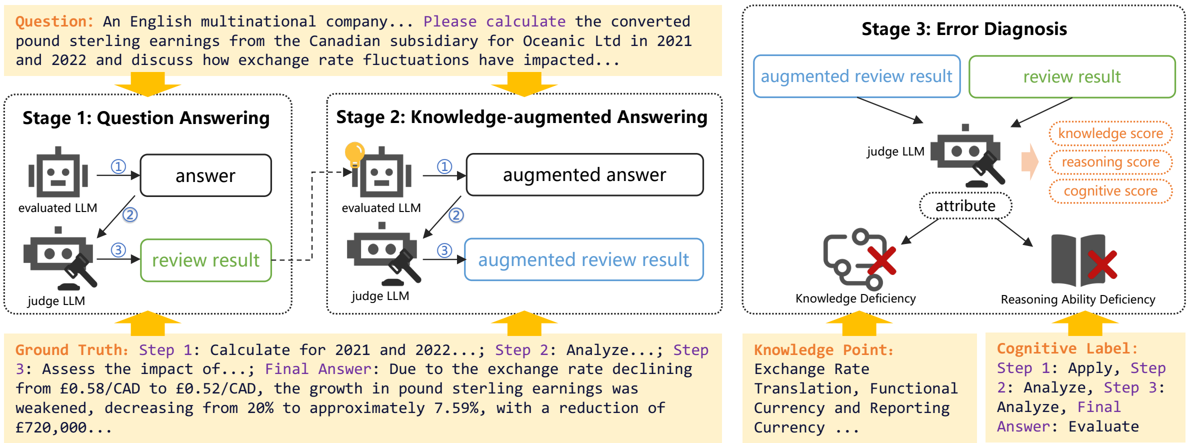


Figure 1: Three-stage evaluation framework of FinEval-KR, and an exemplary sample of the dataset. Note that the original dataset is in Chinese, the figure provides an English translation for readability.

Review Result

<cause of error> The annual interest rate of 6% should be converted to a monthly interest rate, i.e. 0.5% (6%/12). </cause of error>

<location> Step 1 </location>

<missing knowledge> The calculation of fixed monthly payments requires using the monthly interest rate, not the annual interest rate. </missing knowledge>

<result> Incorrect </result>

Figure 2: Example of review result generated by the judge model (original in Chinese, with English translation).

Knowledge Score (KS) This metric quantifies the evaluated model's knowledge coverage in the financial domain.

$$KS = 1 - \frac{\left|\bigcup_{i=1}^{M} K_i'\right|}{\left|\bigcup_{i=1}^{N} K_i\right|}, \tag{1}$$

where M is the number of errors attributed to knowledge deficiency during evaluation. K_i' denotes the set of knowledge points involved in the erroneous reasoning steps for the i-th question whose error was attributed to a knowledge deficiency. N is the total number of evaluation samples. K_i denotes the set of knowledge points involved in the i-th evaluation question. The denominator in the Eq. (1) is the total number of knowledge points across all evaluation questions in the dataset.

Reasoning Score (RS) This metric measures the evaluated model's reasoning ability in the financial domain.

$$RS = \frac{\sum_{i=1}^{N} \mathbb{I}(a_i = a_i^{\text{ref}})}{N - \sum_{i=1}^{N} \mathbb{I}(a_i \neq a_i^{\text{ref}} \wedge r(a_i) = \text{K})}, \tag{2}$$

where a_i is the answer of the evaluated model for the i-th question, and a_i^{ref} is the corresponding reference answer. Thus, the numerator of Eq. (2) is the total number of questions with correct reasoning. $r(a_i)$ represents the root cause for the erroneous result a_i, which can be either K or R, representing knowledge and reasoning ability deficiency, respectively. In addition, since each reasoning step in the dataset is annotated with a cognitive label, the reasoning ability deficit can be further subdivided into the ability deficit at a certain cognitive level, i.e., R^j, $j \in \{1, 2, 3, 4, 5\}$, where j is the index of the level in Bloom's taxonomy, and $1, 2, 3, 4, 5$ denote remembering, understanding, applying, analyzing, and evaluating respectively. The denominator in Eq. (2) is the total number of questions whose reasoning errors are not attributed to knowledge deficiencies.

Cognitive Score (CS) Furthermore, to explore the level of cognition exhibited by LLMs when solving reasoning tasks, we expanded a series of fine-grained metrics drawing on RS. The j-th cognitive level score of the LLM is defined as,

$$CS_j = RS \times (1 - \alpha_j \times \frac{\sum_{i=1}^{N} \mathbb{I}(r(a_i) = \text{R}^j)}{N - \sum_{i=1}^{N} \mathbb{I}(a_i \neq a_i^{\text{ref}} \wedge r(a_i) = \text{K})}). \tag{3}$$

$\alpha_j \in (0, 1)$ is a penalty coefficient, and it is designed to have negative correlation with the cognitive level j. This weighting scheme is designed to heavily penalize errors made by the model during lower-level cognitive reasoning steps. In our experiments, we empirically set $\alpha_1, \cdots, \alpha_5$ to a linearly decreasing sequence 0.9, 0.8, 0.7, 0.6 and 0.5.

Accuracy (Acc) Finally, the success rate for all reasoning tasks is defined as,

$$Acc = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(a_i = a_i^{\text{ref}}). \tag{4}$$

5 Experiments

This section first validates the FinEval-KR framework's effectiveness in identifying potential knowledge weaknesses and performing root cause analysis. Subsequently, we selected a range of LLMs, and evaluated them using the proposed FinEval-KR framework and our open-sourced dataset.

5.1 Alignment Analysis of Judge Model in FinEval-KR

The core of FinEval-KR lies in the design of the judge model for identifying initial reasoning step, recalling potential knowledge weaknesses and pinpointing root causes, which is fundamental for achieving decoupled evaluation. To this end, we design a set of comparative experiments to assess the alignment between the judge model and human evaluation.

The methods participating in the comparison include: (a) *Direct Prompting*: We use advanced models like OpenAI o1, directly prompting them to execute knowledge identification and root cause localization tasks. (b) *Task Decomposition*: Decompose the above two tasks into multiple subtasks and clearly define the logical dependencies between these subtasks (c) *Ours*: Based on task decomposition, this method requires the model to self-reflect and explicitly output reasoning processes step-by-step before generating the final conclusion (i.e., adding the `<Inner Thought>` as shown in Figure 14).

Given that our benchmark dataset is composed in Chinese, we select Qwen2.5-72B_Instruct, which is specifically optimized through extensive pre-training on Chinese corpora, as backbone LLM in the latter two methods. We use the following three metrics to evaluate the performance of these methods: (a) *Accuracy of error identification*: This refers to the proportion of correctly localized initial reasoning step in the Stage 1. (b) *Accuracy of recalled knowledge points*: The percentage of missing knowledge correctly recalled in Stage 1. (c) *Accuracy of error attribution* : The proportion of correctly attributed reasoning errors to knowledge or reasoning ability deficits.

Methods	Error Identification	Knowledge Recall	Error Attribution
Direct Prompting	0.56	0.24	0.20
Task Decomposition	0.85	0.60	0.53
Ours	**0.92**	**0.94**	**0.93**

Table 2: Accuracy for three evaluation tasks.

All metrics are calculated after a manual review of the model outputs by financial domain experts. Experimental results are presented in Table 2. It can be observed that FinEval-KR outperforms the comparison methods across all metrics, demonstrating its superior performance in error attribution and knowledge identification tasks.

Additionally, we discuss the limitations of adopting Qwen2.5-72B_Instruct as the judge model in the Limitations section.

5.2 Evaluation

For our evaluation, we select 18 leading and representative LLMs, referencing prominent leaderboards like the Chatbot Arena[2]. This selection spans a diverse range of models, including open-source and closed-source systems, models of varying parameter scales, and different architectures such as dense and Mixture-of-Experts (MoE) (see Appendix H for details).

In our analysis, we focus on the relative performance rankings (i.e., performance tiers) of these models rather than their absolute scores. This approach is designed to ensure the robustness of our findings. While an LLM-as-a-judge may have inherent systematic biases, such biases have a smaller impact on the relative ordering of models than on their absolute scores.

5.3 Results and Core Findings

Table 3 lists the average metrics for all 18 models from three independent runs, conducted with a model temperature of 1. The complete results, including standard deviations, are presented in Table 7.

[2] https://openlm.ai/chatbot-arena/

Model/Metrics	Acc	KS	RS	CS_1 (remember)	CS_2 (understand)	CS_3 (apply)	CS_4 (analyze)	CS_5 (evaluate)
Qwen2.5-14B_Instruct	0.5473	0.8490	0.6863	0.6547	0.6603	0.3893	0.6863	0.6820
QwQ-32B-preview	0.7380	0.9073	0.8627	0.8450	0.8503	0.6987	0.8510	0.8597
DeepSeek-v3	0.8270	0.9427	0.9077	0.8963	0.8993	0.7963	0.8943	0.9057
DeepSeek-R1	0.8700	0.9517	**0.9347**	**0.9377**	**0.9397**	**0.8810**	**0.9380**	**0.9433**
Doubao-pro-32k	0.7825	0.9195	0.8750	0.8560	0.8600	0.7340	0.8565	0.8720
Moonshot-v1-128k	0.4533	0.8340	0.6020	0.5620	0.5670	0.2763	0.5653	0.5973
Ernie-bot-4.0	0.5733	0.8627	0.7053	0.6680	0.6753	0.4383	0.6847	0.6927
Qwen-max-latest	0.6467	0.8797	0.7733	0.7507	0.7547	0.5340	0.7440	0.7703
GPT-3.5-turbo	0.2830	0.7527	0.3973	0.3527	0.3603	0.0900	0.3893	0.3970
GPT-4o	0.6853	0.9020	0.8067	0.7847	0.7890	0.5930	0.7870	0.8030
GPT-4.1	0.8263	0.9520	0.9063	0.8957	0.8977	0.7890	0.8927	0.9050
o1-mini	0.7503	0.8997	0.8453	0.8340	0.8363	0.6983	0.8477	0.8450
o3-mini	0.8207	0.9260	0.9070	0.9047	0.9073	0.8127	0.9023	0.9120
Gemini-2.5-pro	**0.8750**	**0.9627**	0.9233	0.9123	0.9163	0.8403	0.9050	0.9120
Gemini-2.5-flash	0.8440	0.9540	0.9203	0.9103	0.9133	0.8307	0.9100	0.9177
Claude-3.7-sonnet	0.7923	0.9390	0.8823	0.8663	0.8703	0.7433	0.8653	0.8803
Xuanyuan-FinX1-preview	0.5890	0.8687	0.7323	0.7063	0.7130	0.4610	0.7323	0.7300
Fin-R1-7B	0.4153	0.7510	0.5570	0.5190	0.5277	0.2170	0.5570	0.5527

Table 3: Reasoning Accuracy (Acc), Knowledge Score (KS), Reasoning Score (RS), and Cognitive Score (CS) of evaluated LLMs on the FinEval-KR. The complete results please refer to Table 7 in the appendix.

The analysis of the model performance echelons and discussion of the results are detailed in in Appendix I. In summary, the comprehensive analysis of all evaluation metrics identifies the current Tier 1 models as DeepSeek-R1, Gemini-2.5-pro and Gemini-2.5-flash. These models typically have parameters in excess of a hundred billion and use the MoE architecture to optimize computational resources. Furthermore, they specifically optimize reasoning capabilities through methods like reinforcement learning, and demonstrate outstanding performance in knowledge coverage and the completeness of reasoning paths.

Bottleneck in Knowledge Applying Abilities Our analysis reveals that it is reasoning and specific cognitive skills, not merely knowledge, that truly drive performance in advanced LLMs. While the Knowledge Score (KS) and Reasoning Score (RS) are positively correlated, KS scores converge among top-tier models, indicating that sheer knowledge is no longer the primary performance bottleneck. Instead, RS shows a strong correlation with accuracy, establishing reasoning ability as crucial for success. A deeper cognitive analysis pinpoints the ability to apply knowledge (CS_3) as the critical differentiator, evidenced by a sharp drop in this metric, which directly degrades their reasoning and accuracy.

Crucially, this weakness is not confined to lower-tier models. Even top-tier LLMs exhibit a significant deficit in applying knowledge (CS_3) compared to their abilities in analyzing (CS_4) or evaluating (CS_5). For instance, GPT-4.1 scores 0.7890 in CS_3 versus 0.8927 in CS_4. This universal shortcoming underscores a fundamental limitation of current models: a profound difficulty in transferring theoretical knowledge to practical, real-world application.

The Dilemma of Financial LLMs Our evaluation establishes Xuanyuan-FinX1-preview as the leading specialized financial LLM, consistently outperforming counterparts like Fin-R1-7B. However, a more critical finding emerges when comparing it to state-of-the-art general LLMs. Despite its domain leadership, Xuanyuan-FinX1-preview exhibits a significant performance gap of above 20%, a deficit that spans across financial complex reasoning, and higher-order cognitive skills. We attribute this gap to the superior generalization capabilities of leading general LLMs, which are developed through pre-training on vast, multi-domain datasets and benefit from more rapid iteration cycles. This advantage allows them to achieve strong performance even in financial fields, highlighting the limitations of current financial LLMs in terms of data diversity and development velocity.

Consequently, we predict a dual-track future for developing high-performance financial LLMs. The first track involves building upon state-of-the-art general foundation models to leverage their vast world knowledge and robust generalization. The second requires employing advanced fine-tuning techniques to distill and align these models for specialized financial reasoning, moving beyond a simple reliance on domain data.

6 Conclusion

This paper introduces FinEval-KR, a novel evaluation framework designed to decouple and assess the knowledge and reasoning abilities of LLMs in the financial domain, supplemented by a cognitive perspective and a new dataset. Our evaluation results indicate that reasoning and higher-order cognitive abilities are crucial for reasoning accuracy. Even top models encounter a bottleneck in knowledge application, and specialized financial models generally lag behind top general LLMs.

Limitations

A potential limitation of this study lies in the choice of the judge model. Our primary experiments were conducted using Qwen-2.5-72B_Instruct, which represented the state-of-the-art among publicly available models with strong Chinese support during our experimental phase in late 2024. With the rapid evolution of large language models, even more capable reasoning models like DeepSeek-R1 have since emerged.

To investigate the impact of this evolution, we performed a small-scale evaluation using DeepSeek-R1 as the judge model. The results revealed a clear performance-efficiency trade-off: while DeepSeek-R1 yielded a marginal accuracy improvement of approximately 3%, it nearly doubled the inference time, posing significant challenges for large-scale evaluation. Crucially, we found that although the absolute scores of the evaluated models slightly increased, their relative rankings and the performance gaps between them remained highly consistent. Since our research focuses on the comparative performance of different methods, this consistency confirms the robustness of our conclusions.

Our future work will focus on enhancing the framework's robustness by incorporating reasoning LLMs and more diverse evaluation paradigms. Specifically, we plan to employ multiple, heterogeneous models for dataset generation and implement a cross-validated, multi-judge evaluation pipeline to minimize potential biases, raise the evaluation ceiling, and bolster the benchmark's overall reliability.

References

Nancy E Adams. 2015. Bloom 's taxonomy of cognitive learning objectives. *Journal of the Medical Library Association: JMLA*, 103(3):152.

Lorin W Anderson and David R Krathwohl. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives: complete edition*. Addison Wesley Longman, Inc.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, and 1 others. 2023. Improving image generation with better captions. *Computer Science. https://cdn.openai.com/papers/dall-e-3.pdf*, 2(3):8.

Yixin Cao, Shibo Hong, Xinze Li, Jiahao Ying, Yubo Ma, Haiyuan Liang, Yantao Liu, Zijun Yao, Xiaozhi Wang, Dan Huang, and 1 others. 2025. Toward generalizable evaluation in the LLM era: A survey beyond benchmarks. *arXiv preprint arXiv:2504.18838*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Thomas Huber and Christina Niklaus. 2025. LLMs meet Bloom 's taxonomy: A cognitive view on large language model evaluations. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 5211–5246. Association for Computational Linguistics.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 517–545.

Yang Lei, Jiangtong Li, Ming Jiang, Junjie Hu, Dawei Cheng, Zhijun Ding, and Changjun Jiang. 2023. Cfbenchmark: Chinese financial assistant benchmark for large language model. *arXiv preprint arXiv:2311.05812*.

Haohang Li, Yupeng Cao, Yangyang Yu, Shashidhar Reddy Javaji, Zhiyang Deng, Yueru He, Yuechen Jiang, Zining Zhu, Koduvayur Subbalakshmi, Guojun Xiong, and 1 others. 2024a. InvestorBench: A benchmark for financial decision-making tasks with LLM-based agent. *arXiv preprint arXiv:2412.18174*.

Zhiming Li, Yushi Cao, Xiufeng Xu, Junzhe Jiang, Xu Liu, Yon Shin Teo, Shang-Wei Lin, and Yang Liu. 2024b. LLMs for relational reasoning: How far are we? In *Proceedings of the 1st International Workshop on Large Language Models for Code*, pages 119–126.

Xinyu Liu and Ke Jin. 2024. MTFinEval: A multi-domain chinese financial benchmark with eurypalynous questions. *arXiv preprint arXiv:2408.10921*.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, and 1 others. 2025. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*.

Ying Nie, Binwei Yan, Tianyu Guo, Hao Liu, Haoyu Wang, Wei He, Binfan Zheng, Weihao Wang, Qiang Li, Weijian Sun, and 1 others. 2024a. Cfinbench: A comprehensive chinese financial benchmark for large language models. *arXiv preprint arXiv:2407.02301*.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024b. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint arXiv:2406.11903*.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2024. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. *arXiv preprint arXiv:2406.12624*.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36:38975–38987.

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. FinGPT: Instruction tuning benchmark for open-source large language models in financial datasets. *arXiv preprint arXiv:2310.04793*.

Weiqi Wang and Yangqiu Song. 2024. MARS: benchmarking the metaphysical reasoning abilities of language models with a multi-task evaluation dataset. *arXiv preprint arXiv:2406.02106*.

Xinlin Wang and Mats Brorsson. 2025. Can large language model analyze financial statements well? In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 196–206.

Zhihu Wang, Shiwan Zhao, Yu Wang, Heyuan Huang, Sitao Xie, Yubo Zhang, Jiaxin Shi, Zhixing Wang, Hongyan Li, and Junchi Yan. 2024. Re-task: Revisiting LLM tasks from capability, skill, and knowledge perspectives. *arXiv preprint arXiv:2408.06904*.

Kevin Wu, Eric Wu, and James Zou. 2024. How faithful are rag models? quantifying the tug-of-war between rag and llms' internal prior. *CoRR*, abs/2404.10198.

Minghao Wu and Alham Fikri Aji. 2025. Style over substance: Evaluation biases for large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 297–312.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, and 1 others. 2024. FinBen: A holistic financial benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:95716–95743.

Ziyue Xu, Peilin Zhou, Xinyu Shi, Jiageng Wu, Yikang Jiang, Dading Chong, Bin Ke, and Jie Yang. 2024. FintruthQA: A benchmark dataset for evaluating the quality of financial information disclosure. *arXiv preprint arXiv:2406.12009*.

Liwen Zhang, Weige Cai, Zhaowei Liu, Zhi Yang, Wei Dai, Yujie Liao, Qianru Qin, Yifei Li, Xingyu Liu, Zhiqiang Liu, and 1 others. 2023. FinEval: A chinese financial domain knowledge evaluation benchmark for large language models. *arXiv preprint arXiv:2308.09975*.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8783–8817.

Zhihan Zhang, Yixin Cao, and Lizi Liao. 2025. XFIN-BENCH: Benchmarking llms in complex financial problem solving and reasoning. *arXiv preprint arXiv:2508.15861*.

A The Dataset and Prompts Used in Preliminary Experiments

We manually construct a dataset consisting of 200 samples. Each sample in this dataset includes the question, the financial formula being examined, the mapping between the formula's variables and the specific numerical values, and the ground truth to the question. Figure 3, 4, and 5 show the prompt templates and example samples used in experiments 1, 2, and 3, respectively. In these figures, the content of the prompt template is shown in blue text, while the test samples are shown in black text. The ground truth for this problem is 0.1024 or 10.24%.

你是一名金融学专家，现在你需要用中文解答以下题目。
You are a finance expert and now you need to answer the following question in Chinese.

题目：某投资者于2022年初购入了一只股票基金，初始投资额为50,000元。该基金在过去几年表现出色，截至2025年4月7日，基金价值增长至67,000元。同时，据市场分析，同类基金的平均市盈率为18倍，股息率为2%，而该基金的贝塔系数为1.2。考虑到这些因素，请计算该投资者在这段时间内的复合年均增长率。
Question: An investor purchased a stock fund at the beginning of 2022 with an initial investment of 50,000 yuan. The fund has performed well in recent years, and as of April 7, 2025, the fund's value has grown to 67,000 yuan. Meanwhile, according to market analysis, the average P/E ratio for similar funds is 18 times, the dividend yield is 2%, and the beta coefficient of this fund is 1.2. Considering these factors, please calculate the investor's Compound Annual Growth Rate during this period.

请一步一步思考，并在最后给出你的最终答案。
Please think step by step and give your final answer at the end.

Figure 3: The prompt for experiment 1 and an exemplary sample (original in Chinese, with English translation).

你是一名金融学专家，现在你需要用中文解答以下题目。
You are a finance expert and now you need to answer the following question in Chinese.

题目：某投资者于2022年初购入了一只股票基金，……
Question: An investor purchased a stock fund at the beginning of 2022...

这是计算本题目你需要用到的变量及其值：'初值（PV）': '50000.0', '终值（FV）': '67000.0', '年数（n）': '3.0'
Here are the variables and their values you will need to calculate this problem: 'Initial Value (PV)': '50000.0', 'Final Value (FV)': '67000.0', 'Number of Years (n)': '3.0'

请一步一步思考，并在最后给出你的最终答案。
Please think step by step and give your final answer at the end.

Figure 4: The prompt for experiment 2 and an exemplary sample (original in Chinese, with English translation).

你是一名金融学专家，现在你需要用中文解答以下题目。
You are a finance expert and now you need to answer the following question in Chinese.

题目：某投资者于2022年初购入了一只股票基金，……
Question: An investor purchased a stock fund at the beginning of 2022...

这是计算本题目你需要用到的变量及其值：'初值（PV）': '50000.0', '终值（FV）': '67000.0', '年数（n）': '3.0'
Here are the variables and their values you will need to calculate this problem: 'Initial Value (PV)': '50000.0', 'Final Value (FV)': '67000.0', 'Number of Years (n)': '3.0'

这是本题主要考察的公式：CAGR = (FV / PV)^(1/n) - 1
This is the main formula examined in this problem: CAGR = (FV / PV)^(1/n) - 1

请一步一步思考，并在最后给出你的最终答案。
Please think step by step and give your final answer at the end.

Figure 5: The prompt for experiment 3 and an exemplary sample (original in Chinese, with English translation).

B The Prompt Templates for the Dataset Generation

Figure 6 shows a prompt template for generating questions for a given subfield based on a given piece of corpus. Figure 7 shows a prompt template that generates an solution to a given question. Figures 8 and 9 show the prompt templates for labeling knowledge points and step-level cognitive abilities, respectively.

C Data Sources for the FinEval-KR Dataset

To ensure our benchmark is both authoritative and comprehensive, we constructed the source corpus from nine classic textbooks in modern finance. This selection provides extensive coverage across key subfields, including corporate finance, investments, financial markets, risk management, and monetary policy. These foundational texts supply a rich combination of core theoretical principles and practical case studies, forming a robust basis for evaluating financial knowledge and reasoning.

We processed the corpus using a three-stage pipeline: extraction, cleaning, and standardization. (1) Extraction: We used OCR to convert all text and mathematical equations from the source materials into a machine-readable Markdown format. (2) Cleaning: We then manually curated the extracted content, removing non-essential sections (e.g., prefaces, appendices) and performing quality assurance checks. (3) Standardization: Finally, we transformed the cleaned content into a structured format suitable for automated processing. This rigorous process ensures the final dataset is of high quality, integrity, and utility.

Corporate Finance

- Selected Textbook: *Corporate Finance* (13th edition, 2021) by Stephen A. Ross, Randolph W. Westerfield, Jeffrey Jaffe, and Bradford D. Jordan.

- Rationale: This textbook is widely used in MBA and undergraduate finance courses. It systematically explains core concepts of modern corporate finance, such as arbitrage theory, net present value (NPV), the efficient market hypothesis, agency theory, and the risk-return tradeoff.

- Covered Financial Subfields: Corporate financing, capital structure, investment decisions, dividend policy, firm valuation, etc.

- Role in the Benchmark Dataset: Provides a solid theoretical foundation and abundant practical examples for reasoning and computational questions in corporate finance.

```
请根据以下材料设计一个以段落形式呈现的{子学科名}题目。
Please design a {Subfield Name} problem presented in paragraph form based on the following materials:
---
{从书本中提取的文本}
{Text Extracted From the Book}
---
题目应满足以下要求：
The problem should meet the following requirements:
1. 提供详尽的数据和背景信息：题目应包含{某些关键信息}，确保学生能够基于这些信息进行准确的计算与分析。
Provide detailed data and background information: The problem should include necessary information such as {Some Key Information}, ensuring students can perform accurate calculations and analysis based on this information.
2. 基础难度：题目难度应适合本科生的基础水平，要求学生进行不超过三步的简单计算和基本的逻辑推理即可解答。
Basic difficulty: The problem's difficulty should be suitable for undergraduate basic level, requiring students to perform simple calculations and basic logical reasoning in no more than three steps
3. 实际意义：题目设计应紧密结合{实际场景}。
Practical significance: The problem design should be closely linked to {Some Practical Scenarios}.
4. 单一问题：题目应集中于一个计算问题，严禁包含多个子问题，以确保焦点明确。
Single question: The problem should focus on a single calculation question and strictly prohibit the inclusion of multiple sub-questions to ensure a clear focus.
5. 段落形式：整个题目应以连贯的段落形式呈现，避免使用分项列表或标题。
Paragraph form: The entire problem should be presented as a continuous paragraph, avoiding the use of bullet points or headings.
6. 语言要求：题目必须以中文形式呈现。
Language requirement: The problem must be presented in Chinese.
```

Figure 6: Prompt template for generating questions for a given subfield based on a given piece of corpus (original in Chinese, with English translation).

```
请一步步地思考以下问题，并在每个步骤中展示推导过程，确保解答准确无误。对于每一个计算步骤，提供详细的解释，并确保逻辑清晰、推理严谨。
Please think step-by-step about the following problem, showing the derivation process in each step to ensure the solution is accurate. For each calculation step, provide a detailed explanation and ensure the logic is clear and the reasoning is rigorous.

问题描述：{生成的金融学推理题}
Problem Description: {Generated Financial Reasoning Problem}
```

Figure 7: Prompt template that generates an solution to a given question (original in Chinese, with English translation).

```
请根据以下题目，归纳总结出该题目涉及的主要知识点。确保知识点数量在3至4个之间，并且简洁明了。
Based on the following problem, summarize the main knowledge points involved. Ensure the number of knowledge points is between 3 and 4 and that they are concise and clear.

题目：{生成的金融学推理题}
Problem: {Generated Financial Reasoning Problem}
```

Figure 8: Prompt template for labeling knowledge points for a given question (original in Chinese, with English translation).

Investments

- Selected Textbook: *Investments* (13th edition, 2023) by Zvi Bodie, Alex Kane, and Alan J. Marcus.
- Rationale: This book deeply explores securities market efficiency, risk-return relationships, and asset allocation strategies. Its content is highly aligned with the CFA (Chartered Financial Analyst) exam syllabus.
- Covered Financial Subfields: Securities markets, asset pricing, portfolio theory, behavioral finance, derivatives, etc.
- Role in the Benchmark Dataset: Offers authoritative theoretical support and practical guidance for reasoning and computational questions in investments.

Financial Institutions and Markets

- Selected Textbook: *Financial Markets & Institutions* (13th edition, 2020) by Jeff Madura.
- Rationale: This book comprehensively analyzes the operational mechanisms and regulatory frameworks of financial institutions like commercial and investment banks. It also provides empirical and case analyses on contemporary hot topics such as stock valuation and market microstructure.
- Covered Financial Subfields: Financial institutions, financial markets, central banking, monetary policy, market regulation, etc.
- Role in the Benchmark Dataset: Supplies a systematic theoretical framework and practical examples for reasoning and computational questions concerning financial institutions and markets.

Money and Banking

- Selected Textbook: *The Economics of Money, Banking, and Financial Markets* (13th edition, 2021) by Frederic S. Mishkin.
- Rationale: This book offers an in-depth analysis, from both theoretical and empirical perspectives, of money demand and supply, commercial banking operations and regulation,

```
请根据布鲁姆分类法的认知层次以及每个层次对应的动词，将以下解题步骤和最终答案映射到一个或多个合适的认知层次。请在每个步骤和最终答案下明确标注映射的层次，并简要说明理由。

布鲁姆分类法认知层次及相关动词：
1. 记忆
  - 定义：能够回忆和识别题中需要的公式、定理和基本概念。
  - 相关动词：定义、列举、回忆、识别。
2. 理解
  - 定义：理解各个公式和概念的意义，能够解释其在不同情境中的应用方式。
  - 相关动词：解释、说明、推断、阐释、理解、总结。
3. 应用
  - 定义：能够将已掌握的公式和方法正确地应用到具体的题中，解决实际问题。
  - 相关动词：执行、使用。
4. 分析
  - 定义：能够分解复杂的题目，识别并理解各个部分之间的关系和结构，从而制定解决策略。
  - 相关动词：分析、分类、验证、归因。
5. 评价
  - 定义：能够评估所得到的结果的正确性和合理性，检查过程中的步骤是否准确，并考虑是否有更高效的解决方法。
  - 相关动词：判断、评价。

任务要求：
将每个步骤和最终答案映射到布鲁姆分类法的一个或多个认知层次，并简要说明映射的理由。

示例：
# 输入
步骤一：{步骤 1关键词} {解题过程 1}
步骤二：{步骤 2关键词} {解题过程 2}
......
最终答案：{最终答案}

# 输出：
步骤一：{步骤 1关键词}
- 内容：{解题过程 1}
- 映射的认知层次：{认知层次标签1}，理由：{分析理由1}，......

步骤二：{步骤 2关键词}
- 内容：{解题过程 2}
- 映射的认知层次：{认知层次标签1}，理由：{分析理由1}，......

最终答案：{最终答案}
- 内容：{最终答案内容}
- 映射的认知层次：{认知层次标签1}，理由：{分析理由1}，......
```

```
Based on the Bloom's Taxonomy and the corresponding verbs for each level, map the following problem-solving steps and the final answer to one or more appropriate cognitive levels. Please clearly indicate the mapped level(s) below each step and the final answer, and briefly explain the reasoning.

Bloom's Taxonomy Cognitive Levels and Related Verbs:
1. Remember
  - Definition: Ability to recall and recognize formulas, theorems, and basic concepts needed in the problem.
  - Related verbs: Define, list, recall, recognize.
2. Understand
  - Definition: Understanding the meaning of various formulas and concepts, and being able to explain how they are applied in different situations.
  - Related verbs: Explain, describe, infer, interpret, understand, summarize.
3. Apply
  - Definition: Ability to correctly apply learned formulas and methods to specific problems to solve practical issues.
  - Related verbs: Execute, use.
4. Analyze
  - Definition: Ability to decompose the complex problems, identify and understand the relationships and structure between different parts, and thus formulate a solution strategy.
  - Related verbs: Analyze, classify, verify, attribute.
5. Evaluate
  - Definition: Ability to assess the correctness and reasonableness of the obtained result, check the accuracy of the steps in the process, and consider if there are more efficient solution methods.
  - Related verbs: Judge, evaluate.

Task Requirements:
Map each step and the final answer to one or more cognitive levels of Bloom's Taxonomy, and briefly explain the reason for the mapping.

Example:
# Input
Step 1: {Keywords of Step 1} {Process 1}
Step 2: {Keywords of Step 2} {Process 2}
... ...
Final Answer: {Final Answer}

# Output:
Step 1: {Keywords of Step 1}
- Content: {Process 1}
- Mapped Cognitive Level(s): {Cognitive Label 1}, Reason: {Reason 1}, ...

Step 2: {Keywords of Step 2}
- Content: {Process 2}
- Mapped Cognitive Level(s): {Cognitive Label 1}, Reason: {Reason 1}, ...

Final Answer: {Final Answer}
- Content: {Content of Final Answer}
- Mapped Cognitive Level(s): {Cognitive Label 1}, Reason: {Reason 1}, ...
```

Figure 9: Prompt template for labeling step-level cognitive labels for a given answer (original in Chinese, with English translation).

and the interaction mechanisms between monetary policy tools and financial markets.

- Covered Financial Subfields: Monetary theory, banking systems, monetary policy, financial markets, etc.

- Role in the Benchmark Dataset: Delivers in-depth theoretical analysis and empirical support for reasoning and computational questions in money and banking.

International Finance

- Selected Textbook: *International Financial Management* (6th edition, 2023) by Jeff Madura and Roland Fox.

- Rationale: In the context of globalization, this textbook discusses cross-border capital flows, exchange rate volatility, and risk management strategies. It uses numerous case studies to examine practical operations in the international financial environment.

- Covered Financial Subfields: International capital flows, exchange rate theory, foreign exchange markets, international investment, multinational corporate financial management, etc.

- Role in the Benchmark Dataset: Provides a global perspective and practical examples for reasoning and computational questions in international finance.

Financial Risk Management

- Selected Textbook: *Risk Management and Financial Institutions* (6th edition, 2022) by John C. Hull.

- Rationale: This book comprehensively reviews methods for measuring and hedging market risk, credit risk, and operational risk. It places particular emphasis on the application of financial derivatives in risk management.

- Covered Financial Subfields: Risk management, financial derivatives, financial institution regulation, risk measurement and hedging, etc.
- Role in the Benchmark Dataset: Offers a systematic risk analysis framework and practical guidance for reasoning and computational questions in financial risk management.

Fixed Income Securities

- Selected Textbook: *Fixed Income Securities: Tools for Today's Markets* (4th edition, 2022) by Bruce Tuckman and Angel Serrat.
- Rationale: This book provides detailed discussions on the pricing principles and trading strategies for fixed income products such as government bonds, interest rate swaps, and credit default swaps.
- Covered Financial Subfields: Fixed income securities, bond pricing, interest rate derivatives, credit risk, etc.
- Role in the Benchmark Dataset: Provides authoritative pricing models and practical examples for reasoning and computational questions related to fixed income securities.

Financial Engineering and Derivatives

- Selected Textbook: *Options, Futures, and Other Derivatives* (10th edition, 2018) by John C. Hull and Basu Sankarshan.
- Rationale: This textbook comprehensively covers core topics in financial engineering, including option pricing models, futures contract structures, and the pricing of interest rate and credit derivatives.
- Covered Financial Subfields: Derivatives markets, option pricing, futures contracts, interest rate derivatives, credit derivatives, etc.
- Role in the Benchmark Dataset: Offers in-depth theoretical analysis and practical guidance for reasoning and computational questions in financial engineering and derivatives.

Monetary Theory and Policy

- Selected Textbook: *Monetary Theory and Policy* (4th edition, 2017) by Carl E. Walsh.
- Rationale: This book systematically explains the framework of modern monetary theory, focusing on the transmission mechanisms of various monetary policy tools and their effectiveness in low-interest-rate environments.
- Covered Financial Subfields: Monetary theory, monetary policy, macroeconomic models, policy transmission mechanisms, etc.
- Role in the Benchmark Dataset: Provides a macroeconomic perspective and policy analysis framework for reasoning and computational questions in monetary theory and policy.

In summary, these nine textbooks are not only authoritative and reliable but also closely aligned with current academic frontiers. They lay a comprehensive and in-depth academic foundation for the financial reasoning and computation benchmark dataset constructed in this study. This ensures that the test questions possess both professional depth and practical relevance.

D Validation in Question and Answer Generation

For both the question and answer generation phases, we adopted a three-stage verification process. The verification focus for each stage is detailed in Table 4 and Table 5, respectively.

E Statistical Characterization of the FinEval-KR Dataset

A complete sample from the constructed dataset is shown in Figure 10. The sample size and its distribution for each subdiscipline in FinEval-KR Dataset is shown in Figure 11, where the subdiscipline categorization methodology refers to Zhang et al. (2023). The top-50 knowledge points in the dataset are shown in Figure 12. In each subdiscipline, the distribution of cognitive labels is shown in Figure 13.

F Prompt Template Adopted by the FinEval-KR Evaluation Framework

Figure 15 shows the prompt template used in Stage 1 where the evaluated model answers the questions

Stage	Aspect	Criteria
Logical Validation	Clarity and Completeness Check	(1) Is the question description clear and unambiguous? (2) Is there any misuse of terminology? (3) Are all necessary conditions and data for calculation provided?
	Plausibility Check	(1) Are numerical values (e.g., interest rates, returns, prices) within a plausible range? (2) Is the scenario self-contradictory or unrealistic?
	Solvability Check	Assuming the data is complete and plausible, does the question have a deterministic solution that can be calculated using financial models?
Consistency Assessment	Relevance Check	Do the core concepts in the question align with the title or keywords of the corresponding textbook chapter?
Final Validation	Samples that fail any of the above checks are marked as "unqualified" and removed from the final dataset.	

Table 4: Validation in question generation stage.

Stages	Aspect	Criteria
Logical Validation	Formula/Model Selection Check	(1) Is the selected formula a standard method for this type of problem? (2) Does the variable required by the question match the variable solved for in the answer?
	Parameter Substitution Check	Do the numerical values from the question correctly correspond to the variables in the formula during calculation?
Consistency Assessment	Calculation Validation	Execute the calculation code output by OpenAI o1 to verify the answer's correctness.
Final Validation	Samples that fail any of the above checks are marked as "unqualified" and removed from the final dataset.	

Table 5: Validation in answer generation stage.

in a free-form format. Figure 16 shows the prompt template used in Stage 2, in which the evaluated model re-answer the question with knowledge point augmented. Figure 14 shows the prompt templates for the judge model to generate review results and augmented review results based on the reference answers in Stage 1 and 2.

G Bias Challenges in Judge Model

Previous research has shown that using LLMs as judges to evaluate the output of other models inevitably introduces certain evaluation biases, which may lead to unfair comparison results. Therefore, this section will discuss the main evaluation biases discovered during the experimental process of this study and their corresponding mitigation methods.

- *Style Bias*: This bias refers to the tendency of the judge LLM to give higher scores to content with a more appealing text style (e.g., clear structure, moderate length), even if the answers contain reasoning errors (Wu and Aji, 2025). To reduce the impact of this type of bias, we did not restrict the output format of the models being evaluated in Stage 1 and Stage 2, encouraging them to reason freely. Subsequently, we used methods such as regular expressions to unify the original output of each model into a format consistent with the reference answer. This processing method effectively reduced evaluation biases caused by differences in text style.

- *Cognitive Bias*: This bias refers to the self-bias that LLMs may exhibit during the evaluation, i.e., a tendency to give higher scores to content they generated themselves, thereby affecting the fairness of the evaluation (Koo et al., 2024). To avoid this type of cognitive bias, we excluded OpenAI o1 from the scope of evaluated models in our experiment, as it had already been used in the dataset construction and validation. Furthermore, during the preliminary experiment, we tested whether Qwen2.5-72B_Instruct, used as the judging model, exhibited significant cognitive bias in the root cause localization and knowledge gap identification tasks. The experimental results showed that this model did not exhibit a significant self-bias tendency in the aforementioned two tasks. We believe this may be because these two tasks are more specific and objective compared to result comparison tasks without sub-task decomposition, and are further aided by the model introspection prompting shown in Figure 14, which helps enhance the

问题:
一家英国跨国公司Oceanic Ltd在加拿大设有一家大型子公司。该子公司在2021年和2022年的当地货币（加元）收益分别为10,000,000加元和12,000,000加元。假设2021年加元对英镑的加权平均汇率为£0.58，加元价值相对稳定；而2022年由于汇率波动，加元对英镑的加权平均汇率下降至£0.52。根据国际会计准则IAS 21的规定，Oceanic Ltd在合并财务报表时需要将加拿大子公司的收益从其功能货币（加元）转换为报告货币（英镑）。
请计算Oceanic Ltd在2021年和2022年来自加拿大子公司的折算英镑收益，并讨论汇率变动是如何影响合并后财务报表中的收益的。

答案与推理过程
步骤一：计算2021年和2022年加拿大子公司的英镑收益。
- 2021年收益：CAD 10,000,000 x £0.58/CAD = £5,800,000
- 2022年收益：CAD 12,000,000 x £0.52/CAD = £6,240,000

步骤二：分析收益增长及汇率变动的影响。
- 加元收益增长率：((CAD 12,000,000 - CAD 10,000,000) / CAD 10,000,000) x 100% = 20%
- 英镑收益增长率：((£6,240,000 - £5,800,000) / £5,800,000) x 100% ≈ 7.59%

步骤三：评估汇率对英镑收益的影响。
- 假设汇率不变，2022年英镑收益应为：CAD 12,000,000 x £0.58/CAD = £6,960,000
- 实际与假设差额：£6,960,000 - £6,240,000 = £720,000

最终答案：由于汇率从£0.58/CAD下降至£0.52/CAD，导致英镑收益增长被削弱，从20%降至约7.59%，减幅为£720,000。这反映了汇率波动对财务报表产生的重要影响。

知识点
跨国公司，汇率折算，IAS 21，功能货币与报告货币，汇率波动影响

认知标签
步骤一：应用
步骤二：分析
步骤三：分析
最终答案：评估

```
Question
A British multinational corporation, Oceanic Ltd, has a large
subsidiary in Canada. This subsidiary's earnings in its local currency
(Canadian dollars) were CAD 10,000,000 in 2021 and CAD 12,000,000 in
2022. Assume the weighted average exchange rate of CAD to GBP was £0.58
in 2021, with the Canadian dollar's value being relatively stable. In
2022, due to exchange rate fluctuations, the weighted average exchange
rate of CAD to GBP decreased to £0.52. According to International
Accounting Standard (IAS) 21, Oceanic Ltd needs to translate the
Canadian subsidiary's earnings from its functional currency (CAD) to
its presentation currency (GBP) when preparing its consolidated
financial statements.
Please calculate Oceanic Ltd's translated GBP earnings from its
Canadian subsidiary for 2021 and 2022, and discuss how the exchange
rate changes affected the earnings in the consolidated financial
statements.

Answer and Reasoning Process:
Step 1: Calculate the GBP earnings of the Canadian subsidiary for 2021
and 2022.
- 2021 Earnings: CAD 10,000,000 x £0.58/CAD = £5,800,000
- 2022 Earnings: CAD 12,000,000 x £0.52/CAD = £6,240,000

Step 2: Analyze the earnings growth and the impact of exchange rate
changes.
- Canadian Dollar Earnings Growth Rate: ((CAD 12,000,000 - CAD
10,000,000) / CAD 10,000,000) x 100% = 20%
- Pound Sterling Earnings Growth Rate: ((£6,240,000 - £5,800,000) /
£5,800,000) x 100% ≈ 7.59%

Step 3: Evaluate the impact of the exchange rate on GBP earnings.
- Assuming the exchange rate remained constant, the 2022 GBP earnings
would have been: CAD 12,000,000 x £0.58/CAD = £6,960,000
- Difference between actual and hypothetical earnings: £6,960,000 -
£6,240,000 = £720,000

Final Answer: Due to the decrease in the exchange rate from £0.58/CAD
to £0.52/CAD, the growth in GBP earnings was weakened, falling from 20%
to approximately 7.59%, a reduction of £720,000. This reflects the
significant impact that exchange rate fluctuations can have on
financial statements.

Knowledge Points:
Multinational Corporation, Exchange Rate Translation, IAS 21,
Functional Currency and Presentation Currency, Impact of Exchange Rate
Fluctuations

Cognitive Tags:
Step 1: Apply
Step 2: Analyze
Step 3: Analyze
Final Answer: Evaluate
```

Figure 10: A complete sample from FinEval-KR dataset (original in Chinese, with English translation).

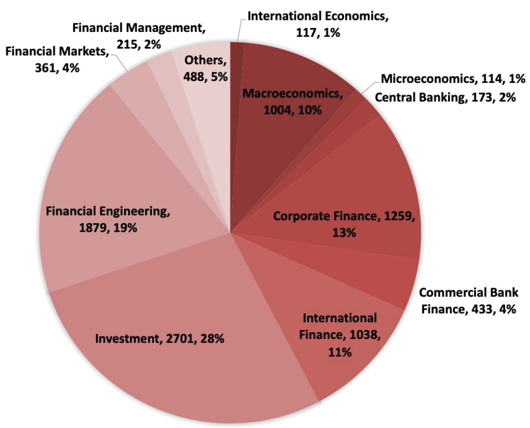


Figure 11: The number of samples in each subdiscipline in the FinEval-KR dataset and their percentage, and "others" in the pie chart includes: econometrics, public finance, insurance, monetary economics, managerial accounting, intermediate financial accounting, corporate strategy and risk management, auditing, cost accounting, taxation and advanced financial accounting.

objectivity of Qwen2.5-72B_Instruct during the evaluation.

- To prevent the judge model from the disturbance of "simple deception" (Thakur et al., 2024), we filter out meaningless content in the generated answers, such as isolated affirmative words like "yes" or "of course", ensuring that the evaluation focuses on the substantive reasoning process rather than superficial linguistic features.

H Details of Experiment

Evaluated Open-source Models We select several popular LLMs, including DeepSeek-R1, DeepSeek-V3, and QwQ-32B-preview. To study the effect of model size on performance, we also include smaller models, such as Qwen2.5-14B_Instruct. In total, this study includes four open-source LLMs.

Evaluated Close-source Models We include a selection of prominent models. This selection covers the latest reasoning models, such as Claude-3.7-sonnet, Gemini-2.5-flash, Gemini-2.5-pro, o3-mini, and o1-mini. We also include the current top non-

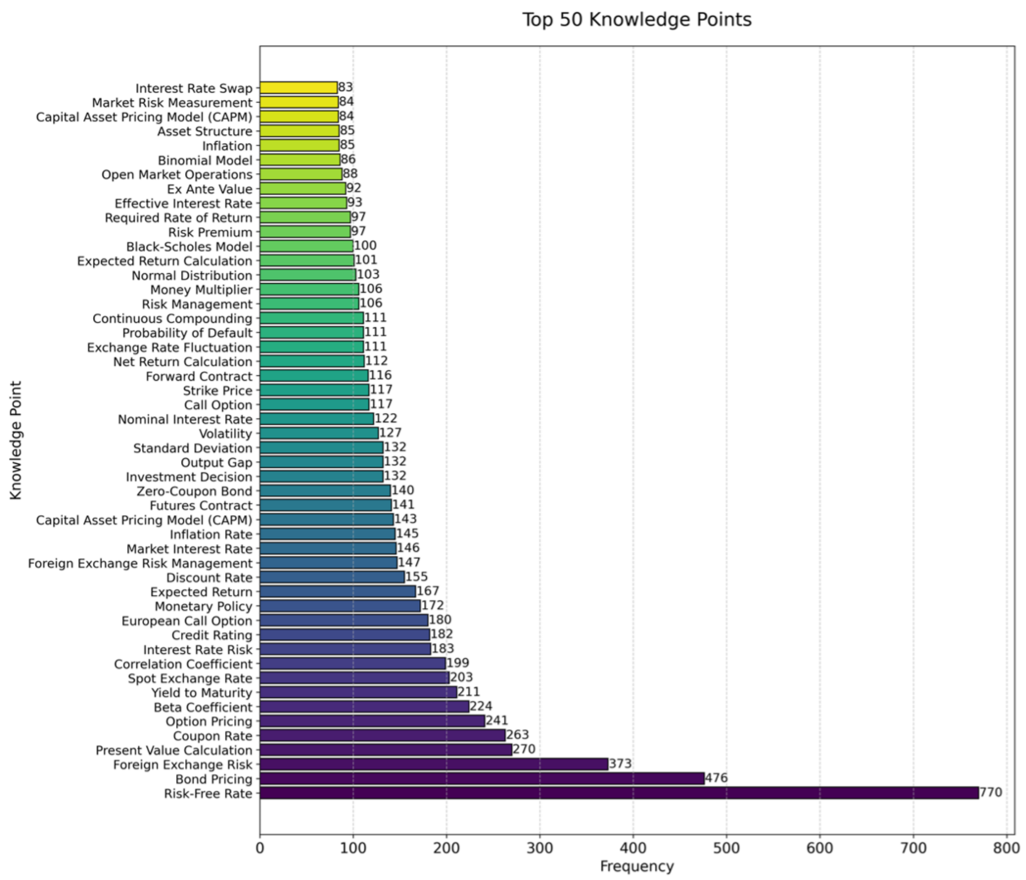


Figure 12: Top 50 knowledge points in the dataset.

reasoning models: GPT-4.1, GPT-4o, and Qwen-max-latest. Furthermore, we add some models released in 2024 and 2023, including Moonshot-v1-128k, Doubao-pro-32k, Ernie-Bot-4.0, and GPT-3.5-turbo.

Financial LLMs Furthermore, we specifically select two financial reasoning LLMs for evaluation. The first is Xuanyuan-FinX1-preview from Duxiaoman AI-Lab, a Chinese financial dialogue and reasoning model designed specifically for the financial domain. It is also the first o1-like model in the financial industry. The second is Fin-R1, a financial reasoning LLM jointly developed by Shanghai University of Finance and Economics and StepFun Technology. This model is trained on Qwen2.5-7B_Instruct and designed for complex financial reasoning tasks, balancing high performance with low deployment cost.

Implement During evaluation, all closed-source models are accessed through the official APIs provided by their respective developers. In contrast, open-source models are accessed using the service provided by either Bailian[3] or ModelScope[4].

Model	Version
Qwen2.5-14B_Instruct	2024-09-19
QwQ-32B-preview	2025-03-06
DeepSeek-V3	2025-03-24
DeepSeek-R1	2025-01-20
Doubao-pro-32k	2024-06-15
Moonshot-v1-128k	2024-01-31
Ernie-Bot-4.0	2023-11-17
Qwen-max-latest	2025-01-25
GPT-3.5-turbo	2024-01-25
GPT-4o	2024-11-20
GPT-4.1	2025-04-14
Gemini-2.5-pro	2025-05-06
Claude-3.7-sonnet	2025-02-19
o1-mini	2024-09-12
o3-mini	2025-01-31
Gemini-2.5-flash	2025-04-17
Xuanyuan-FinX1-preview	2024-12-27
Fin-R1	2025-03-22

Table 6: Version of the model being evaluated.

[3] https://bailian.console.aliyun.com
[4] https://www.modelscope.cn

Figure 13: The distribution of cognitive labels in each subdiscipline.

I Discussion of Experimental Results

We evaluate 18 LLMs listed in Appendix H using our proposed financial reasoning dataset. Table 7 presents the complete evaluation results across several metrics: Knowledge Score (KS), Reasoning Score (RS), Cognitive Scores (CS_1 to CS_5), and Task Accuracy (Acc). Results in the table are from three runs of each model. The CS_1 to CS_5 correspond to remembering, understanding, applying, analyzing, and evaluating in Bloom's taxonomy, respectively.

For all subsequent analyses, our focus is on performance tiers instead of absolute scores, which helps alleviate assessment errors caused by the systematic bias and randomness of the judge model. Additionally, we set the distribution of models across the tiers to 3:4:5:6.

I.1 Analysis of Knowledge Score

The KS measures the breadth of a LLM's knowledge coverage in the financial domain. Based on the evaluation results, models fall into four tiers, as Figure 17 shows.

Tier 1 is exclusively composed of closed-source models that exhibit exceptionally high financial

你是一位资深的金融学专家，现在需要批改{某子学科}题目的答案。接下来你会收到<题目>、<参考答案>以及<待批改的答案>。

批改需要按以下几点要求严格执行:
1. 你需要对比<待批改的答案>与<参考答案>，判断<待批改的答案>是否正确。
 - 如果<待批改的答案>错误，输出第一个错误推理步骤的错误原因到<错误原因>。
 - 如果<待批改的答案>正确，<错误原因>为空。
2. 在判断<定位>时，对<参考答案>进行一步步分析，仔细思考，判断<错误原因>与<参考答案>中哪一步最相似。
3. 在判断<欠缺的知识点>时，根据<错误原因>和<定位>，分析错误原因对应的知识点。
4. 在判断<结果>时，只需要判断<待批改的答案>最后的计算结果是否正确。若推理过程错误，但答案正确，也判断为"正确"。
5. 在你批改之前，首先严格按照<Inner Thoughts>进行内省，然后给出批改结果。

输出格式:
<Inner Thoughts>
1. 对比待批改的答案与参考答案
2. 识别错误
3. 定位错误来源
4. 确定欠缺的知识点
5. 判断最终结果
</Inner Thoughts>
<错误原因>
<待批改的答案>中，第一个错误推理步骤的错误原因
</错误原因>
<定位>
直接输出<参考答案>中与<错误原因>最相关的步骤
</定位>
<欠缺的知识点>
输出<错误原因>包含的概念、定义和公式
</欠缺的知识点>
<结果>
直接输出 "正确" 或者 "错误"
</结果>

请严格按照上述标签格式输出，不要添加额外的文字。

You are a senior finance expert and now you need to review the answers of {a subfield}. Then you will receive <question>, <reference answer>, and <answer to be reviewed>.

The following requirements should be applied strictly during reviewing:
1. You need to compare <answer to be reviewed> with <reference answer> to determine if <answer to be reviewed> is correct.
 - If <answer to be reviewed> is wrong, output the reason for the first wrong reasoning step to <cause of error>.
 - If <answer to be reviewed> is correct, <cause of error> left blank.
2. When determining <location>, analyze the <reference answer> step by step and think carefully to determine which step in <cause of error> is most similar to the one in <reference answer>.
3. When determining <missing knowledge>, analyze the knowledge point corresponding to the cause of the error based on <cause of error> and <location>.
4. When determining the <result>, you need only determine whether the final result of the <answer to be reviewed> is correct or not. If the reasoning process is wrong, but the answer is correct, it is also judged as "correct".
5. Before you review the answers, strictly follow <Inner Thoughts> for introspection and then give the review result.

Output Format:
<Inner Thoughts>
1. Comparing answer to be corrected with reference answer
2. Identify errors
3. Locate the cause of the error
4. Determine what knowledge points are missing
5. Analyze the final result
</Inner Thoughts>
<cause of error>
Causes for the error in the first incorrect reasoning step in <answer to be reviewed>
</cause of error>
<location>
Directly output the steps in <reference Answer> that are most relevant to the <cause of error>
</location>
<missing knowledge>
Output concepts, definitions and formulas involved in the <cause of error>
</missing knowledge>
<result>
Direct output "correct" or "incorrect"
</result>

Please format the output strictly according to the above tags and do not add additional text.

Figure 14: Prompt templates for the judge model (original in Chinese, with English translation).

你是一名金融学专家，你需要用中文解答以下题目，并给出完整的解题过程。
You are a financial expert, and you need to answer the following question in Chinese and provide the complete solution process.
请一步一步想，解题过程中仅考虑题目中的情景，不要额外补充信息。
Please think step-by-step, and only consider the scenario described in the problem during the solution process; do not add extra information.

题目: {给定的金融推理题目}
Question: {Given Financial Reasoning Problem}

Figure 15: Prompt template for Stage 1 where the evaluated model answers the questions in a free-form format (original in Chinese, with English translation).

你是一名金融学专家，你需要用中文解答以下题目，并给出完整的解题过程。
You are a financial expert, and you need to answer the following question in Chinese and provide the complete solution process.
请一步一步想，**充分考虑并正确运用给定的知识点**，解题过程中仅考虑题目中的情景，不要额外补充信息。
Please think step-by-step, **fully consider and correctly apply the given tips**, and only consider the scenario described in the problem during the solution process; do not add extra information.

题目: {给定的金融推理题目}
Question: {Given financial reasoning problem}
提示: {review result 中分析出可能欠缺的知识点}
Tips: {Knowledge point identified from the review result that might be lacking}

Figure 16: Prompt template for Stage 2 where the evaluated model re-answer the question with knowledge point augmented (original in Chinese, with English translation).

knowledge coverage. Tier 2 is dominated by the top-performing open-source models. Although they rank just below Tier 1, the absolute score difference is marginal, indicating that their financial knowledge coverage is nearly on par with the leading closed-source models.

A significant performance drop-off occurs in the lower tiers. In Tiers 3 and 4, the older GPT-3.5-turbo notably outperforms other models within this bracket. At the bottom of the ranking is the specialized financial model, Fin-R1-7B, whose lower performance is primarily attributed to its significantly smaller parameter scale.

In summary, leading closed-source and top open-source reasoning models demonstrate the strongest performance in financial knowledge coverage, which is significantly influenced by model scale. While financial knowledge is a mature capability in most mainstream LLMs and no longer the primary differentiator among top models, it remains a fundamental prerequisite for high-quality reasoning.

Model/Metrics	Acc	Acc.std	KS	KS.std	RS	RS.std	CS_1 (remember)	CS_1.std	CS_2 (understand)	CS_2.std	CS_3 (apply)	CS_3.std	CS_4 (analyze)	CS_4.std	CS_5 (evaluate)	CS_5.std
Open-source lightweight LLMs without reasoning																
Qwen2.5-14B_Instruct	0.5473	0.0006	0.8490	0.0010	0.6863	0.0038	0.6547	0.0015	0.6603	0.0023	0.3893	0.0064	0.6863	0.0038	0.6820	0.0046
Open-source lightweight LLMs with reasoning																
QwQ-32B-preview	0.7380	0.0061	0.9073	0.0057	0.8627	0.0136	0.8450	0.0141	0.8503	0.0143	0.6987	0.0267	0.8510	0.0075	0.8597	0.0136
Open-source LLMs without reasoning																
DeepSeek-v3	0.8270	0.0062	0.9427	0.0050	0.9077	0.0050	0.8963	0.0059	0.8993	0.0059	0.7963	0.0125	0.8943	0.0075	0.9057	0.0057
Open-source LLMs with reasoning																
DeepSeek-R1	0.8700	0.0165	0.9517	0.0171	**0.9347**	0.0153	**0.9377**	0.0186	**0.9397**	0.0179	**0.8810**	0.0358	**0.9380**	0.0190	**0.9433**	0.0158
Closed-source LLMs without reasoning																
Doubao-pro-32k	0.7825	0.0007	0.9195	0.0007	0.8750	0.0057	0.8560	0.0071	0.8600	0.0085	0.7340	0.0113	0.8565	0.0064	0.8720	0.0057
Moonshot-v1-128k	0.4533	0.0015	0.8340	0.0061	0.6020	0.0082	0.5620	0.0108	0.5670	0.0087	0.2763	0.0064	0.5653	0.0074	0.5973	0.0074
Ernie-bot-4.0	0.5733	0.0025	0.8627	0.0081	0.7053	0.0091	0.6680	0.0089	0.6753	0.0091	0.4383	0.0146	0.6847	0.0074	0.6927	0.0251
Qwen-max-latest	0.6467	0.0015	0.8797	0.0050	0.7733	0.0042	0.7507	0.0050	0.7547	0.0057	0.5340	0.0040	0.7440	0.0026	0.7703	0.0042
GPT-3.5-turbo	0.2830	0.0040	0.7527	0.0038	0.3973	0.0040	0.3527	0.0021	0.3603	0.0021	0.0900	0.0036	0.3893	0.0081	0.3970	0.0036
GPT-4o	0.6853	0.0142	0.9020	0.0159	0.8067	0.0080	0.7847	0.0081	0.7890	0.0090	0.5930	0.0145	0.7870	0.0110	0.8030	0.0085
GPT-4.1	0.8263	0.0025	0.9520	0.0040	0.9063	0.0015	0.8957	0.0021	0.8977	0.0025	0.7890	0.0036	0.8927	0.0015	0.9050	0.0017
Closed-source LLMs with reasoning																
o1-mini	0.7503	0.0031	0.8997	0.0076	0.8453	0.0067	0.8340	0.0066	0.8363	0.0031	0.6983	0.0081	0.8477	0.0070	0.8450	0.0060
o3-mini	0.8207	0.0095	0.9260	0.0106	0.9070	0.0052	0.9047	0.0102	0.9073	0.0099	0.8127	0.0110	0.9023	0.0107	0.9120	0.0113
Gemini-2.5-pro	**0.8750**	0.0079	**0.9627**	0.0134	0.9233	0.0238	0.9123	0.0272	0.9163	0.0290	0.8403	0.0291	0.9050	0.0260	0.9120	0.0243
Gemini-2.5-flash	0.8440	0.0061	0.9540	0.0020	0.9203	0.0091	0.9103	0.0100	0.9133	0.0108	0.8307	0.0061	0.9100	0.0104	0.9177	0.0110
Claude-3.7-sonnet	0.7923	0.0040	0.9390	0.0030	0.8823	0.0086	0.8663	0.0100	0.8703	0.0096	0.7433	0.0120	0.8653	0.0093	0.8803	0.0086
Financial LLMs with reasoning																
Xuanyuan-FinX1-preview	0.5890	0.0026	0.8687	0.0032	0.7323	0.0042	0.7063	0.0032	0.7130	0.0044	0.4610	0.0066	0.7323	0.0042	0.7300	0.0035
Fin-R1-7B	0.4153	0.0031	0.7510	0.0346	0.5570	0.0040	0.5190	0.0046	0.5277	0.0065	0.2170	0.0036	0.5570	0.0040	0.5527	0.0045

Table 7: The complete evaluation results across several metrics: Knowledge Score (KS), Reasoning Score (RS), Cognitive Scores (CS_1 to CS_5), and Task Accuracy (Acc).

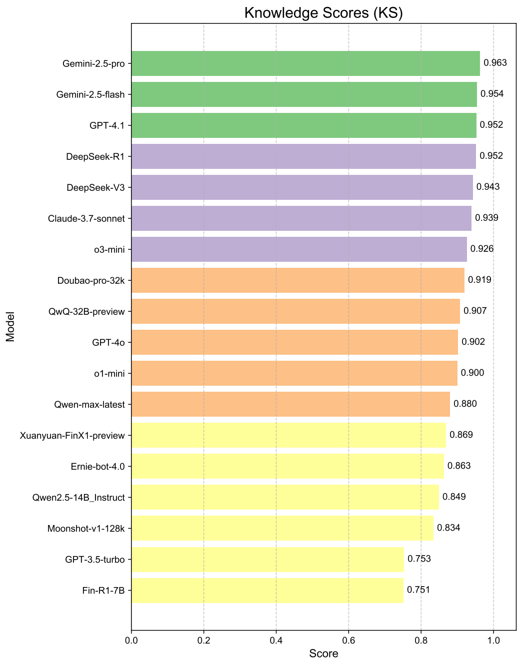


Figure 17: The knowledge score of the models.

I.2 Analysis of Reasoning Score

The RS is inversely related to the proportion of failures caused by incorrect reasoning steps. It reflects a model's reasoning ability. Based on this metric, the evaluated models are also divided into four tiers, as Figure 18 shows.

Tier 1 represents the pinnacle of performance, comprising reasoning-optimized models that demonstrate outstanding accuracy and logical completeness. Tier 2 includes some high-performing, non-reasoning models like GPT-4.1 and DeepSeek-V3. A significant performance gap separates the top two tiers from the bottom two. This clear stratification underscores the need for future model development to prioritize the design and optimization of the reasoning pipeline, which is crucial for enhancing the reliability and stability of complex reasoning tasks.

I.3 Analysis of Cognitive Score

The CS provides a systematic evaluation of models' cognitive abilities based on Bloom's Taxonomy, across five dimensions, that is remembering (CS_1), understanding (CS_2), applying (CS_3), analyzing (CS_4), and evaluating (CS_5). As Table 7 shows, CS scores generally exhibit a positive correlation with the KS and the RS.

While most models achieve high scores at lower cognitive levels (CS_1: Remembering, CS_2: Understanding), their performance diverges significantly on higher-order tasks. These more demanding abilities—Applying (CS_3), Analyzing (CS_4), and Evaluating (CS_5)—reveal clear distinctions among the models. Consequently, our analysis focuses on these three dimensions. We define a primary metric, CS_{avg}, as the average score across these higher-order skills, and stratify the models into four performance tiers based on this metric (see Figure 19).

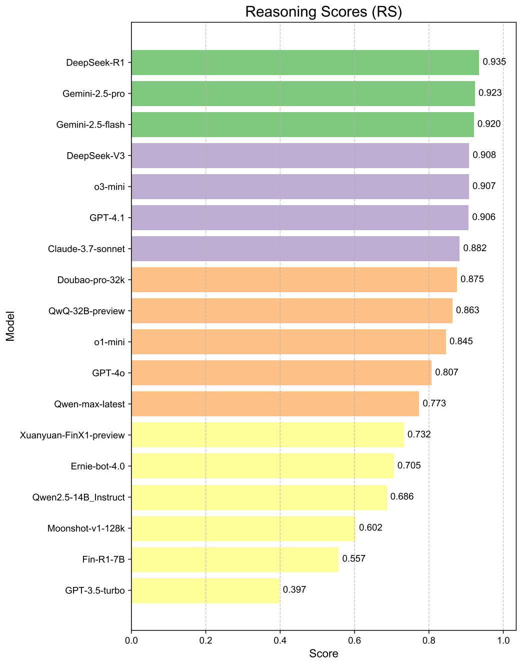


Figure 18: The reasoning score of the models.

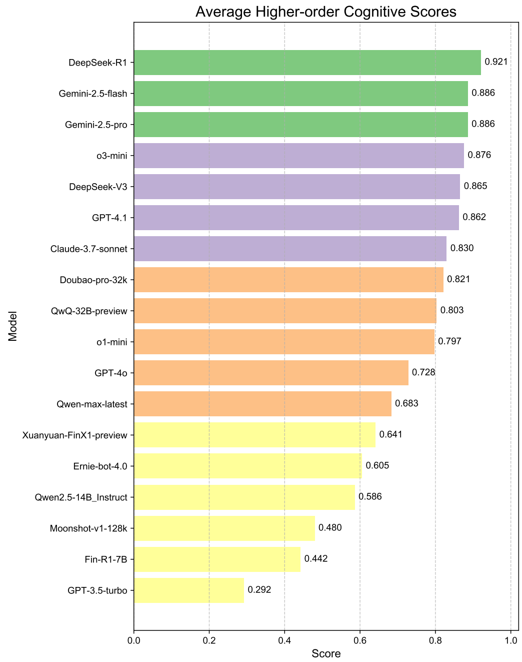


Figure 19: The average higher-order cognitive scores (CS_3 to CS_5) for the model.

Tier 1 models excel across all cognitive levels, demonstrating a distinct advantage in higher-order abilities. This tier is led by DeepSeek-R1, followed by Gemini-2.5-flash and pro. Tier 2 models also exhibit strong higher-order cognitive skills, with performance slightly below that of Tier 1. This tier includes most general-purpose reasoning models as well as the top-performing non-reasoning models, DeepSeek-V3 and GPT-4.1. Tiers 3 and 4 primarily consist of non-reasoning or smaller-scale models.

I.4 Analysis of Task Accuracy

Task Accuracy measures a model's direct success rate in executing reasoning tasks. Achieving high accuracy requires a synthesis of a broad knowledge base, robust reasoning capabilities, and advanced cognitive skills—particularly in application and analysis. Consequently, the performance gradient observed in Task Accuracy closely mirrors those of the RS and CS. The tiers of models based on this metric are shown in Figure 20.

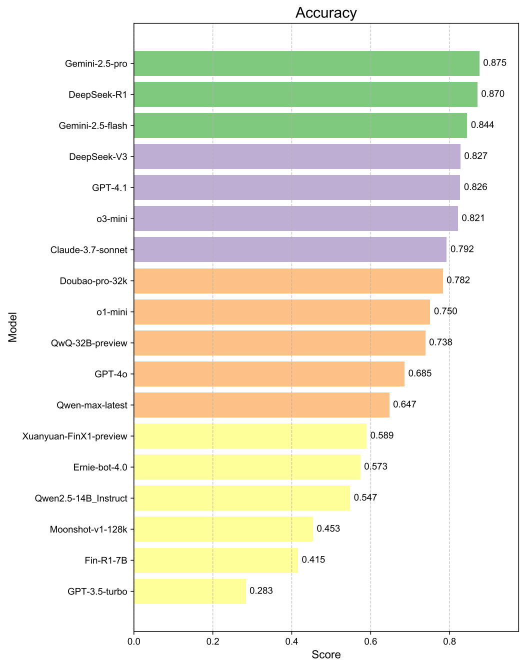


Figure 20: The accuracy of the models.

I.5 Variance Analysis

We evaluate a model's performance stability by the standard deviation of its scores across multiple test runs. We classify stability into two categories: *High Stability* (a standard deviation on the order of 10^{-4} to 10^{-3}), indicating highly consistent and reproducible outputs, and *Low Stability* (an order of 10^{-2} to 10^{-1}), which suggests significant performance fluctuations.

Our key findings are as follows:

- **Knowledge Retrieval is More Stable Than Reasoning.** For most models, the KS is consistently more stable than the RS. This is intuitive, as retrieving a stored fact is a more deterministic process for a well-trained model than performing a complex, multi-step logical deduction, which allows for greater variability.

- **GPT-4o is a Unique Outlier.** GPT-4o defies the general trend. Its reasoning process is remarkably stable, with an RS standard deviation of 8×10^{-3}, which is significantly more stable than its knowledge retrieval (KS standard deviation of 2×10^{-2}). We hypothesize that GPT-4o may possess a highly consistent, almost programmatic reasoning structure, while its knowledge function exhibits greater variance to adapt to diverse queries. This unusual stability profile warrants further investigation.

J Details of Human Evaluators and Validation Process

J.1 Evaluator Qualifications and Number

A total of 30 human experts participated in our validation effort. All experts are postgraduate students with academic backgrounds in finance, economics, or statistics, ensuring they possess an accurate understanding of the relevant professional terminology, fundamental concepts, and practical scenarios.

The entire validation process was conducted on a professional annotation platform provided by a leading technology company to ensure procedural standardization and data security.

J.2 Quality Control Mechanism

To guarantee the reliability of our validation results, we implemented a multi-stage quality control process. First, we randomly sampled 10% of the dataset. Each sample was independently validated by 2 experts to ensure consistency. Following this cross-validation, we organized a team of 3 senior experts to conduct a final quality check on a random 10% of the already-validated sample (amounting to a final check on 1% of the total dataset). This final step was designed to ensure the quality and uniformity of the standards applied during the cross-validation stage.

J.3 The Validation Process

The experts' validation work was divided into three strict, sequential stages:

Stage 1: Question and Knowledge Point Validation In this initial stage, experts were only shown the question and its associated knowledge points. They were tasked with the following checks:

- **Question Validity**: Is the question relevant to a realistic financial scenario? Is professional terminology used correctly? Are the numerical values within a reasonable range? Does the question have a single, definitive answer?

- **Knowledge Point Relevance**: Are the tagged knowledge points accurate and comprehensive? Is the naming of the knowledge points consistent with standard terminology in mainstream textbooks?

Independent Answering After confirming the quality of the question, experts were required to solve the problem independently, without reference to any provided solution. The goal of this step was to obtain a high-quality, unbiased human answer to serve as a benchmark for subsequent comparisons.

Stage 3: Reasoning Steps and Cognitive Labels Validation Finally, the system presented the experts with the answer, the step-by-step solution, and the Bloom's Taxonomy cognitive label from our dataset. The experts were required to perform the following checks:

- **Answer and Solution Process Verification**: First, they compared their own answer to the one in the dataset. If the answers did not match, the sample was immediately flagged as "unqualified". If the answers matched, they proceeded to meticulously review the solution steps provided in the dataset, assessing whether the logic was clear, the steps were reasonable, and the calculations were correct.

- Cognitive Label Accuracy Check: Based on the predefined verb list corresponding to each cognitive level (as defined in Figure 9), the experts had to judge whether the cognitive label assigned to the question was accurate.

K License and Usage Constraints

The released dataset is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

The dataset is only for for evaluating LLMs in non-commercial academic research. The dataset is explicitly not authorized for use in training or fine-tuning machine learning models , including pre-training, instruction-tuning, or reinforcement learning stages. Access conditions ensure that all derived data products remain confined to research contexts, with no transfer or application permitted in industrial, governmental, or other operational domains.

L AI Assistants Usage Disclosure

This study did not employ any AI assistants during the research design, data analysis, or coding phases. During manuscript preparation, the authors exclusively utilized Google Gemini for grammatical refinement and stylistic polishing. No AI-generated content was incorporated into the methodology and results, ensuring the work's originality and human-driven intellectual integrity.

Towards Efficient FinBERT via Quantization and Coreset for Financial Sentiment Analysis

Avinash Kumar Sharma, Aisha Hamad Hassan, Tushar Shinde
Indian Institute of Technology Madras, Zanzibar

Abstract

Real-time financial sentiment classification from social media is critical for applications in algorithmic trading, risk assessment, and market surveillance. However, deploying large-scale models like FinBERT on edge devices remains impractical due to their high memory and compute demands. Meanwhile, financial text poses unique challenges such as class imbalance, noisy syntax, and temporal drift. We propose a unified framework that jointly applies coreset selection and post-training quantization to achieve scalable and efficient financial NLP. Our method reduces training data by up to 90% through coreset selection and compresses model size by up to $4\times$ via 8-bit quantization, while preserving over 90% of the original classification accuracy on benchmark financial sentiment datasets. This demonstrates the viability of deploying domain-specific NLP models in constrained environments, offering a principled solution for low-latency, resource-efficient financial text processing.

1 Introduction and Related Work

Financial sentiment analysis is a cornerstone of modern quantitative finance, enabling predictive insights for algorithmic trading, risk modeling, and market surveillance (Smailović et al., 2014; Cortis et al., 2017; Du et al., 2024). Unlike general sentiment tasks, financial sentiment exhibits domain-specific characteristics: formalized language, market-sensitive expressions, and nonlinear or latent connections to asset prices. As such, this task poses unique linguistic and deployment challenges.

We highlight three core bottlenecks that hinder scalable financial sentiment classification. First, the task is challenged by domain specificity and the need for interpretability. Financial sentiment is often subtle, expressed through technical vocabulary and domain-specific idioms. As a result, model predictions must not only be accurate but also interpretable to satisfy both regulatory standards and institutional trust requirements (Wang et al., 2025). Second, deployment environments often impose stringent resource constraints. Financial sentiment models are expected to operate in latency-sensitive and memory-limited contexts such as mobile trading applications or edge-based market monitoring systems, where large-scale transformer models become impractical (Shinde et al., 2025). Third, the availability of high-quality labeled data is limited. Financial text corpora are typically sparse, exhibit temporal non-stationarity, and are expensive to annotate, which restricts the scalability of supervised learning approaches (Wang et al., 2025).

Domain Adaptation in Financial NLP. To address linguistic specificity, domain-adapted language models such as FinBERT (Araci, 2019) and BloombergGPT (Wu et al., 2024) have emerged. These models are pre-trained on financial corpora, achieving superior performance on downstream tasks including entity recognition, sentiment tagging, and financial QA (Yang et al., 2020). However, their training and fine-tuning require substantial computational resources and data access, making them inaccessible to many institutions.

Model Compression Techniques. Model compression offers a viable route to scalable deployment. Quantization reduces memory and compute requirements by constraining weights and activations to lower-bit representations (Jacob et al., 2018; Shinde, a; Shinde and Tukaram Naik, 2024), while pruning removes redundant parameters with minimal accuracy loss (Han et al., 2015). Let $W \in R^{d\times h}$ be the weight matrix of a transformer layer; quantization maps W to a lower precision space $\hat{W} = Q(W)$ such that:

$$\hat{W}_{ij} = \text{round}\left(\frac{W_{ij} - \mu}{\Delta}\right)\Delta + \mu, \tag{1}$$

where Δ is the quantization step. Recent work in-

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

corporates mixed-precision and layer-wise adaptive schemes for optimal compression without quality degradation (Sun et al., 2022; Kuzmin et al., 2023).

Coreset Selection for Data Efficiency. To mitigate data scarcity, coreset selection identifies informative training subsets that retain performance while reducing training cost. Margin-based (Sener and Savarese, 2017), information-theoretic (Bachem et al., 2017), and forgetting-based strategies (Toneva et al., 2018; Shinde, b) prioritize samples that influence decision boundaries. Given training data $\mathcal{D}$, the coreset $\mathcal{C} \subset \mathcal{D}$ is chosen such that:

$$E_{(x,y)\in\mathcal{D}}\left[\ell(f_\theta(x), y)\right] \approx E_{(x,y)\in\mathcal{C}}\left[\ell(f_\theta(x), y)\right] \tag{2}$$

where ℓ is the task loss and f_θ is the model. Recent advances in zero-shot or training-free adaptation offer alternatives to full fine-tuning. Proxy tuning (Liu et al., 2024) enables logit-space adaptation by aligning decision boundaries between expert and base models. This is especially useful in financial domains, where annotation is costly and model update cycles must be rapid. Such methods enable lightweight personalization without retraining. While model compression and data-efficient learning have each advanced separately, their synergy in financial NLP remains underexplored. This work introduces a unified framework that combines coreset selection with quantization for real-time, resource-aware financial sentiment classification.

2 Methodology

This section presents our integrated framework for efficient financial sentiment classification using the Twitter Financial News Sentiment dataset. The framework combines coreset selection and quantization-based model compression to address both computational cost and memory footprint in resource-constrained scenarios, such as mobile and edge deployment environments.

2.1 Framework Overview

Our framework comprises three sequential components: class-balancing preprocessing, coreset-based sample selection, and adaptive quantization-aware model compression. Initially, we mitigate class imbalance inherent in financial sentiment data by preserving all samples from minority classes: Bearish (1,442) and Bullish (1,923), and reducing the majority Neutral class to match the size of the largest minority class, resulting in a balanced dataset with 5,288 training samples distributed equally across the three sentiment classes. This ensures unbiased model training while retaining representative information from all categories.

Subsequently, we employ coreset selection to extract informative subsets from the balanced dataset, evaluating coreset fractions in the set $\{1.0, 0.5, 0.25, 0.1, 0.05\}$. These fractions are used to create reduced yet representative training sets, thereby enabling systematic analysis of data efficiency and computational cost reduction.

2.2 Fine-tuning Procedure

We fine-tune the pretrained FinBERT model on the selected coreset using hyperparameters optimized for the financial social media domain. Validation is conducted at the end of each epoch to monitor overfitting and performance stability.

2.3 Quantization-Aware Model Compression

To facilitate efficient inference, we compress the fine-tuned FinBERT model through post-training quantization. We evaluate uniform symmetric quantization across bit-widths $b \in \{8, 7, 6, 5, 4, 3, 2, 1\}$ to study the trade-off between model size, accuracy, and compute efficiency. For a given weight tensor w_i in layer i, we compute the quantization scale as:

$$\text{scale}_i = \frac{\max(w_i) - \min(w_i)}{2^b - 1} \tag{3}$$

The weights are quantized using the following transformation:

$$w_i^{(q)} = \text{round}\left(\frac{w_i - \min(w_i)}{\text{scale}_i}\right) \cdot \text{scale}_i + \min(w_i) \tag{4}$$

This linear mapping ensures that weights are projected into a discrete set of 2^b values, reducing memory requirements and enabling faster inference on low-power devices. We evaluate performance degradation due to quantization at each bit level, measuring accuracy, precision, recall, and F1-score on the test set.

2.4 Joint Evaluation with Coreset Selection

We perform a comprehensive evaluation of the combined effects of coreset size and quantization bit-width. For each coreset fraction, we apply quantization at all bit-widths from 8 to 1, creating a grid of models. Each model is evaluated for classification performance and compression efficiency. This systematic design allows us to jointly analyze

Table 1: Distribution of sentiment classes across train, validation, and test sets.

Class	Train	Validation	Test
Bullish	1,670	358	358
Bearish	1,253	269	269
Neutral	5,118	1,097	1,097

the impact of training data reduction and quantization granularity, providing insights into the optimal trade-off between accuracy and computational efficiency for real-world financial NLP deployment.

3 Experimental Setup

Dataset Description. We conduct our experiments using the publicly available Twitter Financial News Sentiment dataset (Zeroshot, 2022), which consists of 11,932 annotated tweets related to financial news and market discourse. Each tweet is categorized into one of three sentiment classes: Bullish (positive market outlook), Bearish (negative market outlook), and Neutral (no clear directional sentiment). The dataset exhibits a pronounced class imbalance, with approximately 65% of the samples labeled as Neutral, 20% as Bullish, and 15% as Bearish.

To ensure robust model learning and unbiased evaluation, we adopt a stratified data split strategy, reserving 70% of the dataset for training, 15% for validation, and 15% for testing. The class distributions are preserved across the splits. Table 1 summarizes the class distribution in each subset.

Experimental Configuration. All models were implemented using the PyTorch framework, and experiments were conducted on the Kaggle cloud platform equipped with an NVIDIA Tesla P100 GPU (16GB). To ensure reproducibility, we fixed the random seed across runs and used deterministic training settings wherever supported.

We fine-tuned the pretrained `FinBERT` model using the AdamW optimizer with a learning rate of 2×10^{-5}, batch size of 16, and linear learning rate warmup. Each model was trained for 3 epochs. Early stopping based on validation F1-score was used to prevent overfitting. Quantization and compression experiments were performed post-training.

Evaluation Protocol. We evaluate the models using standard classification metrics: Accuracy, Precision, Recall, and F1-score. In addition to classification performance, we measure the model's memory efficiency using the Compression Ratio (CR):

Table 2: Fine-tuning progression on balanced validation set.

Stage	Accuracy	F1 (Macro)	F1 (Weighted)
Pre-trained Baseline	0.326	0.284	0.309
Epoch 1	0.594	0.549	0.577
Epoch 2	0.773	0.765	0.772
Epoch 3 (Final)	0.834	0.833	0.835

$$\text{CR} = \frac{\text{Original Model Size (in MB)}}{\text{Compressed Model Size (in MB)}} \tag{5}$$

This metric quantifies the storage reduction achieved by quantization-based model compression. We also report inference time per batch to assess real-time deployment feasibility.

4 Results and Analysis

4.1 Fine-tuning Performance Analysis

We first evaluate the impact of domain-specific fine-tuning on the pre-trained `FinBERT` model using the balanced validation set. As shown in Table 2, accuracy increases from 32.6% (pre-trained) to 83.4% after three epochs of fine-tuning. This represents an absolute improvement of +50.8 percentage points, accompanied by similar gains in both macro and weighted F1-scores. These results confirm the substantial benefit of adapting language models to domain-specific financial discourse, particularly in the context of noisy, sentiment-rich social media text.

4.2 Coreset Selection Efficiency

We next assess the effect of coreset selection by training on reduced fractions $\{1.0, 0.5, 0.25, 0.1, 0.05\}$ of the full balanced training dataset. Table 3 reports performance metrics and training speedups. Remarkably, using only 10% of training data retains 90% of the full-model accuracy (75.1% vs. 83.4%) while yielding an 8.3× reduction in training time. This highlights the potential of representative subset selection in high-dimensional, redundant financial language datasets.

4.3 Quantization Performance

We evaluate post-training quantization of the fine-tuned model across bit-widths $b \in \{8, 7, ..., 1\}$. Table 4 summarizes the accuracy, macro F1-score, and corresponding compression ratios. Notably, 6-bit quantization maintains 97.4% of the full-precision accuracy (81.2% vs. 83.4%) with a 1.3×

Table 3: Coreset selection performance across data fractions.

Fraction	Samples	Accuracy	F1 (Macro)	Speedup
100%	1,297	0.834	0.833	1.0×
50%	648	0.825	0.822	1.9×
25%	324	0.798	0.795	3.6×
10%	129	0.751	0.748	8.3×
5%	64	0.687	0.684	14.3×

Table 4: Quantization impact on model accuracy and compression. CR - Compression Ratio

Bit-Width	Accuracy	F1 (Macro)	CR
8-bit	0.834	0.833	1.0×
7-bit	0.829	0.826	1.1×
6-bit	0.812	0.809	1.3×
5-bit	0.785	0.782	1.6×
4-bit	0.743	0.740	2.0×
3-bit	0.687	0.684	2.7×
2-bit	0.542	0.539	4.0×
1-bit	0.334	0.331	8.0×

model size reduction, representing an effective trade-off between model compactness and predictive performance.

4.4 Integrated Efficiency Trade-offs

We investigate the combined effect of coreset selection and quantization to identify optimal operating points for deployment. Table 5 presents the resulting accuracy and efficiency gain for selected configurations. The results suggest a Pareto frontier: configurations offering strong accuracy-efficiency trade-offs for specific deployment scenarios such as mobile inference or low-latency market monitoring.

5 Discussion

Data Efficiency in Financial NLP. Our findings indicate that financial sentiment classification benefits significantly from intelligent data reduction. As little as 10% of the original training data achieves over 75% accuracy, supporting the hypothesis that social-financial discourse contains high levels of redundancy. This has practical implications for reducing annotation costs, accelerating model development cycles, and supporting rapid model deployment in emerging financial events.

Quantization for Deployment-Grade Models. Among the evaluated bit-widths, 6-bit quantization offers a desirable trade-off, preserving over 97% of full-precision model performance. This balance is crucial in financial contexts, where inference latency and memory constraints are critical, yet even minor accuracy degradation may result in measurable trading loss or poor risk signal estimation.

Table 5: Combined coreset selection and quantization results.

Configuration	Fraction	Bit-Width	Accuracy	Efficiency Gain
Baseline	100%	8	0.834	1.0×
High Efficiency	25%	6	0.776	3.1×
Balanced	50%	6	0.809	2.6×
Quality Focused	100%	6	0.812	1.3×
Maximum Compression	10%	4	0.658	5.0×

Real-World Deployment Implications. The proposed integrated framework offers significant advantages for a range of financial deployment scenarios. In mobile trading applications, low-latency sentiment inference is critical for responsive user experience, while in edge-based market monitoring systems, the reduced model size alleviates bandwidth and storage constraints. High-frequency trading (HFT) systems can benefit from real-time sentiment feeds with minimized inference delay, ensuring rapid decision-making under stringent timing requirements. Additionally, cloud-based financial analytics platforms can leverage the compressed models to lower infrastructure costs while maintaining robust sentiment classification capabilities. These deployment contexts all benefit from the dual advantage of reduced memory footprint and faster inference, without a substantial loss in accuracy.

Limitations. This study is focused on Twitter-based financial sentiment. Extension to other modalities, such as earnings call transcripts, SEC filings, or institutional reports, remains to be explored. Furthermore, quantization is simulated in software; actual deployment on hardware (e.g., INT4 inference on FPGAs or mobile NPUs) may exhibit different characteristics. Finally, we do not currently evaluate fairness or robustness under domain shift, which are important concerns in financial NLP.

6 Conclusion and Future Work

We propose an efficient framework for financial sentiment analysis that combines coreset selection with systematic model quantization. Our experiments show that training on only 10% of the data selected via coreset methods preserves approximately 90% of the original model's accuracy. Coupled with 6-bit quantization, this yields a model that is significantly smaller and faster, yet remains competitive in classification performance. The approach reduces training data by up to 90% and achieves about a $4\times$ compression factor in model size, while maintaining accuracy within 10% of the full-data, full-precision FinBERT baseline on the Twitter Financial News Sentiment dataset. This demonstrates the practical applicability of transformer models in latency-critical financial settings such as real-time trading and mobile applications.

Future work will explore extending this framework to other financial text sources, and integrating additional compression techniques like structured pruning, low-rank factorization, and knowledge distillation.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Olivier Bachem, Mario Lucic, and Andreas Krause. 2017. Practical coreset constructions for machine learning. *arXiv preprint arXiv:1703.06476*.

Keith Cortis, André Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. 2017. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 519–535.

Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. 2024. Financial sentiment analysis: Techniques and applications. *ACM Computing Surveys*, 56(9):1–42.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. 2018. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713.

Andrey Kuzmin, Markus Nagel, Mart Van Baalen, Arash Behboodi, and Tijmen Blankevoort. 2023. Pruning vs quantization: Which is better? *Advances in neural information processing systems*, 36:62414–62427.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Tushar Shinde. a. Adaptive quantization and pruning of deep neural networks via layer importance estimation. In *Workshop on Machine Learning and Compression, NeurIPS 2024*.

Tushar Shinde. b. High-performance lightweight vision models for land cover classification with coresets and compression. In *TerraBytes-ICML 2025 workshop*.

Tushar Shinde, Avinash Kumar Sharma, Shivam Bhardwaj, and Ahmed Silima Vuai. 2025. Navigating coreset selection and model compression for efficient maritime image classification. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1608–1616.

Tushar Shinde and Sukanya Tukaram Naik. 2024. Adaptive quantization of deep neural networks via layer importance estimation. In *International Conference on Computer Vision and Image Processing*, pages 220–233. Springer.

Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič. 2014. Stream-based active learning for sentiment analysis in the financial domain. *Information sciences*, 285:181–203.

Mengshu Sun, Zhengang Li, Alec Lu, Yanyu Li, Sung-En Chang, Xiaolong Ma, Xue Lin, and Zhenman Fang. 2022. Film-qnn: Efficient fpga acceleration of deep neural networks with intra-layer, mixed-precision quantization. In *Proceedings of the 2022 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 134–145.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*.

Yuxiang Wang, Yuchi Wang, Yi Liu, Ruihan Bao, Keiko Harimoto, and Xu Sun. 2025. Proxy tuning for financial sentiment analysis: Overcoming data scarcity and computational barriers. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 169–174.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2024. Bloomberggpt: A large language model for finance, 2023. *URL https://arxiv. org/abs/2303.17564*.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Zeroshot. 2022. Twitter financial news sentiment. https://huggingface.co/datasets/zeroshot/twitter-financial-news-sentiment. Dataset available on Hugging Face.

LAVA: Logic-Aware Validation and Augmentation Framework for Large-Scale Financial Document Auditing

Ruoqi Shu[*†], **Xuhui Wang**[*†], **Isaac Wang**[‡], **Yanming Mai**, **Bo Wan**

BMO Financial Group
ruoqi.shu@unisystechconsulting.com, xuhuieric.wang@bmo.com,
{isaac.wang, yanming.mai, bo.wan}@bmo.com

Abstract

Financial document validation in production—such as payroll auditing, tax compliance, and loan underwriting—demands exceptional accuracy, consistency, and reproducibility under strict enterprise constraints. In practice, documents arrive with heterogeneous layouts and formats, semantically rich, context-dependent content, and embedded business rules that current pipelines struggle to process reliably. We introduce LAVA (Logic-Aware Validation and Augmentation)—a modular, backbone-agnostic pipeline built on multimodal large language models—that integrates a four-stage design: document–rule retrieval, layout-preserving information extraction, auxiliary metadata enrichment, and auditable symbolic/arithmetic verification. LAVA supports robust rule grounding, fine-grained error attribution, and consistent, traceable end-to-end execution—capabilities essential for high-stakes deployment. Evaluated on a large real-world benchmark with diverse financial documents and dozens of expert-curated validation rules, LAVA outperforms baselines in hallucination control and edge-case handling while maintaining efficient token usage, demonstrating practicality for high-volume, time-critical validation.

1 Introduction

Regulatory penalties, financial losses, and reputational damage can all result from a single error in financial document validation, making accuracy, consistency, and auditability non-negotiable. Financial institutions process millions of documents daily across workflows such as loan underwriting, payroll auditing, tax compliance, and fraud detection. The challenge is acute for semi-structured documents like statements, invoices, and tax slips, which span multiple pages, exhibit irregular layouts, encode domain-specific business logic, and often arrive as noisy scans or non-standard PDFs (Bhattacharyya et al., 2025; Ding et al., 2024a; Xu et al., 2020; Chen et al., 2024).

Recent years have witnessed rapid progress in visually rich document understanding through layout- and structure-aware pretraining of multimodal large language models (MLLMs). LayoutLM (Xu et al., 2020) pioneered spatial–textual joint encoding, followed by DocFormer (Appalaraju et al., 2021), DocLLM (Wang et al., 2024), mPLUG-DocOwl2 (Hu et al., 2025), and ROP (Zhang et al., 2024), advancing multimodal architectures for structural representation. This shift moves beyond pure text modelling to multimodal document intelligence. However, evaluations remain dominated by perceptual and question answering (QA) tasks, probing reasoning in a narrow, task-specific manner while rarely accessing validation—reasoning that tests both interpretation and reliability under structured and cross-field constraints.

Enterprise-grade document validation presents requirements beyond those of perception or reasoning alone. It calls for symbolic rule enforcement, cross-field consistency, and multi-step logical coherence—capabilities only partially reflected in existing benchmarks (Wang et al., 2023b; Li et al., 2025; Borchmann et al., 2021). Recent datasets have expanded evaluation to layout-aware perception and understanding (Zhu et al., 2024; Wu et al., 2023; Mathew et al., 2021; Stanisławek et al., 2021; Šimsa et al., 2023), but compliance-critical validation logic remains underexplored. In practice, enterprises often patch the gap by pairing general-purpose models with rigid rule-based modules (Shende et al., 2024). Yet even state-of-the-art models that excel at extraction or understanding exhibit ongoing limitations when directly applied to validation: hallucinations remain common, and reasoning traceability is limited. Adaptation across regulatory schemas is fragile, and computational costs escalate with token usage at enterprise scale.

[*]Equal contribution.
[†]Corresponding author.
[‡]Work done during an internship at BMO Financial Group.

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

These shortcomings make validation not merely an extension but a distinct and emerging frontier of document intelligence—one that demands frameworks where accuracy, efficiency, audit readiness, and robustness are central requirements.

Our Work. To address these challenges, we present **LAVA** (**L**ogic-**A**ware **V**alidation and **A**ugmentation), a modular and efficient framework for verifiable reasoning over semi-structured, layout-complex financial documents, designed for real-world applicability, fine-grained error attribution, and rapid adaptation and reproducibility across structurally similar collections. Agnostic to backbone models, LAVA extends beyond static benchmarks by integrating (i) layout-informed knowledge extraction preserving structural cues, (ii) domain-aware augmentation with contextual metadata, and (iii) arithmetic and symbolic verification ensuring factual alignment with business rules, orchestrated in a four-stage system of retrieval, extraction, augmentation, and hybrid reasoning.

We evaluate LAVA on a real-world large-scale industrial benchmark with validation rules curated by senior industry experts reflecting real regulatory conditions. Results show competitive gains in factual accuracy and symbolic correctness, together with lower computational overhead than baseline MLLM pipelines, demonstrating robustness and cost-effectiveness in realistic financial validation scenarios.

Our main contributions are:

- **Task and System.** We formalize *financial document validation* as a multi-document reasoning task—largely absent in existing benchmarks—and instantiate it in **LAVA**, a novel modular framework designed for accurate and auditable validation of semi-structured financial documents.

- **Reasoning Strategy & Auditability.** We design a controllable hybrid reasoning framework unifying factual/contextual templates with symbolic and arithmetic tasks via explicit formula generation, with an external checker fallback ensuring correctness, thereby enhancing accuracy, interpretability, and operational robustness.

- **Evaluation.** We propose a comprehensive evaluation framework covering symbolic correctness, factual alignment, and hallucination control for fine-grained reasoning assessment in realistic validation workflows.

2 Related Work

Visually Rich Document Understanding. Recent work has shifted from extraction pipelines toward LLM-centric modeling that integrates layout and visual cues. DocLayLLM (Liao et al., 2025) adds visual patches and 2D positional tokens, VisDoM (Suri et al., 2025) combines multimodal retrieval with consistency constraints, 3MVRD (Ding et al., 2024b) aligns fine- and coarse-grained signals via multi-task distillation, and LayoutLLM (Fujitake, 2024) applies instruction tuning for unified document tasks. These advances improve representation and generalization; our work is complementary, focusing on *auditable validation*—explicit symbolic checks, cross-field consistency, and reproducible reasoning traces required in compliance-critical workflows. Our framework is model-agnostic, plugging into any MLLM backbone to introduce validation as a controllable layer.

Layout-Guided Document Encoding. Backbones such as LayoutLMv3 (Huang et al., 2022), FormNet (Lee et al., 2022, 2023), and DocFormer (Appalaraju et al., 2021) combine textual, spatial, and visual cues through large-scale pretraining. These excel at form-style entity extraction but remain embedding-level and not optimized for symbolic validation. In contrast, our framework leverages layout-derived structures to enable modular business rule checks and reproducible logic tracing, addressing auditability without massive training effort.

LLM Verification and Rule-based Validation. Progress in LLM factuality—via self-checking (Dhuliawala et al., 2024), retrieval-augmented prompting (Qin et al., 2025), symbolic grounding (Hennigen et al., 2024), and rectification (Kang et al., 2024)—has improved reliability in clean text and formal reasoning (Liu et al., 2025). Yet these methods falter on noisy, heterogeneous documents with long-range dependencies and embedded business rules. Traditional rule engines (e.g., Drools) provide transparency but fail under layout variation, while hybrid LLM–rule or knowledge graph systems (Vertsel and Rumiantsau, 2024; Sadowski and Chudziak, 2025) trade flexibility for interpretability. Our framework instead couples symbolic verification with layout-

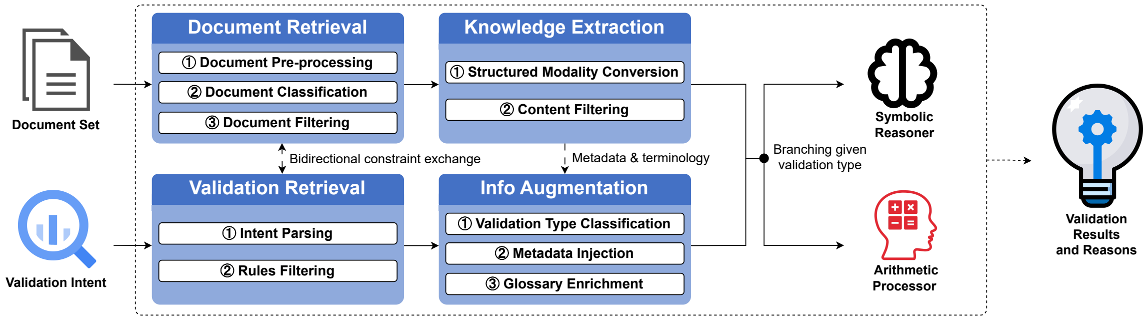


Figure 1: Overview of LAVA. The architecture comprises two parallel pipelines—document processing and rule grounding—interacting via bidirectional constraints (solid arrows: main data flow; dashed arrows: auxiliary exchange). The document-processing track performs retrieval, extraction, and augmentation to produce layout-preserving structured content with enriched metadata. The rule-grounding track retrieves, classifies, and dispatches applicable validation rules to either the Symbolic Reasoner or Arithmetic Processor, depending on reasoning type.

and metadata-informed prompting, merging rule transparency with neural adaptability, making it fit for compliance-critical validation.

3 Method

Problem Formalization. We define **financial document validation** as a human-in-the-loop copilot task, where the system assists users (e.g., underwriters, compliance officers, fraud analysts) in verifying whether a set of financial documents (e.g., from a mortgage application) satisfies certain business rules under real-world conditions of layout complexity, domain-specific logic, and noisy input.

Inputs. The system takes as input: (1) a document set $\mathcal{D} = \{d_1, \dots, d_N\}$ in scanned PDF or image format, where each d_i is a semi-structured financial document (e.g., bank statement, tax form), and (2) a validation intent q, a user-specified verification goal in natural language (e.g., Does gross income exceed loan threshold?).

Objective. The system maps $(q, \mathcal{D})$ to a set of validation outputs $\mathcal{V} = \{v_k\}$, where each v_k includes: a subset of supporting documents $\mathcal{D}_k \subseteq \mathcal{D}$, a set of retrieved business rules $\mathcal{R}_k \subseteq \mathcal{R}$ from a **predefined** rule library, a binary verification label $y_k \in \{\texttt{Pass}, \texttt{Fail}\}$, and an explanation trace e_k for auditability. This formulation supports multi-document, logic-grounded reasoning while ensuring modular, interpretable validation aligned with enterprise workflows.

Architecture. Figure 1 illustrate LAVA architecture, which employs a modular design paradigm to address the complexity of enterprise-grade production systems. Although this design involves the integration of multiple components, it yields critical advantages for deployment. Decoupling the workflow isolates potential points of failure within individual modules, enabling targeted and independent validation and thereby systematically reducing the long-term verification cost and overall operational overhead. Furthermore, the ability to debug, update, or replace modules without systemic disruption enhances maintainability—a stark contrast to the challenges of managing opaque, end-to-end models. This design philosophy is therefore foundational to building a robust, auditable, and scalable system fit for the rigors of real-world financial validation.

3.1 Document and Validation Retrieval

We jointly describe the first two modules, as they operate in a tightly coupled fashion to determine relevant document-rule pairs for downstream extraction. Given a user-specified *validation intent* q and a document set $\mathcal{D} = \{d_1, \dots, d_N\}$, the modules select a subset $\mathcal{D}_v \subseteq \mathcal{D}$ that is temporally valid and relevant to the task, along with a set of executable business rules $\mathcal{R}_v = \{r_1, \dots, r_M\}$. Both subsets are tailored to the verification goal through a bidirectional constraint mechanism, ensuring only applicable document-rule pairs are forwarded for knowledge extraction and augmentation.

Document Retrieval. Documents are first preprocessed to normalize layout and correct visual artifacts (e.g., OCR errors, rotation, skew) (Boudraa et al., 2020), preserving alignment and structural fidelity for downstream modules. Each document d_i is then classified into a predefined document type using a lightweight image-based classifier such as TinyViT (Wu et al., 2022), as financial documents of the same type generally share consistent page-level features. Recognized types are forwarded to Validation Retrieval to constrain rule applicability.

Document-level metadata (e.g., date, coverage period) is extracted using a template-guided NER pipeline with regex patterns and rule-based heuristics. Temporal constraints parsed from the validation intent in the next module (e.g., "past 3 months") are applied to filter out documents outside the relevant time window. In addition, applicable document types extracted from retrieved rules in Validation Retrieval are passed back to further prune documents irrelevant to all candidate rules.

Validation Retrieval. Given q, this module retrieves a subset of rules $\mathcal{R}_v$ from the predefined library $\mathcal{R}$, based on semantic relevance and document compatibility. The rule library is enriched with metadata specifying applicable document types. Lightweight LLMs can parse q to extract temporal constraints, while a sentence encoder (e.g., Sentence-BERT (Reimers and Gurevych, 2019)) encodes q to retrieve top-K semantically relevant rules. Retrieved rules are then filtered using document-type constraints from Document Retrieval, and their own document-type metadata is fed back to further refine $\mathcal{D}_v$.

In conclusion, temporal and semantic cues from q filter the document set, while recognized document types and rule metadata eliminate inapplicable rules. This closed-loop filtering minimizes irrelevant candidates on both sides, reduces reasoning load, and improves accuracy without sacrificing interpretability, ensuring downstream processing operates on the most relevant and valid document-rule pairs.

3.2 Knowledge Extraction

This module transforms each filtered document $d_i \in \mathcal{D}_v$ into a compact, layout-aware hybrid representation for downstream reasoning. Instead of flat key–value pairs, we produce a structured markup that encodes page layout, visual grouping, and field dependencies, serving as a bridge between scanned formats and language-model-friendly input.

Structured Modality Conversion. To capture the rich visual and semi-structural semantics of financial documents, we adopt an HTML-like markup constructed from parsed layout and OCR signals. Prior work shows that retaining tabular alignments, hierarchical sections, and field groupings improves reasoning fidelity (Sui et al., 2024), but raw markup is insufficient for noisy scans. We therefore augment it with: (1) structural parsing via document analysis tools (e.g., LayoutParser (Shen et al., 2021), LayoutLMv3 (Huang et al., 2022)), AWS Textract; (2) OCR-based recovery (e.g., Tesseract OCR (Smith, 2007)) for free-form or scattered content, including reconstruction of long paragraphs into coherent spans; (3) proximity-based grouping to merge fragmented tokens into coherent semantic units; and (4) visual region preservation for inherently non-textual content (e.g., charts, stamps, signatures), where candidate regions are identified from layout cues (e.g., low text coverage or OCR confidence) and retained as image patches for the MLLM input.

Content Filtering. The extracted representation often contains much noise from headers, footers, boilerplate blocks, or placeholders. We prune such elements using DOM structure, field labels (e.g., `Name`, `Address`), and positional cues, reducing token usage while improving attention focus for model prompts.

By combining structure-preserving markup, semantic recovery, selective visual preservation, and noise reduction, this module delivers a high-fidelity hybrid form that maintains interpretability while enabling reliable downstream reasoning.

3.3 Information Augmentation

This module enriches downstream reasoning by injecting auxiliary signals from both documents and retrieved rules. It serves two purposes: routing rules to the appropriate verification pathway and augmenting prompts with rich metadata to improve model understanding.

Each rule $r_k \in \mathcal{R}_v$ is classified as symbolic (context-dependent logic) or arithmetic (numeric computation) via a lightweight LLM query, avoiding brittle heuristics. In parallel, metadata is additionally extracted from layout-preserving outputs: language (via token-based detection), document types (from retrieval module), and domain-specific terms (e.g., withholding or CPP [1]) detected through lexical and structural heuristics targeting low-frequency tokens, abbreviations, and left-hand-side predicates in field labels or rule expressions. This metadata is distilled into concise semantic clarifications for interpretability and disambiguation. The resulting signals are incorporated into prompt headers or reference blocks, sharpening alignment

[1] CPP refers to the Canada Pension Plan, a mandatory pension contribution in Canadian payroll systems.

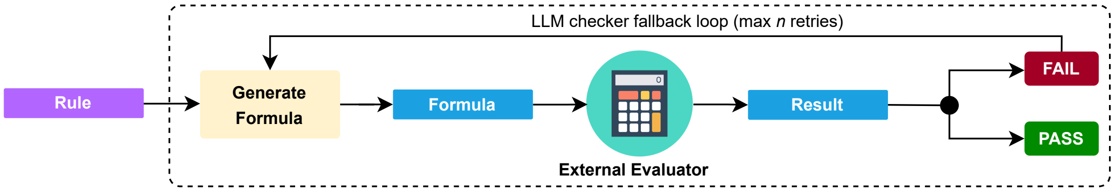


Figure 2: Arithmetic Processor. Given a validation rule, the system generates a formula and delegates evaluation to an external calculator. A checker LLM validates alignment between the rule and result. Failures trigger a fallback loop with negative feedback.

with rule semantics and improving precision in complex validation scenarios, while keeping the verification loop efficient.

3.4 Validation

To support diverse verification needs and enhance the reliability, we adopt a bifurcated framework with two sub-modules: **Arithmetic Processor** for numerical tasks (e.g., verifying tax deductions, calculating gross revenue), and **Symbolic Reasoner** for all other general rules requiring semantic and contextual reasoning.

Arithmetic Processor. As illustrated in Figure 2, when rules involving arithmetic or numerical computation are routed to this sub-module, instead of relying on LLMs for direct computation—prone to hallucinations and numeric instability, we adopt a tool-use paradigm where the model is used solely for generating a task-specific formula, which is then executed by a deterministic external engine such as a Python interpreter (Gao et al., 2023; Chen et al., 2022). To ensure alignment between the formula and the validation semantics, we introduce a fallback auditing loop: the rule and generated formula are reviewed by a secondary LLM "checker". If a mismatch is detected, the formula is regenerated, conditioned on the previous (incorrect) version as a negative example. This loop improves robustness by systematically detecting and correcting errors, mitigating hallucinated computations and flawed reasoning pathways.

Symbolic Reasoner. This sub-module handles semantic or structural reasoning. In contrast to Arithmetic Processor, it directly delegates rules to a general-purpose model, as such reasoning falls within the model's inherent strengths. This design choice reduces integration overhead and inference latency, while remaining sufficient for a wide range of non-arithmetic verification scenarios.

Prompt Excerpt for Symbolic Reasoner

Excerpt: only key parts are shown below, simplified for illustration.

You are an assistant verifying `{document_types}` documents (in `{languages}`), with extracted `{knowledge_names}` ...

Instructions:
1. Use `{structured_representation}` as references for identifying the layout...
2. Use `{semantic_fields}` as references for explicit context...

   ...

n. Output as follows:
   - Only output the fields requested in the validation rule...
   - Format: empty = ""; date = YYYYMMDD...

   Example: `{example_output}`

Task Input:
`{validation_rule}`
- `{term_1}` : `{definition_1}`, ..., `{term_n}` : `{definition_n}`

Output: Return JSON only. No explanation or preamble...

Figure 3: Prompt excerpt for Symbolic Reasoner, dynamically built from the retrieved validation rule, extracted knowledge, and augmentation components. Full versions for both Arithmetic Processor and Symbolic Reasoner are provided in Appendix A.1.1 and A.1.2.

Prompts. We adopt our prompt construction strategy in meta-prompting fashion (Zhang et al., 2023) that presents task-relevant information in a step-wise and zero-shot format rather than relying on illustrative examples. Unlike chain-of-thought (Wei et al., 2022), few-shot prompting (Brown et al., 2020), or self-consistency (Wang et al., 2023a), our approach avoids brittle curated examples, which struggle to generalize across heterogeneous financial documents (Zhou et al., 2023), and removes high inference cost of example-heavy prompting.

To ensure verification prompts are precisely tailored to both the rules and associated documents, we implement a dynamic, template-based prompt construction system. It integrates supplemental information from upstream modules with the rule to populate flexible prompt templates. By leveraging templating libraries like Jinja, we employ programmatic, code-like logic to govern the inclusion and granularity of metadata based on both document layout and task semantics. As illustrated in Figure 3, this mechanism produces prompts that are maximally informative while remaining token-efficient and robust to both arithmetic and symbolic reasoning tasks. By structuring all inputs according to rule logic, metadata, and knowledge, our approach generalizes well across heterogeneous financial documents. The resulting system remains accurate, cost-efficient and context-aware, supporting scalable and reliable verification tasks.

4 Experiments

We evaluate LAVA on real-world Canadian mortgage application documents sampled from a proprietary database to assess the system within a consistent validation scenario. The set includes multiple document types and around 1,000 scanned PDFs or images—such as tax forms, bank/investment statements, and legal agreements—selected for their semi-structured formats, diverse layouts, and rich logical dependencies (see A.4). This focused yet heterogeneous domain enables rigorous testing of LAVA's ability to produce accurate outputs and avoid false positives, particularly where heuristic, zero-shot, and few-shot LLM-based methods struggle with layout variability or reasoning complexity.

To ensure rigorous and interpretable evaluation, we adopt rule-level testing rather than intent-level, with validation rules drawn from a curated library provided by business stakeholders. Each rule corresponds to an atomic validation unit and is applied only to the document types for which it is defined. For fairness and comparability, document type metadata is included with the dataset, since accurate validation cannot be assessed without first matching each document to its correct rule set; this ensures that all pipelines are evaluated under the same conditions, without document and validation retrieval. Ground-truth outcomes are manually annotated by domain experts, and automatic evaluation is complemented by manual audits from business collaborators, ensuring both accuracy and institutional credibility in line with production-grade expectations for document validation.

4.1 Implementations

To ensure fairness, all baselines, LAVA, and LAVA's ablation variants use Claude 3.7 Sonnet (Anthropic, 2024) as the validation component. We configure the model with a maximum response length of 5120 tokens and enable reasoning (thinking) with a budget of 1024 tokens, allowing the model to better understand complex validation rules and generalize across documents with diverse layouts and edge scenarios. For LAVA's Arithmetic Processor, the maximum retry number n in the fallback loop is set to 2, to ensure a practical trade-off between error correction and computational efficiency. And all other LLM parameters are kept identical across evaluations.

For document analysis in experiments, we prioritized a methodology that ensures our findings are portable, reproducible, and immediately accessible. To this end, our experiment exclusively utilizes off-the-shelf tools that do not require custom training or dataset-specific fine-tuning. A foundational layer of raw text and spatial information was established for the experiment using the open-source Tesseract OCR (Smith, 2007). For the processing of structural semantics, such as tables and forms, the publicly available AWS Textract service was employed. This consistent and transparent tooling strategy not only ensures that the comparative evaluation in Sections 4.4 and 4.5 fairly assesses the core architectural and reasoning capabilities of the different pipelines, but also directly supports other researchers to easily replicate and build upon our results.

4.2 Validation Rules

We group several dozen validation rules into five categories, reflecting distinct reasoning and computation demands. These categories assess the pipeline's ability to extract information, integrate rules, and perform contextual reasoning. See A.2 for complete versions of some representative rules.

4.3 Evaluation Metrics

Our evaluation metrics comprehensively assess reasoning quality across all pipeline stages. Given the distinct characteristics of our compliance-sensitive validation task compared to common document understanding settings, we design task-specific metrics with particular emphasis on suppressing false positives—a critical industrial concern causing costly investigations, delays, and loss of trust. We evaluate baselines from three perspectives to cover these aspects. Their formal definitions are given in Appendix A.3, along with visual examples showing how each metric manifests in an illustrative sample document.

Hallucination Control. We report the percentage of responses with two failure types:

Metric	Rule Group	VLM + Field-Level OCR	LLM + Field-Level OCR	LLM + Enhanced OCR	LAVA
Factual Hallucination Rate	C_1	0.03	0.03	0.01	**0.01**
	C_2	0.08	0.10	0.04	**0.03**
	C_3	0.31	0.33	0.30	**0.18**
	C_4	0.05	0.08	0.05	**0.03**
	C_5	0.28	0.30	0.15	**0.05**
Numerical Infidelity Rate	C_1	N/A	N/A	N/A	**N/A**
	C_2	0.08	0.04	0.03	**0.01**
	C_3	0.33	0.31	0.20	**0.10**
	C_4	0.11	0.08	0.05	**0.00**
	C_5	0.25	0.10	0.10	**0.00**
Edge Case Error Rate	C_1	N/A	N/A	N/A	**N/A**
	C_2	0.27	0.25	0.23	**0.02**
	C_3	0.86	0.90	0.82	**0.17**
	C_4	0.46	0.34	0.43	**0.11**
	C_5	0.89	0.92	0.86	**0.08**

Table 1: Performance of baselines across three metrics and five validation rule categories (C_1–C_5). C_1: content extraction; C_2: conditional logic reasoning; C_3: multi-step logic reasoning; C_4: unconstrained arithmetic consistency checking; C_5: constrained arithmetic consistency checking.

- **Factual Hallucination Rate:** Content not grounded in the source document, such as fabricated values or formulas, and unsupported claims.

- **Numerical Infidelity Rate:** Incorrect quantitative reasoning or numerical derivations, such as incorrect formula execution results and conclusions inconsistent with the given numerical evidence.

Edge Case Handling. This metric measures failure rate on complex or exception-driven scenarios, about 10%–25% of document–rule pairs. Such cases require *adaptive logic reasoning* beyond fixed rules and broader *logic coverage* for diverse edge conditions. As these challenges often overlap, we report a single rate to capture both, where higher values indicate weaker generalization and reasoning flexibility.

Token Cost. We measure input and output tokens per rule check to assess computational efficiency and deployment cost, highlighting trade-offs between reasoning quality and efficiency across pipelines. For each document-rule pair, token counts are recorded and aggregated over the full evaluation set.

4.4 Comparative Baselines

Here, *Field-Level OCR* denotes OCR output containing only individual field texts and their spatial information (e.g., from bounding boxes), without higher-level structure such as tables, sections, or relational links between fields. This representation preserves raw layout signals but omits the structured markup used in LAVA's pipeline.

We consider three representative settings:

- **VLM + Field-Level OCR:** Document images with Field-Level OCR in a VQA setup, where OCR text aids visual grounding.

- **LLM + Field-Level OCR:** Same OCR content without images, measuring performance from textual–spatial data alone in a QA setup.

- **LLM + Enhanced OCR (no structural markup):** OCR refined with domain-specific cleanup and recovery steps identical to LAVA's extraction, but without structured markup, isolating LAVA's impact of structured representation and modular reasoning.

Since OCR and markup information might introduce noise (e.g., when text is misrecognized), we also examine an OCR-free configuration to assess the impact of textual knowledge. This setting is evaluated in ablation study (Section 4.5), where all other modules are preserved but the model operates only on document images. It is excluded from the baseline set because the baselines are intended to preserve textual content in some form for fair comparison, whereas this variant probes the limits of visual-only reasoning and represents a more extreme ablation of the pipeline.

As shown in Table 1, LAVA achieves the lowest failure rates across all metrics and categories, with especially large gains in multi-step logic reasoning (C_3) and constrained arithmetic consistency checking (C_5), reducing hallucination and numerical errors by over 10% compared to the best baseline. And it records approximately 0% Numerical Infidelity Rate in three of four categories. LAVA also substantially lowers edge case errors, indicating strong generalization under layout ambiguity and rare conditions. Common VQA and QA settings represented by the baselines underperform in this compliance-oriented validation task due to their limited handling of layout variability and domain-specific rules. These results highlight the benefit of structured prompts and modular reasoning in suppressing unsupported content generation and enabling reliable multi-step, context-aware reasoning over structural data.

We also sampled a subset of DocVQA (Mathew et al., 2021) documents with five manually defined rules each, where LAVA also achieved strong performance, confirming that our framework readily transfers to public data. More importantly, real-world validation tasks—where both documents and rules are richer in semantics, context, and reasoning complexity—pose substantially greater challenges, where our approach proves particularly effective under industry-level conditions.

4.5 Ablation Study

To assess the contribution of each core module in LAVA, we performed an ablation study using a unified metric: the percentage of responses that fail to exactly match the ground truth. This stricter measure was chosen because LAVA's earlier evaluations achieved low error rates in category-specific metrics, so a single comprehensive indicator better reveals performance drops when modules are altered or removed.

We evaluated six configurations: the full LAVA system; three **Knowledge Extraction (KE)** variants; and versions without **Info Augmentation (IA)** or **Arithmetic Processor (AP)**. The KE variants progressively reduce structural detail and change input format:

1. **Markdown KE** replaces LAVA's structured markup with Markdown-like fashion.
2. **Plain-text KE** flattens layout into text aligned to mimic visual spacing but without structural tags.
3. **No KE** omits textual knowledge entirely, prompting the model with original document images. This configuration serves as the OCR-free, vision-only benchmark we referred to in the baseline comparison (Section 4.4).

Pipeline	C_1	C_2	C_3	C_4	C_5
LAVA w/o KE	0.03	0.65	0.67	0.09	0.66
LAVA w/ Plain-text KE	0.02	0.63	0.58	0.08	0.48
LAVA w/ Markdown KE	0.01	0.25	0.54	0.05	0.55
LAVA w/o IA	0.01	0.15	0.45	0.05	0.12
LAVA w/o AP	0.01	0.03	0.29	0.10	0.56
LAVA	**0.01**	**0.03**	**0.28**	**0.05**	**0.07**

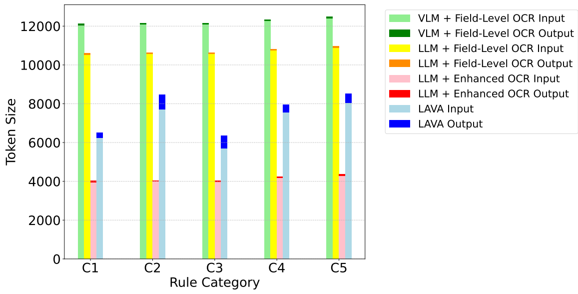
Table 2: Percentage of responses failing to exactly match ground-truth answers across five rule categories.

This stepwise removal strips explicit layout and relational cues, forcing reliance on surface text or raw images. For AP ablation, AP is replaced by Symbolic Reasoner, removing explicit calculation/result validation. The two Retrieval modules are not ablated since all experiments use a fixed rule set, making them independent of validation accuracy.

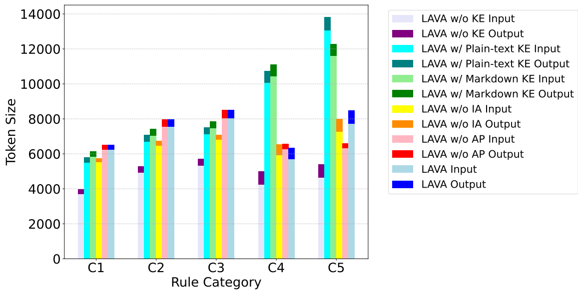
The error patterns in table 2 shows that KE, IA, and AP play complementary roles, with KE being the most critical. Removing KE forces the model to rely almost entirely on visual reasoning, raising C_2, C_3, and C_5 failure rate to 0.65–0.67—over twice LAVA's error in multi-step logic (C_3) and nearly tenfold in constrained arithmetic consistency checking (C_5) even with the validation module's assistance. Two factors explain this: (1) Structural conversion: without KE's structural format that preserves complex and nested layout, grouping, and field dependencies, multi-step logic and table reasoning lose explicit relational cues. Downgrading to Markdown or plain text progressively strips away these signals, with Markdown underperforming plain text in C_5 due to weaker preservation of columnar alignment in tables; (2) Content filtering: without KE's filtering stage, noisy headers, footers, and irrelevant fields dilute attention and amplify hallucination risk. This gradient from fully structured to none shows that noise suppression, explicit spatial and relational cues are essential for stable reasoning, especially in multi-field and arithmetic-heavy rules for financial document validation tasks.

IA mainly impacts logical reasoning: removing it raises C_3 error from 0.28 to 0.45, suggesting that metadata and domain cues help models resolve field semantics and linking conditions across steps.

AP's effect is computation-focused: removing it leaves C_1–C_3 unchanged but drives C_4/C_5 error



(a) LAVA and baselines.



(b) LAVA and ablated variants.

Figure 4: Average input and output token counts across five rule categories.

to 0.10/0.56, reflecting the limits of unconstrained LLM reasoning in arithmetic tasks and the benefit of explicit calculation with fallback mechanism.

Overall, the degradations are consistent with the design intent: KE supplies the structured substrate, IA enriches it with semantic context, and AP enforces numerical fidelity, justifying LAVA's functionally complementary and modular composition.

4.6 Business Impact

While Section 4.4 and 4.5 highlight LAVA's technical advantages in accuracy and robustness, real-world adoption also depends on its operational efficiency and cost-effectiveness. By consolidating token usage, we can further assess how LAVA scales under realistic operational constraints, using token consumption as a proxy for latency and API expenditure in high-volume production workflows.

LAVA amortizes multi-step costs by extracting structured knowledge once and reusing it across rules. As shown in Figure 4a, this design cuts input tokens by 25%–45% versus VLM and LLM baselines with Field-Level OCR, lowering inference costs and enabling faster turnaround for time-critical financial validation processes, while also achieving fewer errors (Table 1).

Ablation results (Figure 4b) show that in non-computation tasks (C_1–C_3), excluding variant w/o KE, LAVA adds minimal tokens with stable output length due to the predefined schema-based response format. For computation tasks (C_4–C_5), the less faithful structural representation from plain-text and Markdown KE inflate inputs by causing more model output errors and retries in the validation loop. Across all ablations, full LAVA adds under 3k tokens per rule ($\sim$ \$0.009 with Claude 3.7 or $\sim$ \$0.006 with GPT-4.1), an overhead outweighed by substantial gains in accuracy and reasoning stability. This demonstrates that LAVA delivers a dual advantage: (1) immediate computational savings from reduced token usage, and (2) long-term operational efficiency from its modular, auditable design. This combination suggests that LAVA's architectural benefits outweigh the modular integration complexity, making it a cost-effective and scalable choice for high-volume, time-critical real-world financial workflows.

5 Conclusion

We presented LAVA, a modular, interpretable and backbone- and domain-agnostic system for enterprise-grade financial document validation. Combining layout-preserving extraction, domain-aware augmentation, and symbolic and arithmetic verification, LAVA achieves high factual alignment and numerical reliability on a large real-world benchmark, with lower computational overhead than monolithic prompting. Evaluation with factual hallucination rate, numerical infidelity rate, and edge case handling shows that structured, multi-stage reasoning enables fine-grained error attribution, stronger robustness in compliance-sensitive workflows, and transfer to structurally similar data. Looking ahead, we aim to handle noisier, less-structured documents and to learn rule generalization across formats and domains.

6 Ethical Considerations

This research was conducted using a proprietary dataset of financial documents, with all data fully anonymized and handled under strict institutional privacy protocols. We acknowledge the potential risk of algorithmic bias in financial decision-making. LAVA's design as an accurate, scalable and auditable system is a deliberate step to mitigate this risk by enhancing transparency and ensuring human accountability. Nevertheless, any production deployment would necessitate rigorous, ongoing audits for demographic bias to ensure fair and equitable outcomes.

References

Anthropic. 2024. Claude 3 model family. `https://www.anthropic.com/news/claude-3-family`. Accessed: 2025-08-07.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 973–983. IEEE.

Aniket Bhattacharyya, Anurag Tripathi, Ujjal Das, Archan Karmakar, Amit Pathak, and Maneesh Gupta. 2025. Information extraction from visually rich documents using LLM-based organization of documents into independent textual segments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17241–17256, Vienna, Austria. Association for Computational Linguistics.

Łukasz Borchmann, Michał Pietruszka, Tomasz Stanislawek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Graliński. 2021. Due: End-to-end document understanding benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Omar Boudraa, Walid Khaled Hidouci, and Dominique Michelucci. 2020. Using skeleton and hough transform variant to correct skew in historical documents. *Mathematics and computers in simulation*, 167:389–403.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks.

Yufan Chen, Jiaming Zhang, Kunyu Peng, Junwei Zheng, Ruiping Liu, Philip Torr, and Rainer Stiefelhagen. 2024. Rodla: Benchmarking the robustness of document layout analysis models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 15556–15566. IEEE.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.

Yihao Ding, Jean Lee, and Soyeon Caren Han. 2024a. Deep learning based visually rich document content understanding: A survey. *ArXiv preprint*, abs/2408.01287.

Yihao Ding, Lorenzo Vaiani, Caren Han, Jean Lee, Paolo Garza, Josiah Poon, and Luca Cagliero. 2024b. 3MVRD: Multimodal multi-task multi-teacher visually-rich form document understanding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15233–15244, Bangkok, Thailand. Association for Computational Linguistics.

Masato Fujitake. 2024. LayoutLLM: Large language model instruction tuning for visually rich document understanding. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10219–10224, Torino, Italia. ELRA and ICCL.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: program-aided language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR.

Lucas Torroba Hennigen, Shannon Shen, Aniruddha Nrusimha, Bernhard Gapp, David Sontag, and Yoon Kim. 2024. Towards verifiable text generation with symbolic references. *Preprint*, arXiv:2311.09188.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2025. mPLUG-DocOwl2: High-resolution compressing for OCR-free multi-page document understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5817–5834, Vienna, Austria. Association for Computational Linguistics.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Haoqiang Kang, Juntong Ni, and Huaxiu Yao. 2024. Ever: Mitigating hallucination in large language models through real-time verification and rectification. *Preprint*, arXiv:2311.09114.

Chen-Yu Lee, Chun-Liang Li, Timothy Dozat, Vincent Perot, Guolong Su, Nan Hua, Joshua Ainslie, Renshen Wang, Yasuhisa Fujii, and Tomas Pfister. 2022. FormNet: Structural encoding beyond sequential modeling in form document information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume*

1: Long Papers), pages 3735–3754, Dublin, Ireland. Association for Computational Linguistics.

Chen-Yu Lee, Chun-Liang Li, Hao Zhang, Timothy Dozat, Vincent Perot, Guolong Su, Xiang Zhang, Kihyuk Sohn, Nikolay Glushnev, Renshen Wang, Joshua Ainslie, Shangbang Long, Siyang Qin, Yasuhisa Fujii, Nan Hua, and Tomas Pfister. 2023. FormNetV2: Multimodal graph contrastive learning for form document information extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9011–9026, Toronto, Canada. Association for Computational Linguistics.

Siqi Li, Yufan Shen, Xiangnan Chen, Jiayi Chen, Hengwei Ju, Haodong Duan, Song Mao, Hongbin Zhou, Bo Zhang, Bin Fu, Pinlong Cai, Licheng Wen, Botian Shi, Yong Liu, Xinyu Cai, and Yu Qiao. 2025. Gdi-bench: A benchmark for general document intelligence with vision and reasoning decoupling.

Wenhui Liao, Jiapeng Wang, Hongliang Li, Chengyu Wang, Jun Huang, and Lianwen Jin. 2025. Doclayllm: An efficient multi-modal extension of large language models for text-rich document understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4038–4049.

Chengwu Liu, Ye Yuan, Yichun Yin, Yan Xu, Xin Xu, Zaoyu Chen, Yasheng Wang, Lifeng Shang, Qun Liu, and Ming Zhang. 2025. Safe: Enhancing mathematical reasoning in large language models via retrospective step-aware formal verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12171–12186, Vienna, Austria. Association for Computational Linguistics.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Yuehan Qin, Shawn Li, Yi Nian, Xinyan Velocity Yu, Yue Zhao, and Xuezhe Ma. 2025. Don't let it hallucinate: Premise verification via retrieval-augmented logical reasoning.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Albert Sadowski and Jarosław A. Chudziak. 2025. Explainable rule application via structured prompting: A neural-symbolic approach. *Preprint*, arXiv:2506.16335.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. Layoutparser: A unified toolkit for deep learning based document image analysis. In *ICDAR*.

Abhishek Shende, Mahidhar Mullapudi, and Narayana Challa. 2024. Enhancing document verification systems: A review of techniques, challenges, and practical implementations. *International Journal of Computer Engineering & Technology*, 15:16–25.

Štěpán Šimsa, Milan Šulc, Michal Uřičář, Yash Patel, Ahmed Hamdi, Matěj Kocián, Matyáš Skalický, Jiří Matas, Antoine Doucet, Mickaël Coustaty, and 1 others. 2023. Docile benchmark for document information localization and extraction. In *International Conference on Document Analysis and Recognition*, pages 147–166. Springer.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 629–633, Washington, DC, USA. IEEE Computer Society.

Tomasz Stanisławek, Filip Graliński, Anna Wróblewska, Dawid Lipiński, Agnieszka Kaliska, Paulina Rosalska, Bartosz Topolski, and Przemysław Biecek. 2021. *Kleister: Key Information Extraction Datasets Involving Long Documents with Complex Layouts*, page 564–579. Springer International Publishing.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A. Rossi, and Dinesh Manocha. 2025. VisDoM: Multi-document QA with visually rich elements using multimodal retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6088–6109, Albuquerque, New Mexico. Association for Computational Linguistics.

Aliaksei Vertsel and Mikhail Rumiantsau. 2024. Hybrid llm/rule-based approaches to business insights generation from structured data. *Preprint*, arXiv:2404.15604.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2024. DocLLM: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8529–8548, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023b. VRDU: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, pages 5184–5193. ACM.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Anran Wu, Luwei Xiao, Xingjiao Wu, Shuwen Yang, Junjie Xu, Zisong Zhuang, Nian Xie, Cheng Jin, and Liang He. 2023. Dcqa: Document-level chart question answering towards complex reasoning and common-sense understanding.

Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. 2022. Tinyvit: Fast pretraining distillation for small vision transformers. In *European conference on computer vision*, pages 68–85. Springer.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Chong Zhang, Yi Tu, Yixi Zhao, Chenshu Yuan, Huan Chen, Yue Zhang, Mingxu Chai, Ya Guo, Huijia Zhu, Qi Zhang, and Tao Gui. 2024. Modeling layout reading order as ordering relations for visually-rich document understanding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9658–9678, Miami, Florida, USA. Association for Computational Linguistics.

Yifan Zhang, Yang Yuan, and Andrew Chi-Chih Yao. 2023. Meta prompting for ai systems.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Fengbin Zhu, Ziyang Liu, Xiang Yao Ng, Haohui Wu, Wenjie Wang, Fuli Feng, Chao Wang, Huanbo Luan, and Tat Seng Chua. 2024. Mmdocbench: Benchmarking large vision-language models for fine-grained visual document understanding.

A Appendix

A.1 Prompt used in LAVA

A.1.1 Prompt for Symbolic Reasoner

Dynamically Generated Prompt

You are an assistant who verifies financial documents. You are given {document_types} (in {languages}) and extracted {knowledge_names}. Your job is to verify the documents based on the validation rule. Please do not make up any values.

<instructions>
1. Use structured representation as references for identifying the layout and structure of the documents.
   <html_tables>{knowledge[tables]}</html_tables>
   <forms>{knowledge[forms]}</forms>
   ...
2. Use the semantic fields as references for providing explicit contextual information.
   <semantic_fields>{knowledge[fields]}</semantic_fields>

...

n. Output as follows:
   - Only output the fields requested in the validation rule, do not include any other fields stated in example_output.
   - Replace all spaces in keys with underscores "_".
   - If any field is not present, set value to "".
   - Output date in YYYYMMDD format.

   <example_output>{example_output}</example_output>
</instructions>

<validation_rule>
{validation_rule}
- {domain_specific_term_1} : {explanation_1}

...
- {domain_specific_term_n} : {explanation_n}

</validation_rule>

```
<output_format>
Return solely the JSON object without any additional explanations, comments, preamble, or formatting like backticks.
</output_format>
```

A.1.2 Prompt for Arithmetic Processor

Prompt for Formula Generation

System Prompt:
You are an assistant who generates a Python-style calculation formula to explain how a request field should be computed. You are given {document_types} (in {languages}), as well as knowledge extracted from the documents and a validation rule.

<instructions>
1. Identify the requested field from the rule. Store its name in the JSON object with the key "field_name", delete any special symbols and replace all spaces in keys with underscores "_".
2. Extract the stated value for the requested field.
   - Store it in the JSON object with the key "stated" and make sure it is a valid float or integer.
   - If the requested field is not present or empty, set "stated" to "NaN".
   - Do not make up any values.
3. Identify the relevant numerical fields in the knowledge that are needed to compute the requested field. Consider how those fields logically combine to produce the value of the requested field.
4. Once you find an appropriate calculation expression, store it in Python-executable format in the JSON object with the key "formula":
   - Do not compute the result.
   - Do not include any functions like "round(...)" or similar.
   - Only use raw numerical operations.
5. Return solely the JSON object without any additional explanations, comments, preamble, or formatting like backticks.

</instructions>

<example_output>
```
{
    "field_name": "Current_Total_Income",
    "stated": 600,
    "formula": "100 + 200 + 300"
}
```
</example_output>

User Prompt:
<knowledge>{knowledge}</knowledge>

<validation_rule>
{validation_rule}
- {domain_specific_term_1} : {explanation_1}

...
- {domain_specific_term_n} : {explanation_n}

</validation_rule>

Prompt for Formula Correction

System Prompt:
You are the greatest financial auditor, logician and deducer. You are given {document_types} (in {languages}), as well as knowledge extracted from the documents, a validation rule, and a calculation expression.

<instructions>
1. The calculation expression in response to the validation rule is wrong based on the knowledge extracted.
2. Check if there is any other correct calculation expression that can be derived from the data given.
3. Once you find an appropriate calculation expression, store it in Python-executable format:
   - Do not compute the result using equal sign.
   - Do not include any functions like "round(...)" or similar.
   - Only use raw numerical operations.
4. Return solely the calculation expression without any additional explanations, comments, preamble, or formatting like backticks.

</instructions>

<example_output_1>100 + 200 + 300</example_output_1>
<example_output_2>80 * 100.5</example_output_2>

User Prompt:
<knowledge>{knowledge}</knowledge>

<wrong_calculation>{wrong_calculation}<wrong_calculation>

<validation_rule>
{validation_rule}
- {domain_specific_term_1} : {explanation_1}

...
- {domain_specific_term_n} : {explanation_n}

</validation_rule>

A.2 Representative Validation Rules and Output Examples Used in Experiment

1. Content Extraction
   Rule:
   (a) Are Current and YTD regular pay/salary amounts present in documents?
   (b) Is Current and YTD CPP/QPP (may appear as Government Pension) present in the paystub?
   (c) Is Social Insurance Number (SIN) present in the T4 document?

   Output Example

   ```
   {
       "Employer_Name": {"present": true, "value": "organization"},
       "Current_Regular_Pay": {"present": false, "value": ""},
   }
   ```

2. Conditional Logic Reasoning
   (a) **Rule:**

<CRA_EI_data>`{EI_data}`</CRA_EI_data>
<EI_verification_rules>

- When Current EI is non-zero, skip the check and set "valid" to true.
- When Current and YTD EI are both 0 or blank, skip the check and set "valid" to false.
- When Current EI is 0 or blank, YTD EI is not 0 or blank. If YTD EI $\leq$ EI cap, set "valid" to true. Otherwise, set "valid" to false.

</EI_verification_rules>
Do the EI values comply with rules based on the Canada Revenue Agency (CRA) guidelines?

Output Example

```
{
    "Pay_Date": "20241128",
    "EI": {"Current": "100", "YTD": "200", "Cap":
"1049.12", "valid": true}
}
```

(b) **Rule:**
<numerical_verification_rules>

- Cells with empty value, or only numbers, symbols, punctuation are valid.
- Cells containing words (alphabetic characters) are invalid.
- If all fields are valid, return an empty object { }.

<numerical_verification_rules>
Based on the rules provided, does every non-header cell in tables contain a valid numerical value?

Output Example

```
{
    "This_Period_Regular": "amount",
    "YTD_CPP": "value"
}
```

3. Multi-step Logic Reasoning
Rule:
<pay_freq_cap>
[
    {"freq": "monthly", "cap": 12},
    {"freq": "semimonthly", "cap": 24},
    {"freq": "biweekly", "cap": 26},
    {"freq": "weekly", "cap": 52}
]
</pay_freq_cap>
<freq_verification_rules>

- If start or end date is empty, set "stated" to "" and set "equal" to None.
- If No. of pay period is empty, set "stated" to "" and set "equal" to None.
- If No. of pay period is not empty, find its corresponding "frequency cap" from pay frequency rules:
  - Compute the expected "current period number": Use "Pay Frequency", "Start Date" and "frequency cap" to calculate how many periods have elapsed since the start of the year.
  - Extract current period number from No. of pay period stated (e.g., 22 on "22 of 26").
  - Check if the "stated" current period number" equals the "expected current period number". If both equal, set "equal" to true. Otherwise, set "equal" to false.

</freq_verification_rules>
Based on the rules provided, does pay frequency and No. of pay period in the extracted date comply with the pay frequency rules?

Output Example

```
{
    "Start_Date": "20240101",
    "End_Date": "20240131",
    "Pay_Frequency": {"stated": "semimonthly", "expected":
"monthly", "equal": false},
    "NO_Pay_Period": {"stated": "11", "expected": "1",
"equal": false}
}
```

4. Unconstrained Arithmetic Consistency Checking
Rule:
   (a) Is the YTD gross pay calculated correctly by summing up all earning items?
   (b) Is Line 23600 (Net Income) correctly calculated as Line 15000 (Total Income) minus Deductions from Total Income?
   (c) Is the ending balance correctly calculated as the starting balance plus total deposits minus total withdrawals for the period?

Output Example

```
{
    "Current_Gross_Pay": {"calculation": "100 + 200 + 300 =
600", "stated": "300", "equal": true},
    "YTD_Gross_Pay": {"calculation": "1500 + 200 + 300 =
2000", "stated": "", "equal": null}
}
```

5. Constrained Arithmetic Consistency Checking
Rule:
   (a) Is the Current regular pay calculated correctly by rate * hours when rounding the result to 2 decimal places? Set formula to

an empty string"" when rate or hours is not present or 0.

(b) Is the current net pay correctly computed as the current gross pay minus all current deductions, with the result rounded to 2 decimal places? If the table includes a "Total Deduction" field, use it directly instead of summing individual deduction items.

Output Example

{
"Current_Gross_Pay": {"calculation": "100 + 200 + 300 = 600", "stated": "300", "equal": true},
"YTD_Gross_Pay": {"calculation": "1500 + 200 + 300 = 2000", "stated": "", "equal": null}
}

A.3 Evaluation Metrics

A.3.1 Factual Hallucination Rate (FHR).

Given a fixed rule r with evaluation document set $\mathcal{D}_r = \{d_1, \ldots, d_{n_r}\}$, let $\hat{\mathcal{E}}_{i,r}$ denote the predicted evidence subset in the model's output explanation trace for (d_i, r), and $\mathcal{E}(d_i, r)$ the gold evidence set from d_i. Let $\hat{\varphi}_{i,r}$ denote the predicted formula for (d_i, r), and g_r the corresponding gold formula.

We mark a sample (d_i, r) as factually hallucinated if either hold:

(a) **Evidence Hallucination:** $\hat{\mathcal{E}}_{i,r} \not\subseteq \mathcal{E}(d_i, r)$.

(b) **Formula Hallucination:** $\hat{\varphi}_{i,r} \not\equiv g_r$ (tested via polynomial identity testing with Schwartz–Zippel).

Formally, the sample-level indicator is

$$\mathrm{FH}_{i,r} = \mathbf{1}\big[\,(\hat{\mathcal{E}}_{i,r} \not\subseteq \mathcal{E}(d_i, r)) \;\vee\; (\hat{\varphi}_{i,r} \not\equiv g_r)\big]. \tag{1}$$

The rule-level rate is

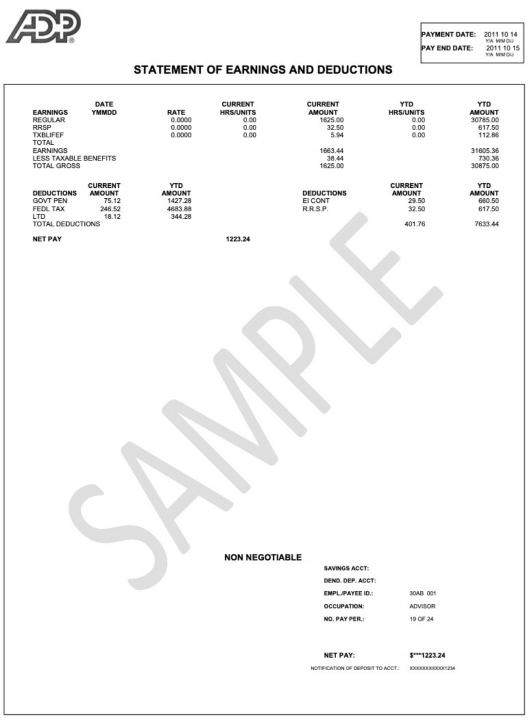
$$\mathrm{FHR}_r = \frac{1}{|\mathcal{D}_r|} \sum_{d_i \in \mathcal{D}_r} \mathrm{FH}_{i,r}, \tag{2}$$

and the group-level average for $C \subseteq \mathcal{R}$ is

$$\mathrm{FHR}^{\mathrm{w}}_C = \frac{\sum_{r \in C} |\mathcal{D}_r| \cdot \mathrm{FHR}_r}{\sum_{r \in C} |\mathcal{D}_r|}. \tag{3}$$

Containment checks normalize case, punctuation, numeric formatting, and unit conventions. Formula equivalence ($\equiv$) is evaluated by polynomial identity testing under randomized substitution.

Example. Consider the rule "Is the YTD total deduction computed as the sum of all YTD items in the deduction table?" The pipeline is asked to extract relevant evidence, provide a calculation, and conclude the validation result for this task. Based on Figure 5, the ground truth is:



PAYMENT DATE: 2011 10 14 (Y/A M/M D/J)
PAY END DATE: 2011 10 15 (Y/A M/M D/J)

STATEMENT OF EARNINGS AND DEDUCTIONS

EARNINGS	DATE YMMDD	RATE	CURRENT HRS/UNITS	CURRENT AMOUNT	YTD HRS/UNITS	YTD AMOUNT
REGULAR		0.0000	0.00	1625.00	0.00	30785.00
RRSP		0.0000	0.00	32.50	0.00	617.50
TXBLIFEF		0.0000	0.00	5.94	0.00	112.86
TOTAL EARNINGS				1663.44		31605.36
LESS TAXABLE BENEFITS				38.44		730.36
TOTAL GROSS				1625.00		30875.00

DEDUCTIONS	CURRENT AMOUNT	YTD AMOUNT	DEDUCTIONS	CURRENT AMOUNT	YTD AMOUNT
GOVT PEN	75.12	1427.28	EI CONT	29.50	660.50
FEDL TAX	246.52	4683.88	R.R.S.P.	32.50	617.50
LTD	18.12	344.28			
TOTAL DEDUCTIONS				401.76	7633.44

NET PAY 1223.24

SAMPLE

NON NEGOTIABLE

SAVINGS ACCT:
DEND. DEP. ACCT:
EMPL./PAYEE ID.: 30AB 001
OCCUPATION: ADVISOR
NO. PAY PER.: 19 OF 24

NET PAY: $***1223.24
NOTIFICATION OF DEPOSIT TO ACCT.: XXXXXXXXXXX1234

Figure 5: Sample document shown for clarity and error illustrations; metrics are document-agnostic and applied unchanged across all document types (sensitive content redacted).

Ground Truth Output

{
"YTD_Total_Deduction": {
"calculation": "1427.28 + 4683.88 + 344.28 + **660.50** + 617.50 = 7733.44",
"stated": "7633.44",
"equal": false
}
}

When *factual hallucination* occurs, the pipeline fabricates one or some of the variables in the calculation formula (e.g., changing 660.50 to 560.50) so that the sum exactly matches the stated total:

Output with Factual Hallucination

{
"YTD_Total_Deduction": {
"calculation": "1427.28 + 4683.88 + 344.28 + **560.50** + 617.50 = 7633.44",
"stated": "7633.44",
"equal": true
},
}

Consider the rule: "Are pay period start and end dates present in the document?" The pipeline is asked to check the presence of specific fields and extract the corresponding value. The ground truth for Figure 5 is:

Ground Truth Output


```
{
    "Start_Date": {"present": false, "value": ""}
    "End_Date": {"present": true, "value": "20111015"}
}
```


When *factual hallucination* occurs, the pipeline fabricates a start date by using the value of another field (in this case, the payment date) and incorrectly marking it as present as a consequence:

Output with Factual Hallucination


```
{
    "Start_Date": {"present": true, "value": "20111014"}
    "End_Date": {"present": true, "value": "20111015"}
}
```


These two examples illustrate distinct modes of factual hallucination. In the *deduction case*, the pipeline invents a new numerical value so that the arithmetic matches the stated total. In the *date case*, the pipeline fabricates a missing field by repurposing another field. In both scenarios, the error arises not from incorrect arithmetic or logical reasoning but from introducing evidence that is absent from the source document.

A.3.2 Numerical Infidelity Rate (NIR).

Given a fixed rule r with evaluation set $\mathcal{D}_r = \{d_1, \dots, d_{n_r}\}$, let $\hat{\varphi}_{i,r}$ denote the predicted formula or reasoning process appearing in the model's output explanation trace for (d_i, r), and $v^\star_{i,r}$ the gold numerical result.

We mark a sample (d_i, r) as numerically infidel if the evaluated result of the predicted formula or reasoning process deviates from the gold value:

$$\mathrm{NI}_{i,r} = \mathbf{1}\left[\mathrm{eval}(\hat{\varphi}_{i,r}) \neq v^\star_{i,r}\right], \tag{4}$$

with deviations judged within absolute/relative tolerances.

The rule-level rate is

$$\mathrm{NIR}_r = \frac{1}{|\mathcal{D}_r|} \sum_{d_i \in \mathcal{D}_r} \mathrm{NI}_{i,r}, \tag{5}$$

and the group-level average for $C \subseteq \mathcal{R}$ is

$$\mathrm{NIR}^{\mathrm{w}}_C = \frac{\sum_{r \in C} |\mathcal{D}_r| \cdot \mathrm{NIR}_r}{\sum_{r \in C} |\mathcal{D}_r|}. \tag{6}$$

Note that hallucinations in formula generation are already captured under FHR; NIR isolates purely numerical inconsistencies after formula or reasoning process generation.

Example. Using the same rule for the YTD deduction value as in A.3.1:

Output with Numerical Infidelity


```
{
    "YTD_Total_Deduction": {
        "calculation": "1427.28 + 4683.88 + 344.28 + 660.50 + 617.50 = 7633.44",
        "stated": "7633.44",
        "equal": true
    },
}
```


Here, all evidence terms are faithfully extracted, and the pipeline correctly generates the formula. But the result is miscomputed as 7633.44 instead of the correct 7733.44, possibly influenced by the stated value in the document. Unlike factual hallucination, this error does not arise from fabricating or misattributing values, but from applying the correct evidence while failing to maintain consistency in quantitative reasoning and arithmetic derivations.

A.3.3 Edge Case Handling (ECH).

For each rule r with evaluation set $\mathcal{D}_r$, we predefine a subset $\mathcal{D}^{\mathrm{edge}}_r \subseteq \mathcal{D}_r$ containing complex or exception-driven cases (typically 10%–25% of document–rule pairs). Such cases include missing or abnormal values, atypical field combinations, and boundary conditions requiring adaptive reasoning beyond commom patterns.

For a document $d_i \in \mathcal{D}^{\mathrm{edge}}_r$, let $\hat{y}_{i,r} \in \{\texttt{Pass}, \texttt{Fail}\}$ be the model's predicted label and $y^\star_{i,r}$ the gold label. We define the indicator

$$\mathrm{EC}_{i,r} = \begin{cases} 1, & \hat{y}_{i,r} \neq y^\star_{i,r}, \\ 0, & \hat{y}_{i,r} = y^\star_{i,r}. \end{cases} \tag{7}$$

The rule-level error rate is

$$\mathrm{ECH}_r = \frac{1}{|\mathcal{D}^{\mathrm{edge}}_r|} \sum_{d_i \in \mathcal{D}^{\mathrm{edge}}_r} \mathrm{EC}_{i,r}, \tag{8}$$

and the group-level average is

$$\mathrm{ECH}^{\mathrm{w}}_C = \frac{\sum_{r \in C} |\mathcal{D}^{\mathrm{edge}}_r| \cdot \mathrm{ECH}_r}{\sum_{r \in C} |\mathcal{D}^{\mathrm{edge}}_r|}. \tag{9}$$

Higher values of ECH indicate weaker generalization and lower robustness on edge conditions.

Example. Consider the rule: "Is the current gross pay calculated correctly by summing up the current amount of all earning items?" In the sample (Figure 5), the earnings table contains a subtotal of 1663.44 followed by a negative adjustment of 38.44. Correct handling requires the pipeline to exclude the subtotal from the formula and include the adjustment as a subtraction. The ground truth is:

Ground Truth Output

```
{
    "Current_Gross_Pay": {
        "calculation": "1625.00 + 32.50 + 5.94 - 38.44 = 1625.00",
        "stated": "1625.00",
        "equal": true
    }
}
```

However, the pipeline misinterprets the table:

Output with Edge Case Error

```
{
    "Current_Gross_Pay": {
        "calculation": "1625.00 + 32.50 + 5.94 + 1663.44 + 38.44 =
3365.32",
        "stated": "1625.00",
        "equal": false
    },
}
```

In this output, the pipeline incorrectly reuses the subtotal 1663.44 as if it were another earning item, and also flips the negative adjustment 38.44 into a positive contribution. The error is not due to arithmetic mistakes but to misinterpreting atypical structures in the table. Unlike factual hallucination (fabricating values) or numerical infidelity (mis-computing a correct formula), this case reflects weaker robustness to edge conditions. Instead of applying rules mechanically, a reliable industrial pipeline should capture the underlying business logic and correctly adapt it to the diverse formats and exception cases present in financial documents.

A.4 Evaluation Dataset

Our evaluation uses a proprietary collection of production-level mortgage application documents from an active industrial workflow. The corpus spans a broad range of core and supporting document categories—covering proof of income, property appraisal, account statements, tax forms, and legal agreements—with file lengths varying from single-page forms to multi-dozen-page reports. Unlike public datasets that are often synthetic or visually uniform, our corpus retains the full heterogeneity, layout irregularities, OCR noise, and compliance constraints encountered in real underwriting. While individual files and categories vary across evaluations, the overall data composition and associated challenges are fully described, enabling reproducibility on comparable financial document collections.

Table 3: Representative mortgage document types and key characteristics.

Document Type	**Key Structural Features**	**Key Semantic Features**	**Valid Rule Categories**
Appraisal	Multi-page PDF; photos + valuation tables; firm templates; checkboxes	Market value; comparable properties; adjustments; effective date	C_1, C_2
Bank Account Statement	Multi-column ledgers; footnotes inside tables; embedded logos and stamps	Balance progression; transaction category; cross-month reconciliation	$C_1 – C_5$
Employment Letter	Letterhead; signatures; uneven paragraphs; low-contrast scans	Title; start date; compensation; pay frequency vs. paystubs	C_1, C_2, C_3
Investment Account Statement	Dense holdings tables; various formats; small-font disclosures and disclaimers	Issue/maturity date; interest rate; holding list; balance	C_1, C_3, C_4, C_5
Mortgage Statement	Multi-section layout; various formats; tables with merged cells	Outstanding balance; interest rate and type; payment due date; amount due	C_1, C_3, C_4, C_5
Notice of Assessment	CRA form; dense numeric blocks; tiny labels; presence of official headers	Total/taxable income; refund amount; social insurance number; match T4	C_1, C_2, C_3, C_4
Paystub	Multiple tables and forms; various layouts; faded scans	Gross; net; deductions; start/end/pay date	C_1 - C_5
Personal Income Tax Return	Multipage; mixed sections; checkboxes; pre-filled and handwritten; cross-page linkage	Address; citizenship; total/employment/net income	C_1, C_2, C_4
Property Tax Bill	Multiple tables; various layouts; inconsistent labelling for tax components; watermarks and logos	Annual tax; installments/penalties; roll/assessment IDs; schedule math	C_1 - C_5
Purchase Sales Agreement	Long contract; initials/signatures; clause order varies; appendices with separate numbering	Property address; price; closing date; buyer/seller name	C_1, C_2, C_3
Realtor Listing	Embedded images; dense tables and forms; inconsistent field ordering	List price; property address; listing date and status; square footage	C_1, C_2
Realtor Listing UW	Listing + UW notes; handwritten overlays	Adjusted price; remarks; reconcile with appraisal/APS	C_1, C_2, C_3
Rental Lease Agreement	Multi-page legal contract; various templates; small-font clauses; handwritten	Monthly rent; term dates; payment plan; residence overlap	C_1, C_2, C_3
Separation Agreement	Multi-page legal document; dense narrative clauses; annex schedules	Support obligations; asset division; enforceable clauses; child custody	C_1, C_2, C_3
T4: Statement of Remuneration Paid	Fixed CRA box layout; small-font numeric fields; bilingual field labels; scan noise	Employment income; Province; EI/CPP exemptions; align paystub/NOA	C_1, C_2, C_4
T4A: Statement of Pension, Retirement, Annuity, and Other Income	Fixed CRA box layout; small-font numeric fields; bilingual field labels; scan noise	Pension/annuity/self-employment income; income tax deducted; social insurance number	C_1, C_2, C_4
T4A(P): Statement of Canada Pension Plan Benefits	Fixed CRA box layout; small-font numeric fields; bilingual field labels; scan noise	CPP benefits amount; Income tax deducted;	C_1, C_2, C_4
T4RIF: Statement of Income From a Registered Retirement Income Fund	Fixed CRA box layout; small-font numeric fields; bilingual field labels; scan noise	Taxable amounts; Payer/issuer name; taxable amount; social insurance number	C_1, C_2, C_4
T5: Statement of Investment Income	Fixed CRA box layout; small-font numeric fields; bilingual field labels; scan noise	Actual amount of dividends; Interest from Canadian sources; Reported income vs. bank statements	C_1, C_2, C_5
T776: Statement of Real Estate Rentals	Fixed CRA box layout; small-font numeric fields; bilingual field labels; scan noise	Gross rents; Total expenses; Net income (loss) before adjustments; Capital cost allowance (CCA) claim	C_1 - C_5

FinCoT: Grounding Chain-of-Thought in Expert Financial Reasoning

Natapong Nitarach, Warit Sirichotedumrong, Panop Pitchayarthorn,
Pittawat Taveekitworachai, Potsawee Manakul, Kunat Pipatanakul

SCB 10X, SCBX Group

Abstract

This paper presents FinCoT, a structured chain-of-thought (CoT) prompting framework that embeds domain-specific expert financial reasoning blueprints to guide large language models' behaviors. We identify three main prompting styles in financial NLP (FinNLP): (1) standard prompting (zero-shot), (2) unstructured CoT (free-form reasoning), and (3) structured CoT (with explicitly structured reasoning steps). Prior work has mainly focused on the first two, while structured CoT remains underexplored and lacks domain expertise incorporation. Therefore, we evaluate all three prompting approaches across ten CFA-style financial domains and introduce FinCoT as the first structured finance-specific prompting approach incorporating blueprints from domain experts. FinCoT improves the accuracy of a general-purpose model, Qwen3-8B-Base, from 63.2% to 80.5%, and boosts Fin-R1 (7B), a finance-specific model, from 65.7% to 75.7%, while reducing output length by up to 8.9× and 1.16× compared to structured CoT methods, respectively. We find that FinCoT proves most effective for models lacking financial post-training. Our findings show that FinCoT does not only improve performance and reduce inference costs but also yields more interpretable and expert-aligned reasoning traces.

1 Introduction

Financial decision–making, such as stochastic modeling, risk assessment, portfolio optimization, and algorithmic trading (Markowitz, 1952; Black and Scholes, 1973a; Rockafellar and Uryasev, 2000; Avellaneda and Stoikov, 2008), demands precise mathematics and domain-specific reasoning (Lewkowycz et al., 2022; Wen and Zhang, 2025). Recent advances in large foundation models for finance, such as the multimodal FinTral (Bhatia et al., 2024) and language-centric Fin-R1 (Liu et al., 2025), demonstrate progress but still face challenges in interpretability and domain alignment (Nie et al., 2024; Arya.ai, 2024; Lee et al., 2025). Accordingly, these shortcomings motivate stricter control over a model's intermediate reasoning path, which we explore via prompt design.

Prompting guides LLM reasoning without extra training. Methods such as Chain-of-Thought (Wei et al., 2023), Code Prompting (Hu et al., 2023), Plan-and-Solve (Wang et al., 2023), and Self-Reflection (Renze and Guven, 2024) encourage stepwise thinking but remain domain-agnostic. In finance, this can lead to omissions in valuation checks or confusion between basis points and percentages. Yet real-world analysis follows well-defined workflows—valuation, discounting, portfolio attribution—that depend on explicit intermediate structure. Embedding such structure in the prompt helps the model verify units, formulas, and boundary conditions, improving interpretability and alignment with expert practice.

We design **FinCoT**, a zero-shot prompt that injects expert financial workflows-encoded as Mermaid blueprints-into a structured CoT template, yielding auditable reasoning without fine-tuning. Across ten CFA domains, FinCoT significantly boosts accuracy (most in quantitative areas) and produces shorter, clearer outputs than standard or structured CoT prompts. This paper offers three main contributions:

- We provide a comprehensive investigation and the first taxonomy of financial prompting–covering standard prompting, unstructured CoT, and structured CoT/FinCoT–clarifying how each paradigm addresses domain-specific reasoning requirements.
- We release nine blueprint templates—conceptual diagrams modeled after the Unified Modeling Language (UML) (Engels et al., 2000) and rendered in Mermaid syntax–that LLMs can parse as plain–text hints to drive structured reasoning both

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

within FinCoT and in broader domain-specific prompting scenarios.

- On 1.032k CFA–style questions across ten financial domains and four open-source LLMs, FinCoT shows notable gains–up to +17.3 pp in accuracy ($p < 0.001$)–particularly on pretrained models and quantitatively structured tasks. While less effective on instruction-tuned or niche domains, FinCoT consistently reduces verbosity ($\sim 8\times$ fewer tokens) and improves reasoning trace clarity under a three–point interpretability rubric.

2 Background and Related Work

2.1 Prompt Engineering

Standard Prompting (SP): Refers to the baseline technique of simply providing a natural language instruction to an LLM, without providing any intermediate `'thinking'` steps, demonstrations, or explicit reasoning cues-i.e., a zero-shot setup. While the GPT-3 paper (Brown et al., 2020) popularized few-shot prompting via exemplars, more recent work formalizes and benchmarks zero-shot prompting as a distinct paradigm (Wei et al., 2022). Our implementation follows the formulation shown in Appendix Listing 1 (ZS) in (Callanan et al., 2024), and is used to represent the standard prompting baseline.

Unstructured Chain-of-Thought (UST-CoT): A general-purpose reasoning strategy using free-form CoT to establish a baseline for unconstrained prompting. These include:
- **Chain-of-Thought (CoT)** (Wei et al., 2023): Decompose reasoning into intermediate steps, encouraging the model to deliberately and systematically `'think'` before finalizing an answer.
- **Code Prompting** (Hu et al., 2023): Translates problems into executable code, allowing the model to simulate logic or perform precise computations. In other words, LLMs are elicited to reason explicitly and transparently through code.
- **Plan-and-Solve** (Wang et al., 2023): Separates planning from solving, where the model first outlines a high-level plan, then executes the reasoning based on that plan.

In addition to these, other prompting techniques have emerged, such as Tree of Thoughts (ToT) (Yao et al., 2023), which explores multiple reasoning paths via tree-structured search; Graph of Thoughts (GoT) (Besta et al., 2024), which frames reasoning as a graph with LLM-generated nodes and edges for flexible information flow. These methods enhance expressiveness for general tasks; they are not tailored for finance, which requires mathematical precision and domain-specific constraints. We adopt the template from Appendix Listing 2 CoT (Callanan et al., 2024) as the default prompt for this baseline.

Structured Chain-of-Thought (ST-CoT): ST-CoT augments a prompt with tags such as <thinking> and <output> that break the model's response into explicit, modular stages, promoting incremental, easily replaceable reasoning blocks. This tag-driven format has already appeared in open-source trials.[1] Figure 1 visually contrasts ST-CoT with SP, UST-CoT, and FinCoT.

FinCoT (§3) inherits ST-CoT's structure but injects domain-specific Mermaid blueprints to ground each step in expert workflows. Unlike Universal Self-Adaptive Prompting (Wan et al., 2023), which derives few-shot exemplars from LLM memory, FinCoT encodes human-crafted financial reasoning, favoring interpretability and control. The three categories Standard Prompting (direct instruction), Unstructured CoT (free-form reasoning), and Structured CoT (tag-driven with optional expert hints) offer a unified lens for classifying financial prompting.

2.2 LLMs in Domain-Specific Financial Reasoning

Existing approaches to adapting large foundation models for financial reasoning currently fall into three broad paradigms:

Prompting-based: Methods use few-shot prompts with CFA-style queries, as in "Can GPT Models be Financial Analysts?" (Callanan et al., 2024), which evaluate ChatGPT and GPT-4 on mock CFA exams.

Fine-tuning-based: Methods adapt models via supervised fine-tuning with curated QA/classification datasets (Ma et al., 2023; Harsha et al., 2025) or continued pretraining on domain-specific corpora (Yang et al., 2020; Lee et al., 2024; Bhatia et al., 2024; Ke et al., 2025; Liu et al., 2025). While effective, these approaches rely on labeled data and general language modeling objectives, lacking structuring of financial reasoning, thus limiting interpretability and alignment with expert logic.

[1] `https://gist.github.com/davidmezzetti/1fac6bd406857431f1cdc74545bdfba9`

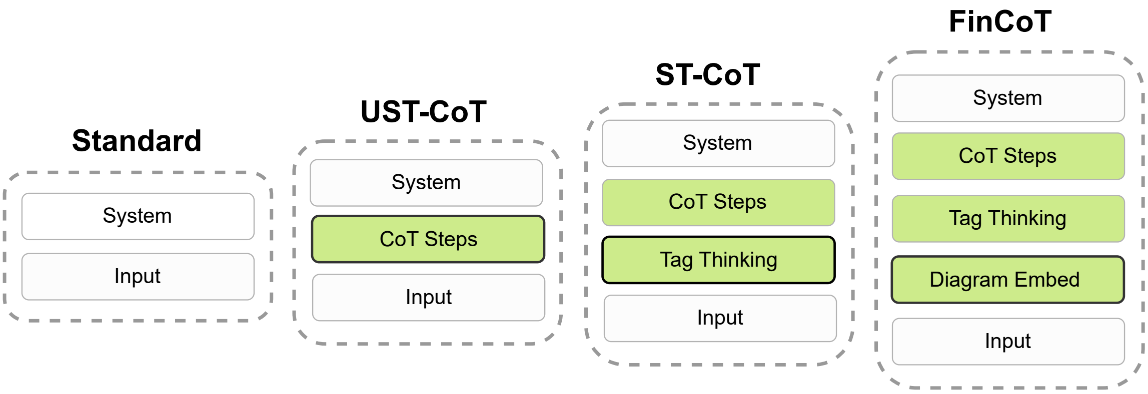


Figure 1: Taxonomy of prompting strategies by reasoning structure: SP, UST-CoT, ST-CoT, and FinCoT. Green blocks indicate added reasoning control–CoT steps, tagged thoughts, and expert diagrams (Diagram Embed)–showing the evolution toward more interpretable, domain-aligned prompts.

Despite advances, prior work has largely overlooked structured reasoning grounded in real-world workflows. We introduce a domain-based prompting framework designed to reflect step-by-step expert logic and evaluate it on CFA-style exam tasks.

3 FinCoT: Augmenting CoT with Structured Financial Expertise

We introduce FinCoT (Financial Chain-of-Thought), a structured prompting framework that enhances LLM reasoning in specialized financial domains. Building upon ST-CoT approaches, FinCoT explicitly embeds expert-derived problem-solving methodologies directly into prompts, guiding LLMs to follow domain-specific reasoning pathways without requiring model fine-tuning. Figure 2 illustrates the FinCoT architecture, which integrates expert-guided reasoning layers and reflective validation to improve performance in financial tasks.

3.1 The FinCoT Prompt Framework

The FinCoT prompting framework integrates the following key components and logical steps:

1. **System:** A single, top-level message that frames the task (e.g., “You are a CFA candidate; treat the following as a finance question”).
2. **Guided Step-by-Step Execution:** The prompt reserves two tag blocks <thinking> for intermediate reasoning and <output> for the final answer-thereby enforcing a structured chain-of-thought (ST-CoT) in one turn.
3. **Expert Reasoning Blueprint (via Mermaid Diagram (Sveidqvist and contributors, 2025):**

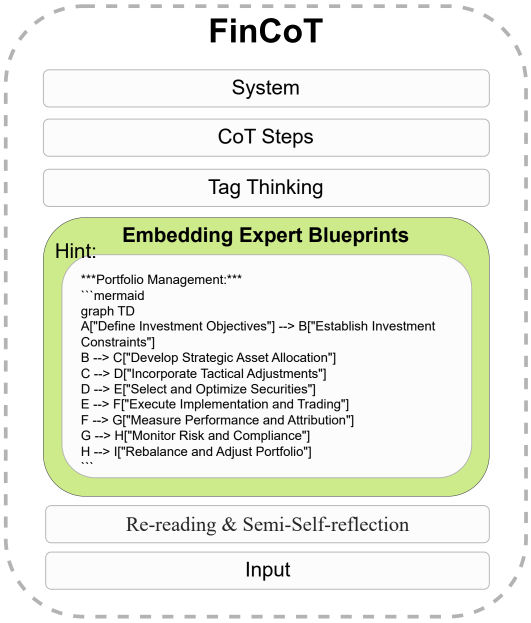


Figure 2: Overview of FinCoT, integrates structured, expert-guided reasoning layers Mermaid diagrams, plan-and-solve scaffolding, and reflective validation to improve performance in financial tasks.

A domain-specific, embedded expert blueprint with Mermaid diagram concerning the generation of diagrams (see §4), serve as a "Hint" within the context of the prompt. This blueprint explicitly outlines the recommended step-by-step problem-solving strategy for the given financial domain and is curated through a systematic process detailed in section 3.2 to ensure consistency and domain alignment.

4. **Re-Reading & Semi Self-Reflection:** Inside the <output> tags, the model briefly checks

its reasoning for consistency before committing the final answer. We call this "semi-reflection" because we drop the separate <reflection> block-avoiding per-step scoring and self-bias noted by Xu et al. (2024) yet still include a short self-check within <output>.

3.2 Embedding Expert Blueprints

The creation of effective expert reasoning blueprints involves a systematic multistage process designed to transform a wide range of financial knowledge into structured and actionable LLM diagrams.

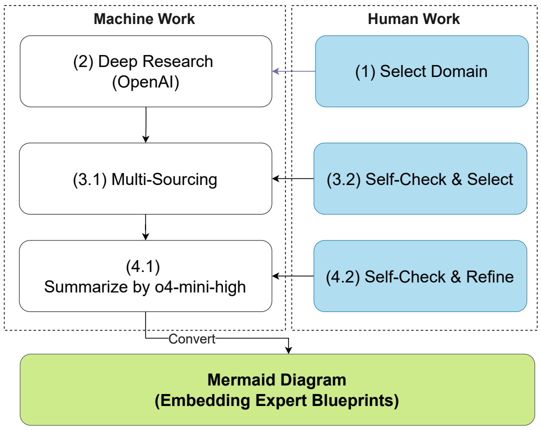


Figure 3: Pipeline for curating financial expert reasoning. Each stage systematically transforms raw financial knowledge into structured Mermaid diagrams for FinCoT prompting.

Curation Pipeline for Expert Reasoning: To construct expert blueprints for each financial domain, we implement a hybrid pipeline combining machine assistance and human judgment, as illustrated in Figure 3. The process includes the following stages:

1. **Scope Definition and Knowledge Aggregation:** The target CFA domain is scoped (e.g., Economics focusing on supply and demand), and relevant expert strategies are aggregated, using Deep Research[2], from diverse authoritative sources. Outputs are validated by human-in-the-loop reviewers with financial knowledge (e.g., finance graduates or CFA charterholders) to ensure conceptual accuracy and domain alignment.
2. **Validation and Synthesis:** We cross-reference and synthesize the aggregated knowledge to ensure accuracy, identify core principles and filter redundancies.
3. **Iterative Refinement into Structured Workflows:** The synthesized expert knowledge is iteratively transformed into logical step-by-step reasoning workflows. This refinement process focuses on ensuring the coherence, correctness, and clarity of the resulting problem-solving strategies for each financial domain.
4. **Mermaid Diagram Generation:** This refined workflow is translated into a Mermaid diagram (Bari et al., 2024) using its text-based syntax. We selected Mermaid due to its LLM prompt compatibility and clear visual guidance aligning with the FinCoT prompt. The diagrams are constructed based on the source content validated and synthesized first in 2, and then applied to each financial domain, with the entire collection organized and described in Appendix A as reference blueprints[3].
5. **Prompt Integration:** The text-based Mermaid blueprint is embedded as "Hint" within the FinCoT prompt template (Appendix B.2), directly guiding the LLM's reasoning.

4 Experimental Setup

Model Configurations and Inference Parameters: We selected the Qwen family of models due to their strong baseline performance in zero-shot financial reasoning. In preliminary evaluations, **Qwen2.5-7B-Instruct** achieved 69.7% accuracy on standard task prompts, substantially outperforming **Llama3.1-8B-Instruct** (46.3%), motivating its use as our primary model family. To evaluate the impact of both instruction tuning and domain-specific adaptation, we compare two model groups.
(A) General-purpose foundation models

- **Qwen2.5-7B (pretrained model)** vs. **Qwen2.5-7B-Instruct**: to assess the effect of instruction tuning on a strong general-purpose foundation model.
- **Qwen3-8B-Base (pretrained model)** vs. **Qwen3-8B**, **Qwen3-8B (Thinker)**: to isolate the impact of ST-CoT prompting within the same architecture.

[2]An AI agent for retrieving/synthesizing knowledge from public sources such as CFA notes, academic texts, and industry guides.

[3]The resulting expert blueprints are reviewed for conceptual consistency and practical correctness (but not guaranteed precision) by CFA Level III charterholders.

- **Gemma-3-12B-IT (Text-only)**: an instruction-tuned model recently released, comparable in size to Qwen3-8B. It achieved 52.81% accuracy on the Flare-CFA benchmark, outperforming Llama3.1-8B-Instruct, and serves as a competitive baseline.

(B) Financial-specific reasoning models

- **Fin-R1 (7B)**: adapted from Qwen2.5-7B-Instruct using supervised and reinforcement learning on a financial dataset distilled from DeepSeek-R1 (Liu et al., 2025).
- **DianJin-R1-7B**: fine-tuned from Qwen2.5-7B-Instruct using CFLUE, FinQA, and CCC, with GRPO to improve domain-specific reasoning (Zhu et al., 2025).
- **Fin-o1-8B**: built on Qwen3-8B and trained on the FinCoT[4] corpus using SFT and GRPO, setting a strong benchmark in financial reasoning (Qian et al., 2025).

All experiments used a maximum sequence length of 16.384k tokens. Following best practices for decoding stability (Du et al., 2025), we set the generation temperature to 0.2 to encourage focused and consistent outputs under evaluation conditions.

Prompting Strategies Compared: Our study evaluates the effectiveness of FinCoT against three baseline prompting strategies: SP, UST-CoT, and ST-CoT, which were detailed in Section 2. For clarity in this section:

- **SP:** The model receives only the target question.
- **UST-CoT:** In addition to the question, the model is given a generic cue to reason step-by-step.
- **ST-CoT:** This strategy employs structural tags (e.g., <thinking>, <output>) to guide the model in generating an organized step-by-step reasoning trace for the target question.
- **FinCoT:** A zero-shot prompting method that integrates expert domain templates (excluding the Ethics domain). Each prompt includes a Mermaid diagram as a "Hint" to guide structured financial reasoning via relevant domain insights.

[4]The FinCoT dataset is constructed by TheFinAI and publicly available at `https://huggingface.co/datasets/TheFinAI/FinCoT`. It combines financial QA datasets such as FinQA, ConvFinQA, TATQA, DocMath-Eval, Econ-Logic, BizBench-QA, and DocFinQA, with GPT-4o-generated reasoning traces to enhance structured financial question answering. Note that this dataset is not derived from our prompting approach.

While all methods operate in a zero-shot setting, FinCoT uniquely injects expert-guided structure through diagrams. Recent studies suggest that even large reasoning models may struggle with instruction-following when overloaded with reasoning cues (Li et al., 2025; Fu et al., 2025; Jang et al., 2025; Yao et al., 2025), though this remains underexplored in financial contexts. We thus evaluate whether CoT-style prompts (UST-CoT, ST-CoT, FinCoT) enhance instruction-following compared to SP.

Evaluation Benchmark: To assess financial reasoning, we use 1.032k multiple-choice questions from the CFA-Easy subset of FinEval (also referred to as Flare-CFA), originally introduced by Ke et al. (2025). This curated set reflects the rigor of CFA exams and enables evaluation across both theoretical and practical domains. Each question is categorized into one of ten CFA domains using GPT-4o with a dedicated classification prompt (see Appendix B.3), and Figure C shows the resulting domain distribution.

Evaluation Metrics: We report **accuracy** as the metric, defined as the percentage of questions where the model's prediction matches the ground truth. To assess response efficiency, we also report the **average output length** (in tokens) across questions. For statistical significance, we use a paired bootstrap test (Efron and Tibshirani, 1994) with $B{=}10k$ resamples over binary correctness scores, reporting the mean difference (Δ), 95% confidence interval, and p-value. Additionally, we measure the proportion of financial domains where a method improves accuracy by at least 1% over SP and compute the average domain-wise accuracy gain.

5 Results and Discussions

5.1 Financial Reasoning Performance

Baseline Performance: Table 1 reports zero-shot accuracies for four prompting strategies (SP, UST-CoT, ST-CoT, FinCoT) across our model suite. Under the basic SP prompt, the instruction-tuned **Qwen3-8B (Thinker)** attains the highest accuracy among general-purpose models (88.18%), while the financial model **Fin-o1-8B** leads its group at 79.65%. These strong baselines clearly highlight the effectiveness of instruction tuning and domain specificity.

Pretrained Models: On **Qwen2.5-7B**, FinCoT (All Blueprints) yields a +7.95 pp improvement

Prompt	Accuracy (%)								
	General models						Financial models		
	Qwen2.5-7B	Qwen2.5-7B Instruct	Qwen3-8B Base	Qwen3-8B	Qwen3-8B (Thinker)	Gemma-3-12B IT	Fin-R1 7B	DianJin-R1 7B	Fin-o1 8B
SP	54.07	69.67	63.18	74.42	88.18	52.81	65.70	78.39	**79.65**
UST-CoT	67.83 (↑13.76)	**75.68*** (↑6.01)	72.58 (↑9.40)	**82.36*** (↑7.94)	**89.05*** (↑0.87)	**77.81*** (↑25.00)	75.19 (↑9.49)	67.73 (↓10.66)	79.36 (↓0.29)
ST-CoT	**70.35*** (↑16.28)	74.52 (↑4.85)	78.49 (↑15.31)	81.01 (↑6.59)	88.18	76.74 (↑23.93)	74.32 (↑8.62)	68.80 (↓9.59)	78.39 (↓1.26)
FinCoT	62.02 (↑7.95)	74.22 (↑4.55)	**80.52*** (↑17.34)	81.10 (↑6.68)	87.21 (↓0.97)	75.58 (↑22.77)	**75.78*** (↑10.08)	**79.75*** (↑1.36)	77.23 (↓2.42)
Domain-wise performance of FinCoT									
Economics	69.09 (↑15.02)	73.26 (↑3.59)	79.26 (↑16.08)	79.55 (↑5.13)	86.92 (↓1.26)	74.61 (↑21.80)	73.45 (↑7.75)	55.52 (↓22.87)	78.00 (↓1.65)
FixedIncome	68.12 (↑14.05)	73.35 (↑3.68)	78.88 (↑15.70)	80.81 (↑6.39)	87.21 (↓0.97)	76.45 (↑23.64)	74.22 (↑8.52)	66.86 (↓11.53)	76.74 (↓2.91)
Quant.Meth.	68.02 (↑13.95)	**75.19** (↑5.52)	80.14 (↑16.96)	80.91 (↑6.49)	**87.79** (↓0.39)	75.68 (↑22.87)	74.90 (↑9.20)	65.79 (↓12.60)	77.42 (↓2.23)
EquityInvest.	69.09 (↑15.02)	74.22 (↑4.55)	79.26 (↑16.08)	80.52 (↑6.10)	86.72 (↓1.46)	76.45 (↑23.64)	74.42 (↑8.72)	62.31 (↓16.08)	78.68 (↓0.97)
Port.Manage.	67.54 (↑13.47)	74.13 (↑4.46)	**80.72** (↑17.54)	80.91 (↑6.49)	86.92 (↓1.26)	77.03 (↑24.22)	75.00 (↑9.30)	62.02 (↓16.37)	76.55 (↓3.10)
Derivatives	68.90 (↑14.83)	73.64 (↑3.97)	79.84 (↑16.66)	80.81 (↑6.39)	87.21 (↓0.97)	**77.23** (↑24.42)	**76.16** (↑10.46)	**71.80** (↓6.59)	**79.94** (↑0.29)
Fin. Reporting	**69.28** (↑15.21)	73.84 (↑4.17)	79.07 (↑15.89)	**81.10** (↑6.68)	87.02 (↓1.16)	75.87 (↑23.06)	72.87 (↑7.17)	62.69 (↓15.70)	77.23 (↓2.42)
Alter.Invest.	68.99 (↑14.92)	74.90 (↑5.23)	78.97 (↑15.79)	79.94 (↑5.52)	87.50 (↓0.68)	76.94 (↑24.13)	74.90 (↑9.20)	56.98 (↓21.41)	78.88 (↓0.77)
Corp.Issuers	68.31 (↑14.24)	74.32 (↑4.65)	79.26 (↑16.08)	79.36 (↑4.94)	87.02 (↓1.16)	**77.23** (↑24.42)	75.58 (↑9.88)	60.08 (↓18.31)	79.07 (↓0.58)

Table 1: Comparison of accuracy (%) of prompting techniques. 'FinCoT' simultaneously applies expert reasoning blueprints from all CFA domains, while each '(DomainName)' (e.g., 'Economics') row applies domain-specific blueprints individually. (↑/↓) Denote accuracy improvement or decline relative to the SP baseline, colored green for (↑) and red for (↓). **Bold** values highlight the best-performing prompt variant for each model. (*) Indicates that the accuracy improvement among the model-level prompt variants is statistically significant ($p < 0.05$) based on paired bootstrap testing; domain-specific rows are not tested for significance.

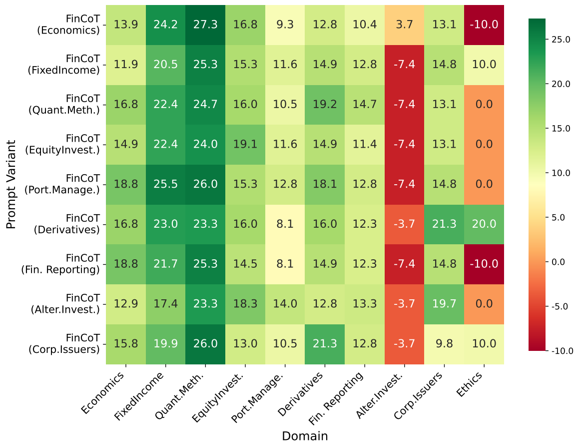
over SP (95% CI [6.30, 9.59], $p < 0.001$), while UST-CoT and ST-CoT also exceed the baseline by +13.76 pp and +16.28 pp. When FinCoT is applied using a single-domain blueprint (e.g., Financial Reporting), the gains increase substantially to +15.21 pp. Similarly, on **Qwen3-8B-Base**, FinCoT delivers the strongest overall boost (+17.35 pp, 95% CI [15.02, 19.77], $p < 0.001$). These findings further underscore the importance of structured domain knowledge, particularly for models lacking instruction or domain alignment.

Instruction Models: We evaluate three prominent instruction-tuned variants. On **Qwen2.5-7B-Instruct**, FinCoT improves accuracy by +4.55 pp over SP, compared to -1.46 pp with UST-CoT and -0.3 pp with ST-CoT. On **Qwen3-8B (Thinker)**, FinCoT yields a slight drop (–0.96 pp), while ST-CoT shows no change and UST-CoT yields a modest +0.87 pp. **Gemma-3-12B-IT**, a strong instruction-tuned baseline (52.81% SP), benefits substantially from all strategies: +25.00 pp (UST-CoT), +23.93 pp (ST-CoT), and +22.77 pp (FinCoT). Notably, domain-specific FinCoT prompts (e.g., Derivatives, Corporate Issuers) provide even larger boosts (+24.42 pp), indicating that blueprint reasoning complements instruction tuning by addressing specialized financial gaps.

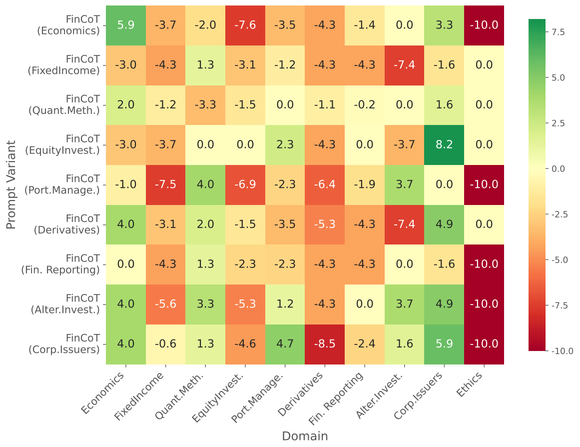
Financial-Specific Models: FinCoT also helps specialized models like **Fin-R1**, confirming that blueprint prompting provides complementary gains beyond fine-tuning. However, for models with strong built-in reasoning, such as **DianJin-R1-7B** and **Fin-o1-8B**, FinCoT offers limited improvement or slight degradation—likely due to conflicts between external scaffolds and internal reasoning routines. These outcomes suggest diminishing returns for CoT prompting when domain alignment and reasoning are already deeply encoded.

Overall, FinCoT is most impactful for models lacking prior task-specific adaptation. By grounding reasoning in structured financial workflows, it bridges key gaps in zero-shot settings without requiring additional tuning. This pattern highlights a trade-off between model internalization and prompt-time controllability. *Future work could explore hybrid strategies that adapt prompting depth based on model alignment.*

Cross-domain behavior of FinCoT: This section examines how pretrained and finance-specific models respond to structured prompting with FinCoT. Each domain-specific blueprint is applied across all CFA domains to evaluate its transferability, and accuracy differences relative to SP are measured. Figure 4a and 4b visualize results



(a)



(b)

Figure 4: Accuracy improvements (%) of each FinCoT domain-specific prompt compared to Standard Prompting (SP). Subfigure (a) shows results on Qwen3-8B-Base (pretrained), while (b) shows results on Fin-o1-8B (finance-specific).

for Qwen3-8B-Base (pretrained) and Fin-o1-8B (finance-specific), while Table 1 provides a comprehensive summary of overall model-level accuracy. On Qwen3-8B-Base, FinCoT generally improves performance, though gains are not universal. Prompts from quantitative domains such as *Derivatives*, *Portfolio Management*, and *Corporate Issuers* yield average gains exceeding +13 pp. The blueprint structure provides inductive guidance that enhances decomposition, formula selection, and financial term alignment. In several domains, Qwen3-8B-Base with FinCoT matches or surpasses the SP baseline of Fin-o1-8B, despite the absence of any task-specific training.

On Fin-o1-8B, gains from FinCoT are more modest, typically within the +1–4 pp range. Minor declines appear in some domains (e.g., Fixed Income, Equity Investments), suggesting that additional scaffolding may interfere with optimized internal reasoning acquired during fine-tuning. Structured prompts may over-specify solutions or reduce instruction-following flexibility.

These findings highlight FinCoT's complementary role. For pretrained models, FinCoT acts as a lightweight yet effective augmentation layer at inference time, reducing the performance gap with fine-tuned models. For already fine-tuned models, careful prompt selection or adaptation may be necessary to preserve existing reasoning strengths without introducing conflict. A broader breakdown of FinCoT performance across additional models is provided in Appendix F, where radar plots illustrate domain-wise patterns and complement the main analysis.

5.2 Efficiency Analysis

Effective deployment of foundation models in financial settings requires balancing verbosity and accuracy while emphasizing efficiency. FinCoT offers a prompt-based alternative to fine-tuned models (Fin-o1, DianJin-R1, Fin-R1), demonstrating similar accuracy as seen in Fig. 5, with token lengths detailed in Appendix D.2 (Tab. 3). Our analysis concentrates on output tokens since prompt encoding occurs once with parallel self-attention $O(n_{\text{in}}^2)$ (rapid) (Vaswani et al., 2023), while decoding involves $O(n_{\text{out}})$ sequential steps with per-step KV-cache updates (high memory usage) (Gu et al., 2018; Shazeer, 2019). Comprehensive input and output token data are provided in Appendix D.1(Tab. 2), and recent decoding accelerations such as speculative sampling (Chen et al., 2023) and FlashAttention (Dao et al., 2022) further underscore output as the primary latency driver.

Output Token Length vs. Accuracy: In deployments, prompting strategies that achieve high accuracy with minimal output length are desirable.

For general-purpose models such as Qwen3-8B-Base and Qwen3-8B (Thinker), FinCoT reduces output length while preserving or improving accuracy. On Qwen3-8B-Base, FinCoT improves accuracy from 78.49% (ST-CoT) to 80.52% (+2.03 pp) while reducing average output from 3.42k to 0.38k tokens, an 8.9× compression. On Qwen3-8B (Thinker), FinCoT maintains 88.18% accuracy with tokens dropping from 1.35k to 1.23k (∼1.1×). Results show that FinCoT's structured blueprints enable more concise, focused reasoning in general-purpose models. Among financial-

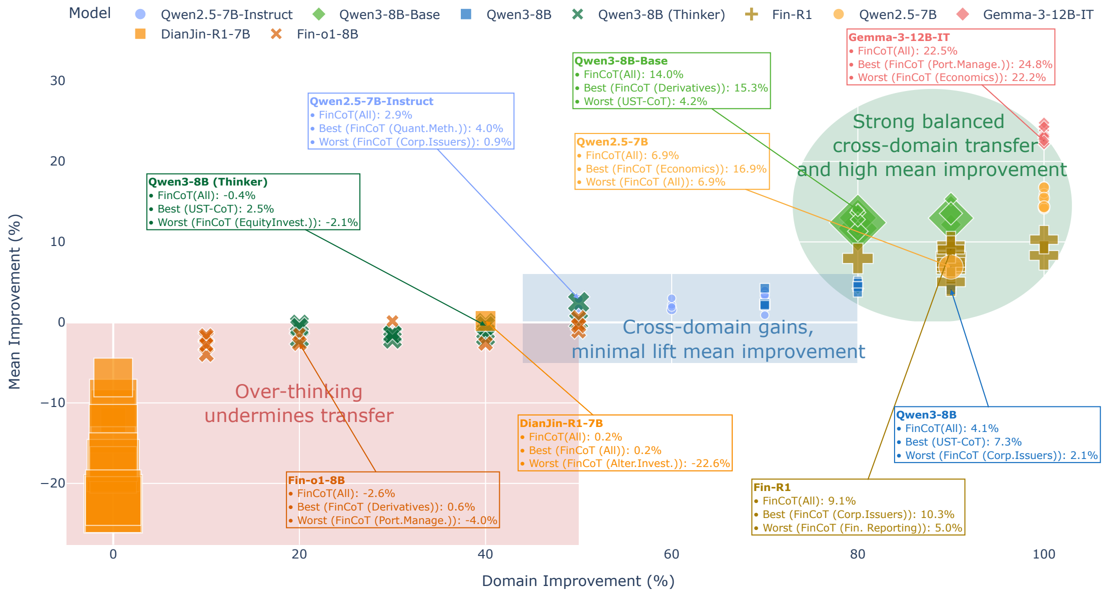


Figure 5: Comparison of prompt strategies across financial domains. Each point represents a domain-method pair, with position indicating accuracy improvement and domain coverage. Circle size encodes generated tokens.

specific models such as DianJin-R1-7B and Fin-o1-8B, FinCoT shows minimal or negative improvement and little benefit in output compression, consistent with fine-tuned models internalizing domain-specific reasoning. Fin-o1-8B suggests that excessive prompt scaffolding may interfere with latent reasoning, reducing effectiveness and leading to overthinking that undermines transfer.

Three behavioral zones emerge from these results. First, models like Gemma-3-12B-IT and Qwen3-8B-Base show high mean improvement and strong cross-domain transfer, reflecting effective generalization. Second, models such as Qwen2.5-7B-Instruct and Qwen3-8B display noticeable cross-domain transfer with lower mean improvement, suggesting limited benefit. Third, models like Fin-o1-8B and Qwen3-8B (Thinker) exhibit low cross-domain adaptability and minimal performance lift, indicating that overly detailed prompting may conflict with internal reasoning.

These findings underscore the importance of aligning the prompting strategy with a model's pre-training or fine-tuning to optimize performance and efficiency. *With limited supervision, FinCoT provides a transparent, cost-effective alternative for enhancing financial reasoning.* For a price–sensitivity analysis of token cost, see Appendix D.2.1 and Fig. 7.

6 Conclusion

We presented **FinCoT**, a zero-shot prompting framework that embeds expert-curated Mermaid diagrams within structured chain-of-thought scaffolds. By grounding reasoning in domain logic, FinCoT bridges human financial workflows with LLM outputs, without model fine-tuning. While broadly applicable, FinCoT yields strong gains for general models (e.g., Qwen3-8B-Base), but more modest or negative effects for instruction or finance-specific models (e.g., Qwen-Thinker, Fin-o1, DianJin-R1), where added structure may interfere with learned reasoning.

Relative to SP, FinCoT improves Qwen3-8B-Base by +17.33 pp and Fin-R1 by +10.08 pp ($p < 0.001$), outperforming some fine-tuned models. In contrast, instruction-tuned models like Qwen3-8B (Thinker) sometimes favor UST-CoT. Cross-domain results show blueprints from quantitative fields transfer best (+27.3 pp on Qwen3-8B-Base). FinCoT also reduces token output by up to $8\times$, offering interpretable, efficient prompting for regulated financial applications. Ultimately, FinCoT suggests that with meticulous prompt design, even general-purpose LLMs can approach the reasoning quality of fine-tuned financial experts in complex decision-making tasks.

Limitations

Our evaluation highlights several limitations. (i) Efficiency gains come mainly from reduced output tokens, but larger inputs from structured templates add cost; overall, FinCoT is still competitive, especially in long-form reasoning. (ii) Domain routing via pre-classification risks template mismatch despite safeguards; adaptive selection methods are needed. (iii) Improvements are uneven, with domains like Alternative Investments and Ethics limited by small samples (∼10–30); larger, balanced benchmarks are required. (iv) Blueprint creation requires expert effort (∼2 hours/domain), and current evaluations are multiple-choice with rubric-based interpretability.

Overall, FinCoT offers structured, auditable reasoning rather than replacing fine-tuned models. Its blueprint methodology and plug-and-play usability demonstrate prompt-level supervision as lightweight knowledge distillation with potential for law, medicine, and engineering.

Acknowledgments

We extend our gratitude to the Hatari Team for generously providing additional GPU resources, which significantly enhanced our computational capacity and enabled the successful execution of large-scale experiments. We also thank Tuntai Boriboonthana (CFA Charterholder) from InnovestX for his expert contributions in verifying blueprint fidelity and conducting random audits to ensure domain classification consistency in GPT-4o outputs. Their support and insights were invaluable to the development and evaluation of this work.

References

300Hours. 2025a. CFA Level 1 Economics Cheat Sheet.

300Hours. 2025b. CFA Level 1 Equity Investments: Our Cheat Sheet.

AnalystPrep. 2025. Economics CFA Level 1 Essential Review Summary.

Arya.ai. 2024. 5 Best Large Language Models (LLMs) for Financial Analysis.

Marco Avellaneda and Sasha Stoikov. 2008. High-Frequency Trading in a Limit Order Book. In *Proceedings of the 12th International Conference on Quantitative Finance*, pages 1–9. Cornell University. Preprint: SSRN 1085964.

Daniele De Bari, Giacomo Garaccione, Riccardo Coppola, Marco Torchiano, and Luca Ardito. 2024. Evaluating large language models in exercises of uml class diagram modeling. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '24)*, pages 393–399. ACM.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Gagan Bhatia, El Moatez Billah Nagoudi, Hasan Cavusoglu, and Muhammad Abdul-Mageed. 2024. Fintral: A family of gpt-4 level multimodal financial large language models. *Preprint*, arXiv:2402.10986.

Fischer Black and Myron Scholes. 1973a. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637–654.

Fischer Black and Myron Scholes. 1973b. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3):637–654.

Bloomberg. 2025. Bloomberg US Aggregate Bond Index.

Financial Stability Board. 2025. OTC Derivatives Market Reforms. Technical report, Financial Stability Board.

Zvi Bodie, Alex Kane, and Alan J. Marcus. 2017. *Investments*, 11th edition. McGraw-Hill Education, New York, NY.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*.

CAIA Association. 2025. Alternative Investment Management.

Ethan Callanan, Amarachi Mbakwe, Antony Papadimitriou, Yulong Pei, Mathieu Sibue, Xiaodan Zhu, Zhiqiang Ma, Xiaomo Liu, and Sameena Shah. 2024. Can GPT models be Financial Analysts? An Evaluation of ChatGPT and GPT-4 on mock CFA Exams. In *Proceedings of the Eighth Financial Technology and Natural Language Processing and the 1st Agent AI for Scenario Planning*, pages 23–32, Jeju, South Korea. -.

CFA Institute. 2024. *CFA Program Curriculum 2025: Level II, Volume 2 – Economics*. Wiley. Includes coverage of classical, neoclassical, and endogenous growth models.

CFA Institute. 2025a. CFA Program Curriculum: Alternative Investments.

CFA Institute. 2025b. CFA Program Curriculum: Equity Investments.

CFA Institute. 2025c. CFA Program Curriculum: Equity Investments & Fixed Income.

CFA Institute. 2025d. CFA Program Curriculum: Financial Reporting and Analysis.

CFA Institute. 2025e. CFA Program Curriculum: Fixed Income (Levels I & II).

CFA Institute. 2025f. CFA Program Curriculum: Level II – Derivatives.

CFA Institute. 2025g. CFA Program Curriculum: Portfolio Management.

CFA Institute. 2025h. *CFA Program Curriculum: Quantitative Methods*. CFA Institute, Charlottesville, VA. Levels I & II; covers TVM, probability, hypothesis testing, regression, portfolio stats.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *Preprint*, arXiv:2302.01318.

Aswath Damodaran. 2012. *Investment Valuation: Tools and Techniques for Determining the Value of Any Asset*, 3rd edition. Wiley, Hoboken, NJ.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Preprint*, arXiv:2205.14135.

Ilia Dichev. 2017. On the Balance Sheet-Based Model of Financial Distress Prediction. *The Accounting Review*, 92(4):1125–1152.

Weihua Du, Yiming Yang, and Sean Welleck. 2025. Optimizing Temperature for Language Models with Multi-Sample Inference. *Preprint*, arXiv:2502.05234.

Efficient Learning. 2025. Program Overview: CFA Economics.

Bradley Efron and Robert J Tibshirani. 1994. *An Introduction to the Bootstrap*. CRC press.

Gregor Engels, Reiko Heckel, and Stefan Sauer. 2000. UML – a universal modeling language? In *Proceedings of the 21st International Conference on Application and Theory of Petri Nets (ICATPN 2000)*, volume 1825 of *Lecture Notes in Computer Science*, pages 24–38. Springer, Heidelberg.

Frank J. Fabozzi. 2012. *Bond Markets, Analysis, and Strategies*, 8th edition. Pearson/Prentice Hall.

Tingchen Fu, Jiawei Gu, Yafu Li, Xiaoye Qu, and Yu Cheng. 2025. Scaling reasoning, losing control: Evaluating instruction following in large reasoning models. *Preprint*, arXiv:2505.14810.

Brian Gordon. 2020a. CFA Exam Level 1 Economics Lecture. `https://www.youtube.com/watch?v=SvqKJnN4Tbo`.

Brian Gordon. 2020b. CFA Level I: Equity Investments Preview. `https://www.youtube.com/watch?v=SvqKJnN4Tbo`.

Brian Gordon. 2020c. CFA Level I: Equity Investments Revision. `https://www.youtube.com/watch?v=SvqKJnN4Tbp`.

Richard C. Grinold and Ronald N. Kahn. 2000. *Active Portfolio Management*. McGraw-Hill.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. *Preprint*, arXiv:1711.02281.

Chetan Harsha, Karmvir Singh Phogat, Sridhar Dasaratha, Sai Akhil Puranam, and Shashishekar Ramakrishna. 2025. Synthetic data generation using large language models for financial question answering. In *Proceedings of the Joint Workshop of the 9th Financial Technology and Natural Language Processing (FinNLP), the 6th Financial Narrative Processing (FNP), and the 1st Workshop on Large Language Models for Finance and Legal (LLMFinLegal)*, pages 76–95, Abu Dhabi, UAE. Association for Computational Linguistics.

Yi Hu, Haotong Yang, Zhouchen Lin, and Muhan Zhang. 2023. Code Prompting: a Neural Symbolic Method for Complex Reasoning in Large Language Models. *Preprint*, arXiv:2305.18507.

John C. Hull. 2017. *Options, Futures, and Other Derivatives*, 10th edition. Pearson.

Investopedia. 2025. Aggregate Demand and Supply (AD–AS) Model.

Investopedia. 2025. Asset-Backed Security (ABS).

Investopedia. 2025a. Balance of Payments.

Investopedia. 2025b. Business Cycle: Definition & 4 Phases.

Investopedia. 2025a. Consumer Price Index (CPI).

Investopedia. 2025b. Convexity in Bonds: Definition, Meaning, and Examples.

Investopedia. 2025c. Coupon Rate Definition.

Investopedia. 2025d. Credit Spread Definition.

Investopedia. 2025e. Duration Definition and Its Use in Fixed Income Investing.

Investopedia. 2025. Exchange Rate Definition.

Investopedia. 2025a. Financial Statements: List of Types and How to Read Them.

Investopedia. 2025b. Fixed-Income Security Definition, Types, and Examples.

Investopedia. 2025a. Fundamental Analysis: Principles, Types, and How to Use It.

Investopedia. 2025b. Government Regulations: Do They Help Businesses?

Investopedia. 2025c. Gross Domestic Product (GDP) Formula.

Investopedia. 2025d. How to Read a Financial Analysis Report.

Investopedia. 2025. Interest Rate Definition & Impact.

Investopedia. 2025a. Law of Supply and Demand.

Investopedia. 2025b. Monetary Policy Definition.

Investopedia. 2025c. Phillips Curve: Trade-Off Between Inflation and Unemployment.

Investopedia. 2025d. Porter's Five Forces Definition.

Investopedia. 2025e. Price Elasticity of Demand.

Investopedia. 2025f. Simple vs. Compound Interest: Definition and Formulas. Explains TVM basics and formula examples.

Investopedia. 2025g. The Rule of 72: What It Is and How to Use It in Investing. Provides quick doubling-time approximation.

Investopedia. 2025a. When to Rebalance a Bond Portfolio.

Investopedia. 2025b. Yield Curve Definition & Types.

iPassFinanceExams. 2025. CFA Economics Study Tips.

Raymond James. 2025. Fixed Income Strategies.

Doohyuk Jang, Yoonjeon Kim, Chanjae Park, Hyun Ryu, and Eunho Yang. 2025. Reasoning model is stubborn: Diagnosing instruction overriding in reasoning models. *Preprint*, arXiv:2505.17225.

Robert A. Jarrow and Stuart M. Turnbull. 1995. Pricing Derivatives on Financial Securities Subject to Credit Risk. *Journal of Finance*, 50(1):53–85.

Robert A. Jarrow and Stuart M. Turnbull. 1996. *Derivative Securities*. South-Western College Publishing.

Kaplan Schweser. 2025. Level I CFA Economics Study Tips.

Zixuan Ke, Yifei Ming, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. Demystifying Domain-adaptive Post-training for Financial LLMs. *Preprint*, arXiv:2501.04961.

Jean Lee, Nicholas Stevens, and Soyeon Caren Han. 2025. Large language models in finance (finllms). *Neural Computing and Applications*.

Sangmin Lee, Suzie Oh, Saeran Park, Guijin Son, and Pilsung Kang. 2024. FINALE : Finance domain instruction-tuning dataset with high-quality rationales via chain-of-thought prompting. In *Proceedings of the Eighth Financial Technology and Natural Language Processing and the 1st Agent AI for Scenario Planning*, pages 89–106, Jeju, South Korea. -.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. *Preprint*, arXiv:2206.14858.

Xiaomin Li, Zhou Yu, Zhiwei Zhang, Xupeng Chen, Ziji Zhang, Yingying Zhuang, Narayanan Sadagopan, and Anurag Beniwal. 2025. When thinking fails: The pitfalls of reasoning for instruction-following in llms. *Preprint*, arXiv:2505.11423.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, Chao Li, Sheng Xu, Dezhi Chen, Yun Chen, Zuo Bai, and Liwen Zhang. 2025. Fin-R1: A Large Language Model for Financial Reasoning through Reinforcement Learning. *Preprint*, arXiv:2503.16252.

Xuezhi Ma, Yan Zhou, Yining Wang, and 1 others. 2023. Financialqa: A reasoning benchmark for financial question answering. *arXiv preprint arXiv:2302.07304*.

Harry Markowitz. 1952. *Portfolio Selection*, volume 7 of *Journal of Finance Classics*. Wiley, New York.

Andrew Metrick and Ayako Yasuda. 2010. *Private Equity and Venture Capital*. Wiley.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges. *arXiv preprint arXiv:2406.11903*.

Krishna G. Palepu, Paul M. Healy, Sue Wright, Michael Bradbury, and Jeff Coulton. 2013. *Business Analysis and Valuation: Using Financial Statements*. Cengage.

Stephen H. Penman. 2012. *Financial Statement Analysis and Security Valuation*, 5th edition. McGraw-Hill Education.

Jerald E. Pinto, Elaine Henry, Thomas R. Robinson, and John D. Stowe. 2015. *Equity Asset Valuation*. CFA Institute Investment Series. Wiley, Hoboken, NJ.

Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Han Yi, Yilun Zhao, Jimin Huang, Qianqian Xie, and Jian yun Nie. 2025. Fino1: On the transferability of reasoning-enhanced llms and reinforcement learning to finance. *Preprint*, arXiv:2502.08127.

Matthew Renze and Erhan Guven. 2024. The Benefits of a Concise Chain of Thought on Problem-Solving in Large Language Models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, page 476–483. IEEE.

R. Tyrrell Rockafellar and Stanislav Uryasev. 2000. Optimization of Conditional Value-at-Risk. Technical Report IFOR Technical Report TR00-27, University of Florida.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *Preprint*, arXiv:1911.02150.

Clyde P. Stickney, Paul R. Brown, and James M. Wahlen. 2007. *Financial Reporting, Financial Statement Analysis, and Valuation: A Strategic Perspective*. Prentice Hall.

Knut Sveidqvist and contributors. 2025. Mermaid: A javascript-based diagramming and charting tool. GitHub repository.

Dominique Tavella and Curt Randall. 2000. *Pricing Financial Instruments: The Finite Difference Method*. Wiley.

Bruce Tuckman and Angel Serrat. 2011. *Fixed Income Securities: Tools for Today's Markets*, 3rd edition. Wiley.

UWorld Finance. 2025a. CFA® Economics: Syllabus & Sample Questions.

UWorld Finance. 2025b. CFA® Finance Study Resources.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas Pfister. 2023. Universal self-adaptive prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7437–7462, Singapore. Association for Computational Linguistics.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *Preprint*, arXiv:2109.01652.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Preprint*, arXiv:2201.11903.

Bo Wen and Xin Zhang. 2025. Enhancing reasoning to adapt large language models for domain-specific applications. *Preprint*, arXiv:2502.04384.

Wikipedia contributors. 2025. Time Value of Money. Overview of present and future value concepts.

Jeffrey M. Wooldridge. 2013. *Introductory Econometrics: A Modern Approach*, 5th edition. Cengage Learning, Boston, MA.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

Yi Yang, Mark Christopher Siy UY, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *Preprint*, arXiv:2006.08097.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Zijun Yao, Yantao Liu, Yanxu Chen, Jianhui Chen, Junfeng Fang, Lei Hou, Juanzi Li, and Tat-Seng Chua. 2025. Are reasoning models more prone to hallucination? *Preprint*, arXiv:2505.23646.

Jie Zhu, Qian Chen, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. 2025. Dianjin-r1: Evaluating and enhancing financial reasoning in large language models. *Preprint*, arXiv:2504.15716.

A Expert Reasoning Blueprints

Economics

```
***Economics:***
```mermaid
graph TD;
  A[Step 1: Question Breakdown] -->|Extract key terms| A1{Identify Topic}
  A1 -->|Micro: Supply & Demand, Market Structures| A2
  A1 -->|Macro: GDP, Growth, Policy, Trade| A3
  A1 -->|Currency & Regulation| A4

  A2 --> B1[Identify model: Elasticity, Cost Curves, Shutdown Points]
  A3 --> B2[Map to AD-AS, Business Cycles, Growth Theories]
  A4 --> B3[Assess Exchange Rates, Trade, Capital Flows, Regulation]

  B1 -->|Check for formula or concept?| C{Numerical or Conceptual}
  B2 --> C
  B3 --> C

  C -->|Numerical| D1[Extract data, apply formulas, check assumptions]
  C -->|Conceptual| D2[Analyze cause-effect, policy impact]

  D1 --> E[Step 4: Solution Development]
  D2 --> E
  E -->|Construct structured response| E1(Core insight + economic rationale)
  E -->|Consider alternative scenarios| E2(Assess different possibilities)

  E1 --> F[Step 5: Answer Validation]
  E2 --> F
  F -->|Check logic, principles, and assumptions| F1(Verify consistency)
  F1 -->|Ensure completeness & clarity| F2(Confirm answer structure)
```
```

Explanation: Step-1: Question Breakdown (A) – Extract key terms by parsing the question to see whether it focuses on microeconomics, macroeconomics, or currency/regulation topics (300Hours, 2025a; UWorld Finance, 2025a).

Step-2: Identify Topic (A1) – Microeconomics (A2): Focus on supply & demand mechanisms and market structures such as perfect competition, monopoly, oligopoly, and monopolistic competition (Investopedia, 2025a; 300Hours, 2025a). – Macroeconomics (A3): Consider aggregate demand–aggregate supply analysis, phases of the business cycle (expansion, peak, contraction, trough), and growth models (Solow, endogenous growth) (Investopedia, 2025b; CFA Institute, 2024). – Currency & Regulation (A4): Examine exchange-rate regimes (floating vs. pegged), trade balances, capital-flow impacts, and relevant government policies (Investopedia, 2025,b).

Step-3: Model Selection or Strategy Mapping (B1–B3) – Micro Models (B1): Choose elasticity calculations and cost-curve analysis (marginal/average cost, shutdown point) for supply–demand or firm-behavior questions (Investopedia, 2025e). – Macro Frameworks (B2): Apply AD–AS curves, Phillips-curve trade-offs, or business-cycle indicators to frame policy or growth analysis (Investopedia, 2025,c). – FX & Regulation (B3): Use exchange-rate determination models, balance-of-payments analysis, or regulatory impact frameworks for currency/trade questions (Investopedia, 2025a).

Step-4: Determine Numerical vs. Conceptual Approach (C) – Numerical (D1): Gather the relevant data (prices, quantities, rates), apply formulae (e.g., Elasticity $= \frac{\%\Delta Q_n}{\%\Delta P}$, GDP $= C + I + G + (X - M)$), and verify assumptions (Investopedia, 2025c). – Conceptual (D2): Construct a narrative explaining cause–effect relationships (e.g., how a monetary-policy change shifts AD or how trade barriers affect capital flows) (Investopedia, 2025b).

Step-5: Solution Development (E) – Structured Response (E1): State the core economic insight first, then provide the step-by-step rationale linking theory to the question context (Kaplan Schweser, 2025). – Alternative Scenarios (E2): Where relevant, outline best-case, base-case, and worst-case scenarios or show how supply–demand curves shift under different assumptions (iPassFinanceExams, 2025).

Step-6: Answer Validation (F) – Verify Consistency (F1): Check that numerical answers satisfy boundary conditions (e.g., correct sign on elasticity, GDP component sums). – Confirm Clarity (F2): Ensure your explanation is complete, logically ordered, and clearly communicates both the result and its limitations (UWorld Finance, 2025b).

Source: 300Hours CFA Level 1 Economics Cheat Sheet (300Hours, 2025a), UWorld Finance's CFA® Economics: Syllabus & Sample Questions (UWorld Finance, 2025a), Kaplan Schweser's Level I Economics tips (Kaplan Schweser, 2025), Efficient Learning's CFA Economics overview (Efficient Learning, 2025), iPass Finance Exams' study guide (iPassFinanceExams, 2025), AnalystPrep's essential review summary (AnalystPrep, 2025), and key Investopedia articles on supply and demand (Investopedia, 2025a), GDP (Investopedia, 2025c), and business cycles (Investopedia, 2025b) as well as Prof. Brian Gordon's CFA Exam Level 1 Economics video (Gordon, 2020a).

Fixed Income

````
***Fixed Income:***
```mermaid
graph TD
    A[Purpose and Scope] --> B3[Analyze Macro Conditions]
    B --> C[Assess Bond Features]
    C --> D[Risk and Yield Analysis]
    D --> E[Develop Recommendations]
    E --> F[Review Performance]

    %% Notes and detailed steps
    A --> |Set objectives| B
    B --> |Review interest rates and inflation| C
    C --> |Focus on duration, spread| D
    D --> |Assess scenarios| E
```
````

Explanation: Step-1: Purpose and Scope – Define the investment objective–income generation, capital preservation, hedging, or total return and establish portfolio constraints and benchmarks such as target yield, duration limits, credit quality floors, or sector allocation guidelines (Investopedia, 2025b).

Step-2: Analyze Macro Conditions – Examine current and forecast interest rate paths, since rising rates erode bond prices and falling rates support them (Investopedia, 2025); monitor inflation indicators (CPI, PPI) to gauge real yield trends (Investopedia, 2025a); and assess yield-curve shapes (normal, inverted, flat) for economic turning points and yield-curve trade opportunities (Investopedia, 2025b).

Step-3: Assess Bond Features – Identify bond type government, corporate, municipal, structured products (ABS/MBS) and note any embedded options (callable, putable, convertible) (Investopedia, 2025); review coupon structure (fixed vs. floating), payment frequency, and maturity to understand cash-flow timing and reinvestment risk (Investopedia, 2025c).

Step-4. Risk and Yield Analysis – Calculate duration to estimate price sensitivity to yield changes (Investopedia, 2025e) and convexity for non-linear price effects (Investopedia, 2025b); analyze credit spreads over benchmarks to gauge default and liquidity risk (Investopedia, 2025d); and stress-test the portfolio under parallel shifts, steepeners, and flatteners to assess P&L impacts.

Step-5: Develop Recommendations – Formulate strategies such as adjusting overall duration (shorten if rates are likely to rise), implementing barbell or laddered maturity structures, or choosing bullet portfolios to manage reinvestment and rate risk (James, 2025).

Step-6: Review Performance – Track total returns (price changes plus coupon income) against benchmarks like the Bloomberg US Aggregate Bond Index (Bloomberg, 2025); perform attribution analysis to decompose yield carry, curve roll-down, and spread effects; and revisit assumptions and rebalance when market conditions or issuer fundamentals change (Investopedia, 2025a).

Source: CFA Program Curriculum for Fixed Income (CFA Institute, 2025e), Fabozzi's Bond Markets, Analysis, and Strategies (Fabozzi, 2012), Tuckman & Serrat's Fixed Income Securities (Tuckman and Serrat, 2011), Jarrow & Turnbull's credit-risk derivatives pricing (Jarrow and Turnbull, 1995), Basel Committee papers on credit risk, Investopedia articles on fixed-income concepts (Investopedia, 2025b,b,e), Reuters coverage of convexity risk, and Dichev's balance sheet model for distress prediction (Dichev, 2017).

Quantitative Methods

```
***Quantitative Methods:***
```mermaid
graph TD
    A["Articulating Purpose and Context"] --> B["Collecting Input Data"]
    B --> C["Processing and Cleaning Data"]
    C --> D["Selecting Quantitative Models and Tools"]
    D --> E["Estimating Parameters and Testing Hypotheses"]
    E --> F["Interpreting Results and Communicating Findings"]
    F --> G["Monitoring and Model Reassessment"]
```
```

Explanation: Step-1: Articulating Purpose and Context (A) Define the research question time value of money calculations, probability distributions, hypothesis testing, regression analysis, or portfolio statistics–and establish the CFA application context (market efficiency, risk estimation, cash-flow forecasting) (CFA Institute, 2025h).

Step-2: Collecting Input Data (B) Gather historical returns, economic indicators, financial statements, and market data from reputable sources; ensure relevance by matching data to the chosen objective (e.g., interest rates for TVM, volatility for risk models) (Investopedia, 2025f).

Step-3: Processing and Cleaning Data (C) Perform data quality checks and remove outliers, handle missing values, confirm consistency–and apply transformations (normalization, log-transforms) before analysis (Wooldridge, 2013).

Step-4: Selecting Quantitative Models and Tools (D) Choose appropriate models–ARIMA for time series, linear/multivariate regression, probability distributions, or Monte Carlo simulation–and leverage CFA-recommended software or spreadsheet tools (Damodaran, 2012; Investopedia, 2025g).

Step-5: Estimating Parameters and Testing Hypotheses (E) Estimate model parameters via regression or maximum likelihood; conduct t-tests, F-tests, or chi-square tests to validate assumptions and results, with Level II emphasis on multivariate regression and sensitivity analysis (Wooldridge, 2013).

Step-6: Interpreting Results and Communicating Findings (F) Translate coefficients, p-values, and confidence intervals into actionable investment insights; prepare clear visual aids (charts, tables) to support recommendations (Bodie et al., 2017).

Step-7: Monitoring and Model Reassessment (G) Track out-of-sample performance against benchmarks; update models as new data arrive, reassess assumptions, and recalibrate parameters to maintain relevance (Wikipedia contributors, 2025).

Source: CFA Program Curriculum: Quantitative Methods (CFA Institute, 2025h), Wooldridge's Introductory Econometrics (Wooldridge, 2013), Damodaran's Investment Valuation (Damodaran, 2012), and Bodie, Kane, & Marcus's Investments (Bodie et al., 2017).

Equity Investments

```
***Equity Investing:***
```mermaid
graph TD
    A[Objective Setting] --> B[Market and Sector Insights]
    B --> C[Industry Competitive Analysis]
    C --> D[Company Review]
    D --> E[Valuation and Risks]
    E --> F[Investment Decision]

    %% Step-specific highlights
    B --> |Look at growth patterns| C
    C --> |Evaluate competitors' positions| D
    D --> |Check financial health| E
    E --> |Combine insights into strategy| F
```
```

Explanation Step-1: Objective Setting (A) Define your investment objectives–capital appreciation, dividend income, or total return in line with your risk tolerance and investment horizon; consider constraints such as liquidity needs, tax implications, regulatory requirements, and any specific mandates (CFA Institute, 2025b).

Step-2: Market and Sector Insights (B) Assess macro indicators (GDP growth, interest rates, inflation) to gauge the overall market environment and identify sectors poised for growth or decline based on economic trends, technological shifts, and consumer behavior (Investopedia, 2025a).

Step-3: Industry Competitive Analysis (C) Apply Porter's Five Forces to evaluate industry attractiveness–competitive rivalry, threat of new entrants, bargaining power of suppliers and buyers, and substitute threats–and assess each firm's market share and competitive moat (Investopedia, 2025d).

Step-4: Company Review (D) Examine financial statements (income statement, balance sheet, cash flows) to measure profitability, liquidity, and stability; evaluate management's track record and strategic vision; and review corporate governance structures to ensure alignment with shareholder interests (Investopedia, 2025d; Bodie et al., 2017).

Step-5: Valuation and Risks (E) Use valuation methods–Discounted Cash Flow (DCF), Price-to-Earnings (P/E) ratios, Dividend Discount Models (DDM)–to estimate intrinsic value; identify key risks such as market volatility, operational challenges, regulatory changes, and competitive threats (Pinto et al., 2015).

Step-6: Investment Decision (F) Formulate your Buy, Hold, or Sell recommendation based on the above analyses and determine how the position fits within the broader portfolio–considering diversification, correlation, and overall risk–return objectives (300Hours, 2025b).

Source: CFA Program Curriculum's Equity Investments module (CFA Institute, 2025b), Investopedia's guides on fundamental analysis (Investopedia, 2025a), Porter's Five Forces stock analysis (Investopedia, 2025d), and reading financial reports (Investopedia, 2025d), 300Hours' CFA Level 1 Equity Cheat Sheet (300Hours, 2025b), Bodie, Kane & Marcus's Investments (Bodie et al., 2017), Pinto et al.'s Equity Asset Valuation (Pinto et al., 2015), and CFA Level I Equity video lectures by Prof. Brian Gordon (Gordon, 2020b,c).

Portfolio Management

```
***Portfolio Management:***
```mermaid
graph TD
    A["Define Investment Objectives"] --> B["Establish Investment Constraints"]
    B --> C["Develop Strategic Asset Allocation"]
    C --> D["Incorporate Tactical Adjustments"]
    D --> E["Select and Optimize Securities"]
    E --> F["Execute Implementation and Trading"]
    F --> G["Measure Performance and Attribution"]
    G --> H["Monitor Risk and Compliance"]
    H --> I["Rebalance and Adjust Portfolio"]
```
```

Explanation: Step-1: Define Investment Objectives – Clarify whether the portfolio is aimed at capital growth, income generation, or a balanced mix. Specify expected returns, risk tolerance, and liquidity needs. This step forms the foundation for aligning investment strategy with client mandates (CFA Institute, 2025g).

Step-2: Establish Investment Constraints – Define legal, regulatory, tax, and unique client considerations such as ESG preferences or geographic limits. These constraints ensure feasibility and compliance of portfolio design (CFA Institute, 2025g).

Step-3: Develop Strategic Asset Allocation – Allocate across major asset classes (equities, fixed income, alternatives, cash) based on expected returns and risk tolerance. Use models from Modern Portfolio Theory and CAPM to inform allocation (Markowitz, 1952; Bodie et al., 2017).

Step-4: Incorporate Tactical Adjustments – Introduce short-term adjustments to the strategic allocation based on market outlook or economic indicators. These shifts aim to enhance returns through asset or sector rotation (Grinold and Kahn, 2000).

Step-5: Select and Optimize Securities – Apply quantitative screens and qualitative research to choose securities. Use optimization techniques such as mean-variance optimization or the Black-Litterman model to maximize risk-adjusted returns (Bodie et al., 2017; Grinold and Kahn, 2000).

Step-6: Execute Implementation and Trading – Implement trade strategies that minimize costs and slippage, considering market impact and liquidity. Align execution with strategic intentions (CFA Institute, 2025g).

Step-7: Measure Performance and Attribution – Track performance using return metrics, Sharpe ratio, alpha, and beta. Perform attribution to evaluate decisions across asset allocation, sector, and security selection (Grinold and Kahn, 2000).

Step-8: Monitor Risk and Compliance – Use tools like Value-at-Risk (VaR), stress testing, and tracking error to monitor portfolio risk. Ensure compliance with constraints and regulations (CFA Institute, 2025g).

Step-9: Rebalance and Adjust Portfolio – Periodically adjust the portfolio to maintain alignment with the strategic asset allocation as market conditions evolve.

Source: CFA Program Curriculum's Portfolio Management module (CFA Institute, 2025g), Bodie, Kane & Marcus's Investments for portfolio theory and risk-return optimization (Bodie et al., 2017), Grinold & Kahn's Active Portfolio Management for advanced attribution and optimization techniques (Grinold and Kahn, 2000), and Markowitz's seminal Portfolio Selection on diversification and risk-adjusted returns (Markowitz, 1952).

Derivatives

```
***Derivatives:***
```mermaid
graph TD
    A[Define Objective and Context] --> B[Identify Derivative Instrument]
    B --> C[Understand Contract Specifications]
    C --> D[Gather Market Data]
    D --> E[Apply Valuation Models]
    E --> F[Assess Risks: Market, Counterparty, etc.]
    F --> G[Construct Payoff Diagrams or Strategies]
    G --> H[Interpret Results and Make Recommendations]
    H --> I[Review, Monitor, and Adjust Strategies]

    %% Example labels or notes (optional)
    A --> |Hedging, speculation, arbitrage| B
    C --> |Features like notional amount, expiration| D
    D --> |Market prices, volatility, risk-free rates| E
    F --> |Sensitivity to Greeks: Delta, Gamma, Vega, etc.| G
    H --> |Adjust based on changing market conditions| I
````
```

Explanation: Step-1: Define Objective and Context – Clarify the purpose of using derivatives: hedging, speculation, or arbitrage. Identify relevant constraints, such as regulatory limitations or portfolio mandates (CFA Institute, 2025f; Hull, 2017).

Step-2: Identify Derivative Instrument – Choose the appropriate derivative: options, futures, forwards, swaps, or structured/exotic products (Jarrow and Turnbull, 1996).

Step-3: Understand Contract Specifications – Review contract parameters, including the underlying asset, strike price, expiration, settlement method (physical or cash), and style (European, American) (CFA Institute, 2025f).

Step-4: Gather Market Data – Collect input variables such as spot price, volatility, risk-free rate, dividends, and term structure of interest rates (Hull, 2017).

Step-5: Apply Valuation Models – Apply pricing frameworks suited to the derivative:

- Black-Scholes model for European options (Black and Scholes, 1973b).
- Binomial Tree for path-dependent or American-style options (Hull, 2017).
- Cost-of-carry model for futures and forwards (Jarrow and Turnbull, 1996).
- Finite-difference methods for complex derivatives (Tavella and Randall, 2000).

Step-6: Assess Risks – Use Greeks (Delta, Gamma, Vega, Theta, Rho) to evaluate sensitivity to market factors. Consider counterparty and credit risk in OTC markets (Hull, 2017; Board, 2025).

Step-7: Construct Payoff Diagrams or Strategies – Visualize outcomes using payoff graphs. Design strategies such as straddles, collars, or protective puts based on desired exposure (Hull, 2017).

Step-8: Interpret Results and Make Recommendations – Translate model output into actionable insights: confirm hedge effectiveness, profit potential, or risk exposure.

Step-9: Review, Monitor, and Adjust Strategies – Continuously monitor derivative positions in light of market conditions, risk metrics, and investment objectives (Board, 2025).

Source: Based on Hull's comprehensive treatment of markets and pricing models (Hull, 2017), the CFA Institute Level II Derivatives readings (CFA Institute, 2025f), Black & Scholes's seminal option pricing model (Black and Scholes, 1973b), Jarrow & Turnbull's practical engineering perspective (Jarrow and Turnbull, 1996), Tavella & Randall's numerical finite-difference techniques (Tavella and Randall, 2000), and the Basel Committee's OTC derivatives reforms for regulatory context (Board, 2025).

Financial Reporting

````
***Financial Reporting:**
```mermaid
graph TD
A[Articulating Purpose and Context] --> B[Collecting Input Data]
    B --> C[Processing Data]
    C --> D[Analyzing and Interpreting Processed Data]
    D --> E[Developing and Communicating Conclusions]
    E --> F[Doing Follow-Up]

    A --> |Defines goals, tools, and audience| B
    B --> |Gather data on economy and industry| C
    C --> |Use tools like ratios and charts| D
    D --> |Interpret data for conclusions| E
    F --> |Periodic review and iteration| A
```
````

Explanation: Step-1: Articulating Purpose and Context
Define the objectives of the analysis–such as assessing profitability, liquidity, or solvency. Identify stakeholders (e.g., investors, creditors, management) and tailor the analysis to their needs. Set the framework, including accounting standards (IFRS or US GAAP) and the time horizon (CFA Institute, 2025d).

Step-2: Collecting Input Data
Gather primary financial statements: income statement, balance sheet, and cash flow statement. Supplement this with industry benchmarks and macroeconomic data. Ensure the quality, accuracy, and completeness of all collected data (Investopedia, 2025a).

Step-3: Processing Data
Standardize data for comparability by adjusting for non-recurring items or differences in accounting policies. Compute financial ratios such as ROE, current ratio, and debt-to-equity. Use visualizations (e.g., charts, graphs) to uncover trends and patterns (Stickney et al., 2007).

Step-4: Analyzing and Interpreting Processed Data
Assess financial health by interpreting computed ratios. Benchmark against peer companies and industry averages. Identify strengths and weaknesses to determine strategic implications (Palepu et al., 2013).

Step-5: Developing and Communicating Conclusions
Summarize findings in a clear, concise report. Offer actionable recommendations–e.g., restructuring debt or improving efficiency. Tailor communication style and depth to fit the audience, whether board members, analysts, or external investors.

Step-6: Doing Follow-Up
Monitor outcomes of implemented actions and assess whether financial targets are met. Update the analysis regularly with new data and refine recommendations. Incorporate feedback to improve future analysis cycles.

Source: CFA Program Curriculum's Financial Reporting and Analysis readings covering ratio analysis, cash flow analysis, and IFRS/GAAP standards (CFA Institute, 2025d) alongside Investopedia's overview of financial statement components (Investopedia, 2025a), Paul R. Brown's strategic perspective on statement analysis and valuation (Stickney et al., 2007), and Palepu & Healy's MBA-level treatment of business analysis and valuation using financial statements (Palepu et al., 2013).

Alternative Investments

```
***Alternative Investments:***
```mermaid
graph TD
    A["Define Investment Objectives and Mandate"] --> B["Identify Alternative Asset Classes"]
    B --> C["Conduct Manager and Strategy Due Diligence"]
    C --> D["Perform Valuation and Pricing Analysis"]
    D --> E["Assess Risk and Liquidity"]
    E --> F["Allocate Alternatives in Portfolio"]
    F --> G["Monitor Performance and Rebalance"]
```
```

Explanation: Step-1: Define Investment Objectives and Mandate – Clarify the purpose of including alternative investments–whether for diversification, higher return potential, or hedging against market volatility. Define constraints such as time horizon, liquidity needs, regulatory frameworks, and risk tolerance (CFA Institute, 2025a).

Step-2: Identify Alternative Asset Classes – Explore the universe of alternatives, including hedge funds, private equity, real estate, infrastructure, commodities, and venture capital. Assess how each class contributes to portfolio diversification via low correlation to traditional assets (Bodie et al., 2017; CAIA Association, 2025).

Step-3: Conduct Manager and Strategy Due Diligence – Evaluate managers based on their track record, investment philosophy, risk management, and operational quality. Understand the specific strategies (e.g., long/short, event-driven, global macro) and their alignment with investment mandates (CAIA Association, 2025; Metrick and Yasuda, 2010).

Step-4: Perform Valuation and Pricing Analysis – Address the unique valuation challenges of illiquid assets. Use models like discounted cash flow (DCF) or mark-to-model, and apply appropriate liquidity or opacity discounts. Compare performance with custom or market benchmarks (Metrick and Yasuda, 2010).

Step-5: Assess Risk and Liquidity – Identify key risks including market, manager, and operational risks. Analyze downside risk and tail event exposure. Evaluate liquidity risks, such as lock-up periods and redemption windows, that may affect rebalancing ability (CFA Institute, 2025a).

Step-6: Allocate Alternatives in Portfolio – Determine appropriate weighting of alternative assets, guided by expected return, volatility, and correlation with traditional investments. Make strategic allocation decisions with room for tactical adjustments based on market conditions (Bodie et al., 2017).

Step-7: Monitor Performance and Rebalance – Track returns over time, evaluate them against relevant benchmarks, and assess if performance remains consistent with expectations. Rebalance periodically to ensure alignment with objectives, risk profile, and current market landscape (CAIA Association, 2025).

Source: CFA Program Curriculum's Alternative Investments readings covering hedge funds, private equity, real assets, and due diligence frameworks (CFA Institute, 2025a)–together with Metrick & Yasuda's deep dive into private equity and venture capital (Metrick and Yasuda, 2010), CAIA Association's comprehensive CAIA-level materials on hedge funds, real estate, commodities, and other alternatives (CAIA Association, 2025), and Bodie, Kane & Marcus's chapters on alternative asset classes and portfolio integration in *Investments* (Bodie et al., 2017).

Corporate Issuers

```
***Corporate Issuer Analysis:***
```mermaid
graph TD
    A["Corporate Issuer Overview"] --> B["Industry Classification"]
    B --> C["Sector Trends and Competitive Landscape"]
    A --> D["Financial Statement Analysis"]
    D --> E["Profitability, Liquidity, Leverage"]
    A --> F["Credit Risk Assessment"]
    F --> G["Rating Agencies and Default Probabilities"]
    A --> H["Capital Structure and Issuance History"]
    H --> I["Bond Issuances and Debt Maturities"]
    A --> J["Corporate Governance and Management"]
    J --> K["Board Quality and Managerial Competence"]
    A --> L["Valuation and Investment Analysis"]
    L --> M["DCF, Relative Valuation, Multiples"]
```
```

Explanation: Step-1: Corporate Issuer Overview – Begin with a high-level understanding of the firm's business model, market positioning, and strategic objectives. This foundational context is essential for both equity and fixed income analysis (CFA Institute, 2025c).

Step-2: Industry Classification and Sector Trends – Classify the firm by sector or sub-sector (e.g., financials, consumer discretionary) and evaluate the competitive landscape. Analyze market trends, industry growth prospects, and systemic risks. This industry context shapes performance expectations and relative valuation (Penman, 2012).

Step-3: Financial Statement Analysis and Key Metrics – Analyze income statement, balance sheet, and cash flow data. Focus on metrics like revenue growth, operating margin, return on equity, and leverage. This step reveals the firm's financial health and operational efficiency (Penman, 2012; CFA Institute, 2025c).

Step-4: Credit Risk Assessment and Rating Measures – Evaluate creditworthiness through agency ratings (e.g., S&P, Moody's), credit spreads, and financial ratios. Analyze the probability of default and credit cycle indicators. This step is vital for bondholders and fixed income portfolio managers (Fabozzi, 2012).

Step-5: Capital Structure, Issuance History, and Debt Profile – Examine the firm's financing structure, including the mix of debt vs. equity, historical issuance patterns, and maturity schedules. This informs views on solvency and refinancing risks (Fabozzi, 2012).

Step-6: Corporate Governance and Leadership Quality – Assess governance practices such as board independence, shareholder rights, and disclosure quality. Evaluate the management team's execution track record and alignment with shareholder interests (CFA Institute, 2025c).

Step-7: Valuation and Investment Analysis – Use valuation models like DCF, P/E, or EV/EBITDA to derive intrinsic value. Develop an investment thesis based on fundamental insights. These valuation techniques are central to both equity and credit investing (Penman, 2012).

Source: CFA Program Curriculum's Equity Investments and Fixed Income readings–which cover firm analysis, industry evaluation, and credit assessment frameworks (CFA Institute, 2025c)–along with Penman's Financial Statement Analysis and Security Valuation for accounting-to-valuation linkages (Penman, 2012), and Fabozzi's Bond Markets, Analysis, and Strategies for credit risk and corporate debt issuance insights (Fabozzi, 2012).

B Prompt Template

B.1 Structured Chain-of-Thought (ST-CoT)

ST-CoT for CFA Exam

```
You are a CFA (chartered financial analyst) taking a test to evaluate your knowledge of finance. You think step-by-step approach
to answer queries.

Follow these steps:
1. Think through the problem step by step within the <thinking> tags.
2. Provide your final, concise answer within the <output> tags.

The <thinking> sections are for your internal reasoning process only.
Do not include any part of the final answer in these sections.
The actual response to the query must be entirely contained within the <output> tags.

### Response Format:
<thinking>
[Reasoning through options A, B, and C to understand and solve the problem.]
</thinking>
<output>
"answer": [Final your answer (A , B , or C )]
</output>
```

B.2 FinCoT

FinCoT for CFA Exam

```
You are taking a test for the Chartered Financial Analyst (CFA) program designed to evaluate your knowledge of different topics in
finance. You think step-by-step approach with reflection to answer queries.

Follow these steps:
1. Think through the problem step by step reflect and verify while reasoning within the <thinking> tags.
2. Please and put the answer your final, concise answer within the <output> tags.

The <thinking> sections are for your internal reasoning process only.
Do not include any part of the final answer in these sections.
The actual response to the query must be entirely contained within the <output> tags.

Hint:{THOUGHT.get("embedding_expert_blueprints_[i]")}

### Response Format:
<thinking>
[Think step by step and respond with your thinking and the correct answer (A, B, or C ), considering the specific sector.]
</thinking>

<output>
"sector": [The sector being addressed],
"question": [The financial question],
"answer": [Reflect and verify the final answer (A, B, or C)]
</output>
```

B.3 Classify Domain

Classify Domain CFA Exam

```
SYSTEM_INSTRUCTION = """You are a CFA expert. Categorize the given CFA question into exactly one
of these categories:

    Ethical and Professional Standards
    - Code of Ethics, Standards of Professional Conduct, professional integrity
    - Professional responsibilities, ethical decision-making, client interests
    Category code: Ethics

    Quantitative Methods
    - Statistical analysis, probability theory, hypothesis testing
    - Time value of money, financial mathematics, regression analysis
    Category code: Quant.Meth.

    Economic Analysis and Market Forces
    - Microeconomics: supply, demand, market structures
    - Macroeconomics: GDP, inflation, monetary policy, economic cycles
    Category code: Economics

    Financial Reporting and Analysis
    - Financial statements, accounting standards, ratio analysis
    - Balance sheets, income statements, cash flow analysis
    Category code: Fin.Reporting

    Corporate Finance and Issuers
    - Capital structure, dividend policy, corporate governance
    - Mergers & acquisitions, capital budgeting, risk management
    Category code: Corp.Issuers

    Equity Investments
    - Stock valuation, equity markets, company analysis
    - Market efficiency, equity portfolio management
    Category code: EquityInvest.

    Fixed Income Investments
    - Bond markets, yield curves, duration analysis
    - Credit analysis, fixed income portfolio management
    Category code: FixedIncome

    Derivative Instruments
    - Options, futures, forwards, swaps
    - Hedging strategies, derivative pricing, risk management
    Category code: Derivatives

    Alternative Investments
    - Real estate, private equity, hedge funds
    - Commodities, structured products, crypto assets
    Category code: Alter.Invest.

    Portfolio Management
    - Asset allocation, portfolio construction, rebalancing
    - Risk management, performance measurement, client objectives
    Category code: Port.Manage.

Respond with only the single most appropriate category code, nothing else. For example: Ethics,
Port.Manage., etc.
"""
```

C Domain Distribution

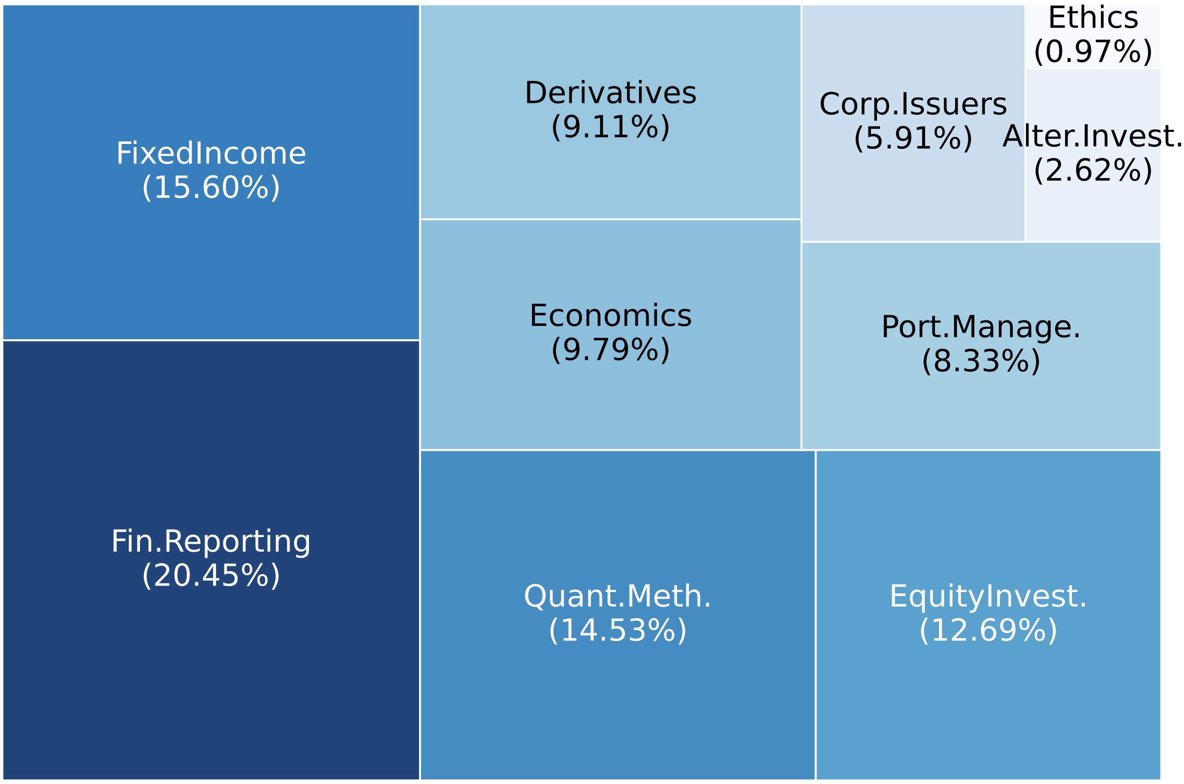


Figure 6: GPT-4o classified the benchmark domain distribution of CFA. A random sample of 100 items was manually audited by a financial expert to validate domain labels.

D Average Input and Output Tokens

D.1 Average Input Tokens

Prompt	Average Input Tokens (k)								
	Qwen2.5-7B	Qwen2.5-7B Instruct	Qwen3-8B Base	Qwen3-8B	Gemma-3-12B IT	Qwen3-8B (Thinker)	Fin-R1	DianJin-R1 7B	Fin-o1-8B
SP	**0.07***	**0.07***	**0.07***	**0.07***	**0.07***	**0.07***	**0.07***	**0.07***	**0.07***
UST-CoT	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09
ST-CoT	0.18	0.18	0.18	0.18	0.19	0.18	0.18	0.18	0.18
FinCoT (All Blueprints)	1.75	1.75	1.75	1.75	1.78	1.75	1.75	1.75	1.75
Domain-wise performance of FinCoT									
FinCoT (Economics)	0.55	0.55	0.55	0.55	0.56	0.55	0.55	0.55	0.55
FinCoT (FixedIncome)	0.34	0.34	0.34	0.34	0.36	0.34	0.34	0.34	0.34
FinCoT (Quant.Meth.)	0.33	0.33	0.33	0.33	0.34	0.33	0.33	0.33	0.33
FinCoT (EquityInvest.)	0.34	0.34	0.34	0.34	0.36	0.34	0.34	0.34	0.34
FinCoT (Port.Manage.)	0.33	0.33	0.33	0.33	0.35	0.33	0.33	0.33	0.33
FinCoT (Derivatives)	0.39	0.39	0.39	0.39	0.44	0.39	0.39	0.39	0.39
FinCoT (Fin. Reporting)	0.37	0.37	0.37	0.37	0.38	0.37	0.37	0.37	0.37
FinCoT (Alter.Invest.)	0.32	0.32	0.32	0.32	0.34	0.32	0.32	0.32	0.32
FinCoT (Corp.Issuers)	0.39	0.39	0.39	0.39	0.41	0.39	0.39	0.39	0.39

Table 2: Comparison of prompting techniques: average input token length (k) across models. **Bold** values highlight the prompt variant that uses the least tokens for each model.

D.2 Average Output Tokens

Prompt	Average Output Tokens (k)								
	Qwen2.5-7B	Qwen2.5-7B Instruct	Qwen3-8B Base	Qwen3-8B	Gemma-3-12B IT	Qwen3-8B (Thinker)	Fin-R1	DianJin-R1 7B	Fin-o1-8B
SP	0.45	**0.05***	0.89	0.32	**0.27***	1.52	0.88	2.18	**0.46***
UST-CoT	0.48	0.28	**0.31***	0.46	0.39	1.50	**0.58***	2.28	0.53
ST-CoT	**0.39***	0.22	3.42	**0.25***	0.31	1.35	2.22	7.20	0.58
FinCoT (All Blueprints)	2.22	0.29	0.38	0.36	0.32	**1.23***	1.92	**1.60***	0.79
Domain-wise performance of FinCoT									
FinCoT (Economics)	0.36	0.38	0.99	0.39	0.38	1.25	**2.01**	12.65	0.76
FinCoT (FixedIncome)	0.42	0.27	4.55	0.30	**0.32**	1.24	2.31	**8.31**	0.81
FinCoT (Quant.Meth.)	0.48	0.27	3.07	**0.31**	0.35	1.22	2.17	8.60	0.80
FinCoT (EquityInvest.)	**0.32**	0.31	7.18	0.37	0.34	1.19	2.16	10.07	0.78
FinCoT (Port.Manage.)	0.38	**0.26**	0.56	0.30	0.33	1.20	2.14	9.46	0.79
FinCoT (Derivatives)	0.36	0.30	**0.42**	0.39	0.34	1.24	2.05	5.54	0.81
FinCoT (Fin. Reporting)	0.46	0.28	0.93	0.33	0.34	1.19	2.13	8.76	**0.73**
FinCoT (Alter.Invest.)	0.47	**0.26**	0.50	0.38	0.34	1.23	2.16	11.53	0.77
FinCoT (Corp.Issuers)	0.52	**0.26**	1.18	0.32	0.33	**1.16**	2.08	11.37	0.82

Table 3: Comparison of prompting techniques: average output token length (k) across models. **Bold** values highlight the prompt variant that uses the least tokens for each model. (*) Indicates that the change in average output token count among the model-level prompt variants is statistically significant ($p < 0.05$) based on paired bootstrap testing; domain-specific rows are not tested for significance.

D.2.1 Efficiency of Input and Output Cost in Simulation

This appendix reports a cost–efficiency analysis under realistic output–input price ratios. Let I and O denote the average input and output tokens for a (prompt, model) pair. For a price ratio r,

$$\mathrm{Cost}(r) = I + r\,O, \qquad \mathrm{Efficiency}(r) = \frac{\mathrm{Cost}_{\mathrm{baseline}}(r)}{\mathrm{Cost}_{\mathrm{prompt}}(r)} = \frac{I_{\mathrm{base}} + r\,O_{\mathrm{base}}}{I_{\mathrm{prompt}} + r\,O_{\mathrm{prompt}}}.$$

Units and normalization. We measure cost in "input-token dollars": the effective input price is 1, and $r = \mathrm{price}_{\mathrm{out}}/\mathrm{price}_{\mathrm{in,eff}}$ carries the output premium. This rescaling makes Efficiency dimensionless and invariant to any common price factor.
Break-even and sensitivity. For a candidate prompt p vs. baseline b, the break-even ratio solving $\mathrm{Cost}_p(r) = \mathrm{Cost}_b(r)$ is

$$r^\star = \frac{I_p - I_b}{O_b - O_p} \qquad (O_p \neq O_b).$$

If $O_p < O_b$, then $\dfrac{d}{dr}\mathrm{Efficiency}(r) = \dfrac{O_b I_p - O_p I_b}{(I_p + rO_p)^2} > 0$: the candidate improves as r increases; if $O_p > O_b$, the trend reverses. When $O_p = O_b$, ranking depends only on inputs (I_b vs. I_p) and is independent of r.
Caching and effective input price. With prompt caching,

$$p_{\mathrm{in,eff}}(K) = p_{\mathrm{read}} + \frac{p_{\mathrm{write}}}{K}, \qquad r(K) = \frac{\mathrm{price}_{\mathrm{out}}}{p_{\mathrm{in,eff}}(K)},$$

where K is the number of reuses. Hence $r(K)$ increases monotonically in K and approaches $\mathrm{price}_{\mathrm{out}}/p_{\mathrm{read}}$ as $K \to \infty$.
Price instantiation (grid for plots). From public price points we use $r \in \{5, 6.9, 8, 14.29, 22.22, 40, 44.44, 50, 80\}$: *GPT-5*[5] input \$1.25/MTok (cached \$0.125/MTok), output \$10/MTok $\Rightarrow r{=}10/1.25{=}8$, $r_{\mathrm{cached}}{=}10/0.125{=}80$; *Claude Opus 4.1*[6] input \$15/MTok, output \$75/MTok; caching write \$18.75/MTok, read \$1.50/MTok $\Rightarrow r{=}5$ (no cache), 6.9 ($K{=}2$), 14.29 ($K{=}5$), 22.22 ($K{=}10$), 40 ($K{=}50$), 44.44 ($K{=}100$), and the read-only limit 50 ($K{\to}\infty$). We display r on a log scale because the grid spans an order of magnitude (5–80).
Worked example (illustrative). Baseline $(I_b, O_b) = (100, 300)$; candidate $(I_p, O_p) = (250, 150)$. At $r = 8$ (GPT-5 no cache): $\mathrm{Cost}_b = 100 + 8 \cdot 300 = 2500$, $\mathrm{Cost}_p = 250 + 8 \cdot 150 = 1450$, so Efficiency ≈ 1.72. Break-even $r^\star = (250 - 100)/(300 - 150) = 1$; the candidate dominates for $r > 1$.
Note (scope). All models evaluated in this appendix are *open-source*. The curves simulate dollar costs by pairing the measured (I, O) token counts from these models with *provider API prices* (GPT-5 and Claude Opus 4.1)—prices are used for *simulation only*; no paid API runs were executed for these experiments.

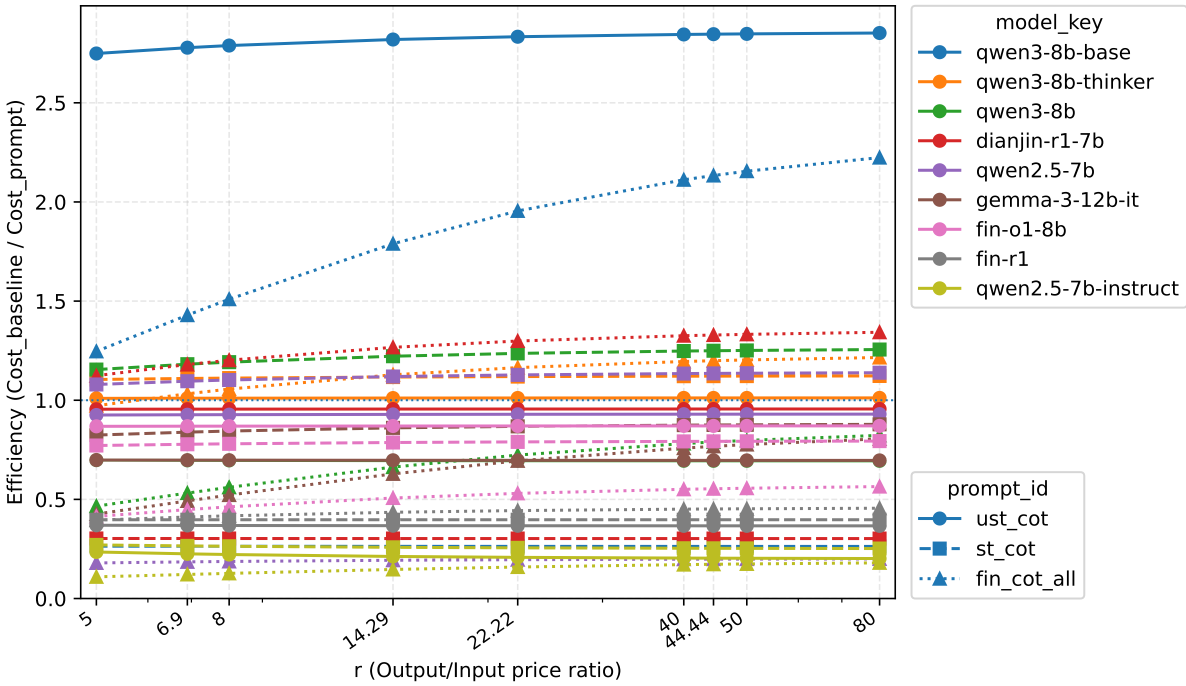


Figure 7: **Cost–efficiency vs. price ratio** r for UST-CoT, ST-CoT, and FinCoT-All across models. Efficiency is $\mathrm{Cost}_{\mathrm{baseline}}/\mathrm{Cost}_{\mathrm{prompt}}$ with $\mathrm{Cost} = I + rO$; values > 1 indicate lower cost than the baseline. r values use provider prices for GPT-5 and Claude Opus 4.1 as described above. *Notation:* MTok = million tokens; USD per MTok.

[5]OpenAI API pricing: https://openai.com/api/pricing/.
[6]Anthropic pricing: https://www.anthropic.com/pricing#api.

E Significance Testing

E.1 Accuracy

Model	Baseline	Comparison	Δ (pp)	95% CI (pp)	p-value	Significant
Qwen2.5-7B	SP	UST-CoT	-13.76	[-15.89, -11.63]	0.0000	✓
	SP	ST-CoT	-16.27	[-18.60, -14.05]	0.0000	✓
	SP	FinCoT (All)	-7.94	[-9.59, -6.30]	0.0000	✓
	UST-CoT	ST-CoT	-2.52	[-3.49, -1.65]	0.0000	✓
	UST-CoT	FinCoT (All)	5.81	[4.46, 7.27]	0.0000	✓
	ST-CoT	FinCoT (All)	8.33	[6.69, 9.98]	0.0000	✓
Qwen2.5-7B-Instruct	SP	UST-CoT	6.00	[4.65, 7.56]	0.0000	✓
	SP	ST-CoT	4.84	[3.59, 6.20]	0.0000	✓
	SP	FinCoT (All)	4.55	[3.29, 5.91]	0.0000	✓
	UST-CoT	ST-CoT	-1.16	[-1.84, -0.58]	0.0000	✓
	UST-CoT	FinCoT (All)	-1.45	[-2.23, -0.78]	0.0000	✓
	ST-CoT	FinCoT (All)	-0.29	[-0.68, 0.00]	0.1024	–
Qwen3-8B-Base	SP	UST-CoT	-9.40	[-11.24, -7.66]	0.0000	✓
	SP	ST-CoT	-15.31	[-17.54,-13.18]	0.0000	✓
	SP	FinCoT (All)	-17.35	[-19.77,-15.02]	0.0000	✓
	UST-CoT	ST-CoT	-5.91	[-7.36, -4.55]	0.0000	✓
	UST-CoT	FinCoT (All)	-7.96	[-9.59, -6.30]	0.0000	✓
	ST-CoT	FinCoT (All)	-2.04	[-3.00, -1.26]	0.0000	✓
Qwen3-8B	SP	UST-CoT	7.95	[6.30, 9.69]	0.0000	✓
	SP	ST-CoT	6.60	[5.14, 8.14]	0.0000	✓
	SP	FinCoT (All)	6.70	[5.23, 8.24]	0.0000	✓
	UST-CoT	ST-CoT	-1.35	[-2.13, -0.68]	0.0000	✓
	UST-CoT	FinCoT (All)	-1.26	[-1.94, -0.68]	0.0000	✓
	ST-CoT	FinCoT (All)	0.10	[0.00, 0.29]	0.7370	–
Qwen3-8B (Thinker)	SP	UST-CoT	-0.87	[-1.45, -0.39]	0.0002	✓
	SP	ST-CoT	0.00	[0.00, 0.00]	2.0000	–
	SP	FinCoT (All)	0.96	[0.39, 1.55]	0.0002	✓
	UST-CoT	ST-CoT	0.87	[0.39, 1.45]	0.0002	✓
	UST-CoT	FinCoT (All)	1.83	[1.07, 2.71]	0.0000	✓
	ST-CoT	FinCoT (All)	0.96	[0.39, 1.55]	0.0002	✓
Gemma-3-12B-IT	SP	UST-CoT	-24.999	[-27.71, -22.38]	0.0000	✓
	SP	ST-CoT	-23.934	[-26.55, -21.32]	0.0000	✓
	SP	FinCoT (All)	-22.765	[-25.39, -20.16]	0.0000	✓
	UST-CoT	ST-CoT	1.065	[0.48, 1.74]	0.0002	✓
	UST-CoT	FinCoT (All)	2.234	[1.36, 3.20]	0.0000	✓
	ST-CoT	FinCoT (All)	1.169	[0.58, 1.84]	0.0000	✓
Fin-R1	SP	UST-CoT	-9.49	[-11.34, -7.75]	0.0000	✓
	SP	ST-CoT	-8.62	[-10.37, -6.88]	0.0000	✓
	SP	FinCoT (All)	-10.07	[-11.92,-8.24]	0.0000	✓
	UST-CoT	ST-CoT	0.87	[0.39, 1.45]	0.0002	✓
	UST-CoT	FinCoT (All)	-0.58	[-1.07, -0.19]	0.0042	✓
	ST-CoT	FinCoT (All)	-1.45	[-2.23, -0.78]	0.0000	✓
Dianjin-R1-7B	SP	UST-CoT	10.662	[8.82, 12.60]	0.0000	✓
	SP	ST-CoT	9.594	[7.75, 11.43]	0.0000	✓
	SP	FinCoT (All)	-1.361	[-2.13, -0.68]	0.0000	✓
	UST-CoT	ST-CoT	-1.068	[-1.74, -0.48]	0.0000	✓
	UST-CoT	FinCoT (All)	-12.023	[-14.05, -10.08]	0.0000	✓
	ST-CoT	FinCoT (All)	-10.956	[-12.89, -9.01]	0.0000	✓
Fino1-8B	SP	UST-CoT	0.294	[0.00, 0.68]	0.0984	–
	SP	ST-CoT	1.264	[0.58, 1.94]	0.0000	✓
	SP	FinCoT (All)	2.429	[1.55, 3.39]	0.0000	✓
	UST-CoT	ST-CoT	0.970	[0.39, 1.55]	0.0000	✓
	UST-CoT	FinCoT (All)	2.134	[1.26, 3.00]	0.0000	✓
	ST-CoT	FinCoT (All)	1.165	[0.58, 1.84]	0.0000	✓

Table 4: Paired bootstrap significance testing ($B = 10{,}000$ samples) for accuracy differences across prompt strategies. Δ indicates average accuracy difference (in percentage points), with 95% confidence intervals (CI) and p-values. A result is considered statistically significant if $p < 0.05$.

E.2 Average Output Tokens

Model	**Baseline**	**Comparison**	Δ (k)	95% CI (k)	p-value	Significant
Qwen2.5-7B	SP	UST-CoT	-0.09867	[-0.09867, -0.09867]	0.0000	✓
	SP	ST-CoT	0.02273	[0.02273, 0.02273]	0.0000	✓
	SP	FinCoT (All)	-0.11555	[-0.11555, -0.11555]	0.0000	✓
	UST-CoT	ST-CoT	0.12140	[0.12140, 0.12140]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.01688	[-0.01688, -0.01688]	0.0000	✓
	ST-CoT	FinCoT (All)	-0.13828	[-0.13828, -0.13828]	0.0000	✓
Qwen2.5-7B-Instruct	SP	UST-CoT	-0.227	[-0.227, -0.227]	0.0000	✓
	SP	ST-CoT	-0.169	[-0.169, -0.169]	0.0000	✓
	SP	FinCoT (All)	-0.241	[-0.241, -0.241]	0.0000	✓
	UST-CoT	ST-CoT	0.058	[0.058, 0.058]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.014	[-0.014, -0.014]	0.0000	✓
	ST-CoT	FinCoT (All)	-0.072	[-0.072, -0.072]	0.0000	✓
Qwen3-8B-Base	SP	UST-CoT	-0.584	[-0.584, -0.584]	0.0000	✓
	SP	ST-CoT	2.522	[2.522, 2.522]	0.0000	✓
	SP	FinCoT (All)	-0.516	[-0.516, -0.516]	0.0000	✓
	UST-CoT	ST-CoT	3.106	[3.106, 3.106]	0.0000	✓
	UST-CoT	FinCoT (All)	0.068	[0.068, 0.068]	0.0000	✓
	ST-CoT	FinCoT (All)	-3.038	[-3.038, -3.038]	0.0000	✓
Qwen3-8B	SP	UST-CoT	0.141	[0.141, 0.141]	0.0000	✓
	SP	ST-CoT	-0.067	[-0.067, -0.067]	0.0000	✓
	SP	FinCoT (All)	0.048	[0.048, 0.048]	0.0000	✓
	UST-CoT	ST-CoT	-0.208	[-0.208, -0.208]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.093	[-0.093, -0.093]	0.0000	✓
	ST-CoT	FinCoT (All)	0.115	[0.115, 0.115]	0.0000	✓
Qwen3-8B (Thinker)	SP	UST-CoT	-0.018	[-0.018, -0.018]	0.0000	✓
	SP	ST-CoT	-1.271	[-1.271, -1.271]	0.0000	✓
	SP	FinCoT (All)	-0.168	[-0.168, -0.168]	0.0000	✓
	UST-CoT	ST-CoT	-1.253	[-1.253, -1.253]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.150	[-0.150, -0.150]	0.0000	✓
	ST-CoT	FinCoT (All)	1.103	[1.103, 1.103]	0.0000	✓
Gemma-3-12B-IT	SP	UST-CoT	0.11985	[0.11985, 0.11985]	0.0000	✓
	SP	ST-CoT	0.03661	[0.03661, 0.03661]	0.0000	✓
	SP	FinCoT (All)	0.04523	[0.04523, 0.04523]	0.0000	✓
	UST-CoT	ST-CoT	-0.08324	[-0.08324, -0.08324]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.07462	[-0.07462, -0.07462]	0.0000	✓
	ST-CoT	FinCoT (All)	0.00862	[0.00862, 0.00862]	0.0000	✓
Fin-R1	SP	UST-CoT	1.526	[1.526, 1.526]	0.0000	✓
	SP	ST-CoT	1.338	[1.338, 1.338]	0.0000	✓
	SP	FinCoT (All)	1.035	[1.035, 1.035]	0.0000	✓
	UST-CoT	ST-CoT	-0.188	[-0.188, -0.188]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.491	[-0.491, -0.491]	0.0000	✓
	ST-CoT	FinCoT (All)	-0.303	[-0.303, -0.303]	0.0000	✓
Dianjin-R1-7B	SP	UST-CoT	0.10023	[0.10023, 0.10023]	0.0000	✓
	SP	ST-CoT	5.02159	[5.02159, 5.02159]	0.0000	✓
	SP	FinCoT (All)	-0.57669	[-0.57669, -0.57669]	0.0000	✓
	UST-CoT	ST-CoT	4.92136	[4.92136, 4.92136]	0.0000	✓
	UST-CoT	FinCoT (All)	-0.67692	[-0.67692, -0.67692]	0.0000	✓
	ST-CoT	FinCoT (All)	-5.59828	[-5.59828, -5.59828]	0.0000	✓
Fino1-8B	SP	UST-CoT	0.06788	[0.06788, 0.06788]	0.0000	✓
	SP	ST-CoT	0.11765	[0.11765, 0.11765]	0.0000	✓
	SP	FinCoT (All)	0.33273	[0.33273, 0.33273]	0.0000	✓
	UST-CoT	ST-CoT	0.04977	[0.04977, 0.04977]	0.0000	✓
	UST-CoT	FinCoT (All)	0.26485	[0.26485, 0.26485]	0.0000	✓
	ST-CoT	FinCoT (All)	0.21508	[0.21508, 0.21508]	0.0000	✓

Table 5: Paired bootstrap significance testing ($B = 10{,}000$ samples) for average output token differences across prompt strategies. Δ indicates mean difference in output length (in thousands of tokens), with 95% confidence intervals and p-values. A result is significant if $p < 0.05$.

F Radar Behavior Accuracy

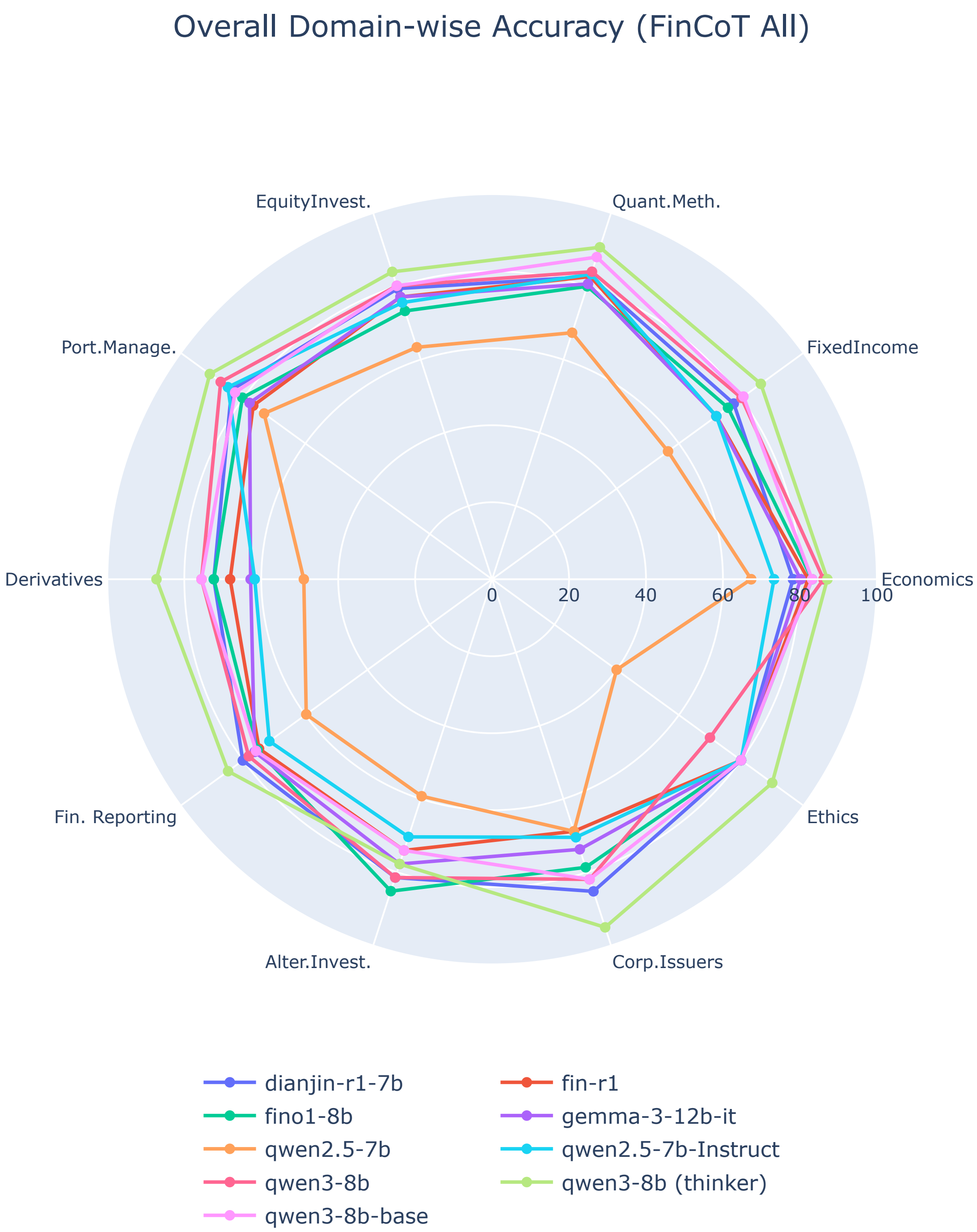
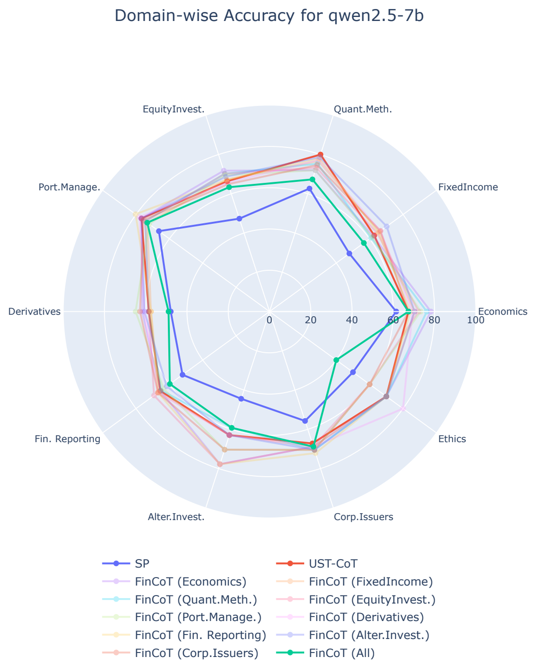
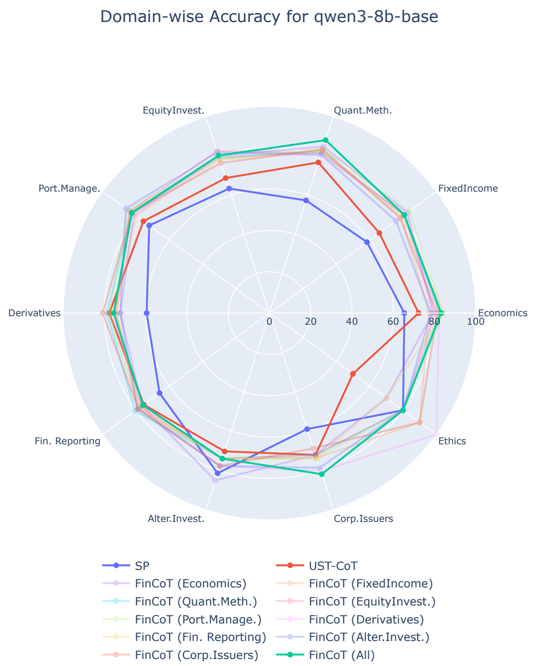


Figure 8: Overall FinCoT behaviour accuracy.

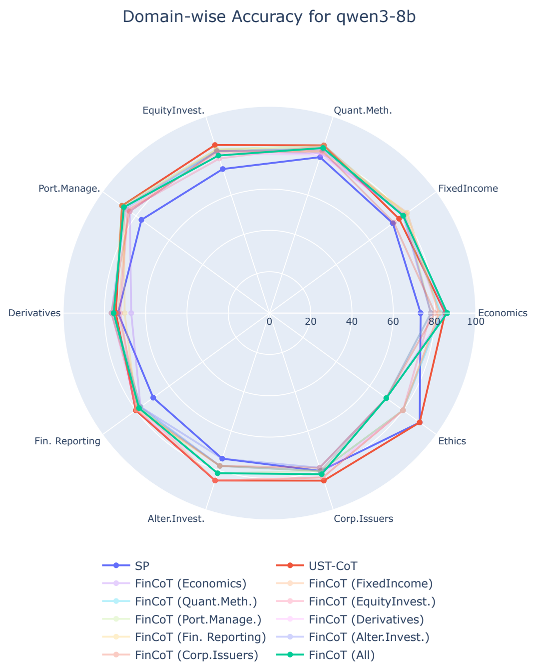


(a) Qwen2.5-7B

(b) Qwen2.5-7B-Instruct

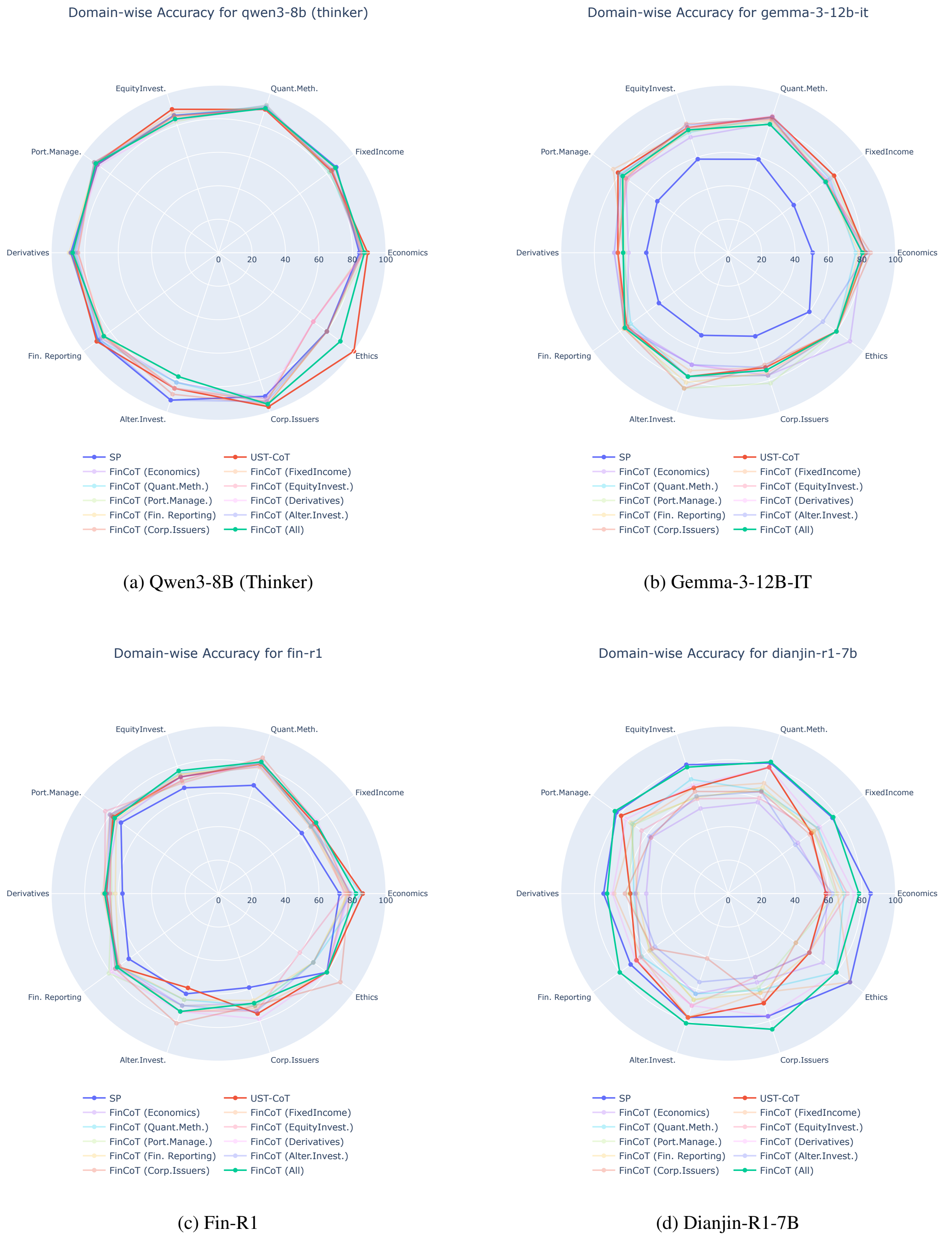


(c) Qwen3-8B-Base



(d) Qwen3-8B

Figure 9: Radar charts for each model variant.



(a) Qwen3-8B (Thinker)

(b) Gemma-3-12B-IT

(c) Fin-R1

(d) Dianjin-R1-7B

Figure 10: Radar charts for each model variant (charts 5–8).

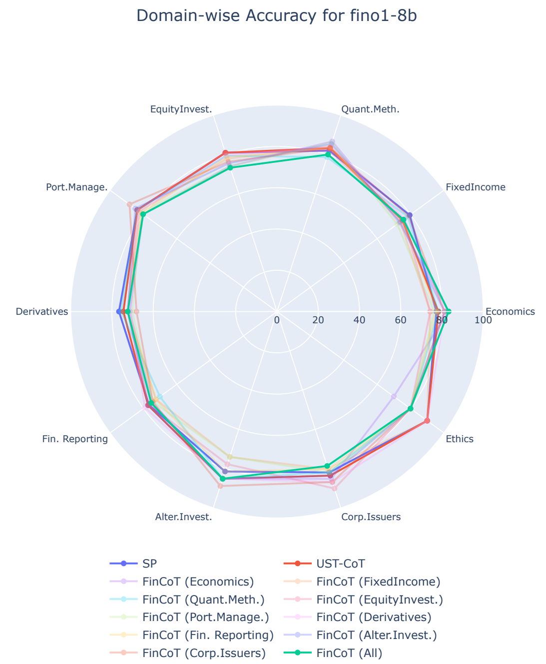


Figure 11: Radar charts for each model variant (charts 9).

Assessing RAG System Capabilities on Financial Documents

Oscar Lithgow-Serrano[1], David Kletz[1], Vani Kanjirangat[1], David Adametz[2], Marzio Lunghi[2], Claudio Bonesana[1], Matilde Tristany Farinha[2], Yuntao Li[2], Detlef Repplinger [2], Marco Pierbattista[2], Stefania Stan[2], Oleg Szehr[1]

[1]SUPSI, IDSIA, Switzerland

[2]UBS Switzerland AG and its affiliates

Correspondence: {david.kletz, oleg.szehr}@supsi.ch, {david.adametz, stefania.stan}@ubs.com

Abstract

Financial institutions are increasingly using Retrieval-Augmented Generation (RAG) systems for document processing. However, there is still limited systematic evaluation focused on industry-specific content. In this study, we evaluated four state-of-the-art RAG architectures for processing of financial documents using FinDoc-RAG, a benchmark we developed for this purpose. This benchmark consists of over 600 question-answer pairs derived from 46 documents from a banking institution. Source materials include product descriptions, investment guides, legal policies, and marketing brochures, all of which contain dense numerical content and complex layouts. Our evaluation shows significant performance gaps: while leading systems achieve an accuracy of 0.91 on factual extraction, performance drops to 0.44 on cross-document synthesis tasks. Our experiments demonstrate varying strengths of the explored RAG approaches across different question complexities in the financial services sector and position *FinDoc-RAG* as a benchmark for measuring progress in this area.

1 Introduction

Financial institutions process thousands of documents that require human interpretation for client advisory tasks, regulatory compliance, and product inquiries. Large Language Models (LLMs) offer automation potential but face deployment challenges such as regulatory constraints that prevent external data transfer and operational complexity challenges, such as documents that combine textual content with numerical data, complex layouts, and requirements. Current LLMs show limitations in quantitative reasoning and cross-document synthesis essential for financial applications. Retrieval-Augmented Generation (RAG) addresses privacy constraints while leveraging LLM capabilities, but systematic evaluation on financial documents remains limited.

Financial documents present unique challenges: they require factual extraction, quantitative reasoning with numerical data, and information synthesis across multiple documents for comprehensive responses. Evaluating RAG performance on these distinct task types requires specialized benchmarks that reflect real-world financial complexity.

Existing Question-Answer (QA) datasets focus on Wikipedia articles (Yang et al., 2018, Kwiatkowski et al., 2019), academic papers (Pramanick et al., 2024), or single-domain sources (Pipitone and Alami, 2024, Ngo et al., 2024), and therefore fail to capture the heterogeneous nature of financial document collections. While financial QA benchmarks often focus on narrow regulatory domains, they neglect the broader spectrum of client-facing content. Moreover, no benchmark systematically evaluates the intersection of financial materials and the diverse types of tasks critical for the deployment of RAG.

We introduce *FinDoc-RAG*, a QA benchmark comprising 600+ QA pairs from 46 documents in English from UBS AG and other UBS entities. Documents include product descriptions, investment guides, legal policies, and marketing materials with dense numerical content and regulatory references. The questions span nine complexity levels (L0-L8) that target factual extraction, quantitative reasoning, and multi-document synthesis. Evaluation of five RAG architectures –vector-based indexing, graph-enhanced RAG, hierarchical summary-style retrieval (e.g., Raptor 4.2), and Knowledge Graph (KG)– reveals systematic performance gaps: leading systems achieve 0.91 accuracy in factual extraction but only 0.44 on multi-document synthesis tasks.

Contributions:

1. We present FinDoc-RAG, a RAG-focused question-answer benchmark over heterogeneous financial documents. It comprises

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

nine task levels, each associated with a predefined difficulty ranging from single-document extraction to multi-document synthesis. The data are published at `https://gitlab-core.supsi.ch/dti-idsia/ai-finance-papers/findoc-rag`.

2. We evaluate four representative RAG architectures, demonstrating their individual strengths and weaknesses. The dataset is released to foster research and compare RAG systems in the financial domain.

Our analysis identifies specific failure modes in current RAG approaches, with quantitative reasoning showing high performance variability and multi-document synthesis proving most challenging across all systems. The benchmark enables systematic evaluation of financial RAG systems and provides deployment readiness assessment for different task types.

2 Related Work

By retrieving relevant passages from external document collections prior to generation, RAG systems improve factual grounding, enhance domain-specific accuracy, and support local deployment with preserved data privacy. This architecture is especially promising in specialized domains like finance, where even state-of-the-art LLMs struggle, when used in isolation, with quantitative reasoning, factual consistency, and multi-document synthesis (Rasool et al., 2024).

The creation of information-seeking QA datasets has been pivotal in driving progress in RAG-based approaches.

2.1 Domain-Specific and Heterogeneous QA Benchmarks

General-purpose benchmarks such as Natural Questions (Kwiatkowski et al., 2019) evaluate QA over real-world queries and Wikipedia passages. Domain-specific datasets target deeper comprehension in specialized settings. For instance, Qasper (Dasigi et al., 2021) covers academic articles in NLP, SPIQA (Pramanick et al., 2024) addresses reasoning over complex figures and tables, and datasets such as MedRGB (Ngo et al., 2024) and LegalBench-RAG (Pipitone and Alami, 2024) focus on medical and legal domains, respectively.

Recent efforts have extended QA evaluation to longer and more complex contexts. HOTPOTQA (Yang et al., 2018) and MultiHop-RAG (Tang and Yang, 2024) test multihop reasoning, while QuALITY (Pang et al., 2022) and MMLongBench-Doc (Ma et al., 2024) challenge models with long documents and structured layouts. Multimodal benchmarks such as VisDoMBench (Suri et al., 2025), MRAG-Bench (Hu et al., 2025), and MuRAG (Chen et al., 2022) further evaluate the integration of textual and visual information.

However, most existing datasets assume homogeneous, well-structured sources and do not reflect the heterogeneity of real-world document collections. In industry settings, especially in finance, documents range from reports and contracts to internal memos, with various formats, styles, and terminology. Financial QA datasets such as FinTextQA (Chen et al., 2024a), FinDER (Choi et al., 2025), and GBS-QA (Sohn et al., 2021) typically focus on narrow domains or single-source documents, limiting their generalizability.

To bridge this gap, we introduce *FinDoc-RAG*, a benchmark designed for QA over heterogeneous financial documents. It captures cross-document reasoning, contextual variability, and structural diversity characteristic of real-world financial information ecosystems.

2.2 Evaluation Strategies for QA Benchmarks

Evaluating QA benchmarks—particularly those involving long, heterogeneous, or domain-specific documents—remains a major challenge. Broadly, evaluation strategies fall into two categories: model-centric, which assess the performance of different LLMs, and method-centric, which compare paradigms such as extractive, abstractive, or RAG.

While early QA benchmarks focused primarily on comparing model performance, recent efforts have shifted toward approach-specific evaluations, particularly in the context of RAG. Despite the strong general QA capabilities of state-of-the-art LLMs such as GPT-4, studies show persistent limitations in multistep reasoning and numerical understanding (Rasool et al., 2024). In contrast, RAG-based methods demonstrate improved factual grounding and reduced hallucination in domain-specific tasks (Chen et al., 2024b).

However, recent findings indicate that no single approach consistently outperforms others across all task types. The LaRA benchmark (Li et al., 2025), for example, demonstrates that both RAG and long-context methods succeed in different scenarios, highlighting the need for nuanced, task-aware evaluation frameworks that account for document complexity, question type, and reasoning depth.

These insights emphasize the importance of benchmarks that capture real-world document heterogeneity while enabling multifaceted evaluations aligned with the strengths and trade-offs of both models and methodologies.

3 *FinDoc-RAG* Benchmark

FinDoc-RAG comprises 600+ QA pairs extracted from 46 documents in English from UBS AG and other UBS entities.

Documents span four categories: product descriptions, investment guides, legal policies, and marketing materials. The collection includes two distinct subsets: V1 contains concise factsheets with dense numerical content averaging 2,400 words, while V2 features comprehensive reports with complex layouts averaging 12,000 words and rich structural elements including tables, footnotes, and cross-references.

3.1 Question Generation Methodology

Questions are structured across nine complexity levels (L0-L8) targeting three task types: factual extraction, information integration, and multi-document synthesis. Each level introduces specific constraints on document scope, quote requirements, and reasoning complexity based on our initial design expectations (see Table 1). However, empirical results reveal that expected difficulty progression does not always align with actual model performance.

Question generation is carried out using two methodological approaches: raw document content and clustered document summaries. Raw document approaches (L0-L4) generate questions directly from the original text, enabling extraction and role-based query formulations. For the other levels, the cluster-based approach first creates document summaries, embed them semantically, and clusters related content using Gaussian Mixture Models. The optimal number of clusters is selected using the Bayesian Information Criterion. The resulting clusters serve as the basis for generating questions that integrate information across related contexts, yielding multi-aspect queries that test narrative understanding rather than isolated fact retrieval.

3.1.1 Factual Extraction Tasks

Levels L0, L1, L4, and L5 involve single-document retrieval tasks that require direct text extraction, without the need for computational reasoning.

L0 Generic prompts applied uniformly to V1 documents (concise factsheets) generate self-contained QA pairs. Questions avoid generic formulations and reference relevant topics when needed. Each generated pair undergoes manual review to ensure groundedness and eliminate hallucinations.

L1 Targets V2 documents (comprehensive reports) using stratified processes that identify specific textual quotes including numerical values, dates, and key definitions. The questions remain strictly answerable from isolated textual details.

L4 Role-based prompting simulates heterogeneous user perspectives through three financial personas: young student exploring digital financial tools, elderly widow prioritizing stability with limited resources, and high-earning digital nomad navigating minimal traditional banking reliance. Each persona generates questions reflecting typical concerns and levels of knowledge.

L5 This method relies on thematic document selection, using the cluster-based approach to identify coherent topic groups. By drawing questions from individual summaries within these clusters, the LLM is guided to generate queries tied to the cluster's topic. This approach provides controlled complexity by focusing on specific topic areas while maintaining single-document question scope.

3.1.2 Information Integration Tasks

Levels L2 and L6 require combining information from multiple document sections or sentence fragments, which may require mathematical operations, numerical comparisons, or logical synthesis of related concepts.

L2 Aggregates information from different sections within a single document, connecting thematically related passages to generate abstract queries requiring numerical synthesis across document parts.

L6 Synthesizes 2–3 direct quotes from distinct sentence fragments, sometimes involving basic mathematical operations, percentage calculations, or numerical comparisons that are strictly entailed by the quoted content.

3.1.3 Cross-Document Synthesis Tasks

The integration of cross-document information that requires reasoning across multiple sources is the core of levels L3, L7, and L8.

L3 Extracts key concepts and topics from multiple documents, combining information into QA pairs that require understanding relationships between different source documents.

L7 Complex multi-document synthesis requiring >=3 quotes from >=3 different documents, testing the ability to integrate information across diverse source materials.

L8 Cluster-based maximum complexity synthesis where N quotes from N documents equal the cluster size. Uses the same clustering methodology as L5 but operates across document boundaries, requiring synthesis within clusters containing multiple document summaries. Similar to L5, the aspects from different summaries enable the generation of more complex and diverse QA pairs, while in L8, the complexity further increases, attributed to its multi-document summarization strategy.

Detailed questions generation settings and parameters can be found in the Appendix B.

3.2 Benchmark Statistics

Document Distribution The dataset consists of 46 documents spanning four radically different templates: product factsheets, legal documents, investment guides, and marketing materials. This heterogeneity ensures a diverse and representative dataset, yielding over 600 high-quality QA pairs. *Legal* documents have the highest average word count (4,000 words), with significant outliers exceeding 10,000 words, while *Research and Reports* are the shortest, averaging ∼500 words. Further details are provided in Appendix C.2, Figure 5.

Lexical Analysis Using Type-Token Ratio (TTR) as a measure of lexical diversity we found that *Research and Reports* documents have more diverse vocabulary (∼ 0.09 TTR) compared to other types of documents (Figure 7). The density of financial terminology peaks in *Product Information* documents (6-7% of tokens) and *Forms and Guides* (4-6%), remaining lowest in *Marketing Materials* (1-2%). The complete analysis is available in Appendix C.2, Figure 11.

Layout Complexity *Legal* documents show the highest structural complexity (scores 10-25+, including outliers at 120), featuring tables, sections, and complex nesting levels. *Product Information* exhibit minimal complexity (scores ∼ 5). Details can be found in Appendix C.2, Figure 9.

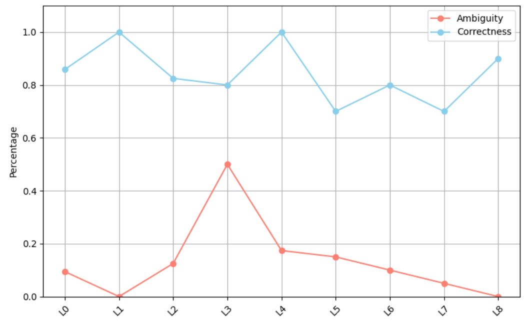


Figure 1: Ambiguity and correctness across levels with human validation.

3.3 Quality Validation

Human validation at all levels reveals correctness rates above 80% for most levels, with L3 and L7 achieving 95% correctness. Ambiguity ranges from 0% (L1, L8) to 50% (L3), with most levels maintaining low-to-moderate ambiguity (see Figure 1 and Appendix C.2).

The validation process assessed maximum 50 randomly sampled questions per level using two annotators with financial domain expertise [1]. Additionally, QA pairs were deduplicated and reviewed by a domain expert to remove those that appeared incorrect based on domain expertise, without formal fact-checking. For all levels, the financial experts performed internal validation of the QA pairs to filter out the suprious pairs.

4 Evaluating Financial RAG Systems

Our objective is to identify which RAG approaches achieve sufficient accuracy across various financial document processing tasks and to uncover fundamental limitations that point to necessary architectural improvements. We evaluated four representative RAG systems, each corresponding to a distinct architectural approach.

4.1 Task Setup

We adopt a standard RAG setup, where each system receives a query and retrieves the text spans from a fixed corpus that the system considers relevant to generate an answer to the query. In our

[1] The publicly released dataset may contain fewer QA pairs per level than manually validated. Additional pairs were validated but cannot be shared due to confidentiality constraints.

Level	# Quotes	# Docs	Expected Difficulty	Special Features
L0	1	1	Easy	V1 document set
L1	1	1	Easy	V2 document set
L2	Multiple	1	Medium	V2 docs, multiple supporting quotes
L3	Multiple	Multiple	Hard	Cross-document information synthesis
L4	1	1	Medium	Non-expert phrasing style
L5	≥ 1	1	Easy	Based on cluster summaries
L6	2-3	≥ 1	Medium	Quotes from different sentence fragments
L7	≥ 3	≥ 3	Hard	Quotes from different summaries
L8	$\geq N$	N	Very Hard	N equals cluster size

Table 1: Question Complexity Levels Description.

case, the corpus consists of the 46 financial documents included in the *FinDoc-RAG* benchmark. The set of questions comprises questions spanning the nine complexity levels (L0–L8) defined in the benchmark.

Each architecture under evaluation (detailed in 4.2) processes the full set of questions in a zero-shot setting. The retriever has access to all 46 documents and selects the subset of documents that it considers relevant to answer the input query. The retrieved documents, along with the query, are then passed to the generator component, which produces the answer. All systems utilize default configurations without fine-tuning, hyperparameter optimization, or preprocessing customization (e.g., chunking, enrichment, propositionalization) to provide baseline performance assessment representative of out-of-the-box deployment scenarios.

The generated answers are evaluated against the expected answers provided in *FinDoc-RAG*. Multiple evaluation metrics are used to assess different aspects of system performance (see Section 4.3). Our assessment evaluates end-to-end RAG system performance, rather than isolated retrieval or generator components. The systems handle document selection from the collection and passage identification within selected documents as integrated processes, with the final answer generation completing the pipeline. This holistic evaluation reflects real-world deployment scenarios in which RAG systems must complete a document-to-answer workflow without human intervention in retrieval decisions.

4.2 Selected RAG Architectures

Vector-based (Vector-RAG) This baseline method is based on an index of dense vector representations. It starts by encoding documents into high-dimensional embeddings using a neural encoder model. These document vectors are stored in a vector index that supports efficient similarity search. When a query is submitted, it is encoded using the same model, and the system retrieves documents with embeddings most similar to the query vector, in this case using cosine as the similarity metric. During retrieval, the system ranks documents based on their vector similarity scores to determine relevance. By leveraging the semantic representativeness of dynamic embeddings produced by neural encoders, this approach can identify topically relevant information even when exact keyword matches are absent. This method is computationally efficient for large-scale retrieval.

Recursive Abstractive Processing for Tree-Organized Retrieval (RAPTOR) Sarthi et al. (2024) is a semi-structured method based on hierarchical summarization organized on a tree structure. It starts by breaking down large texts into smaller chunks, which are then embedded using a BERT-based encoder. These chunks are grouped into clusters using a Gaussian Mixture Model, and a language model summarizes each cluster. This process is repeated to build a tree with multiple levels of summaries. During retrieval, RAPTOR can either traverse the tree layer-by-layer or evaluate nodes across all layers to find the most relevant information. By capturing high- and low-level details about a text, this approach helps with handling a wide range of questions and improves the integration and relevance of the retrieved information.

Graph RAG (GraphRAG) Edge et al. (2025) is a multistep method to answer questions from large text collections. First, it creates a graph-based index by building an entity knowledge graph and generating summaries for groups of related entities.

When a question is asked, these summaries help create partial answers, which are then combined into a final response. Specifically, it uses algorithms such as the Leiden algorithm (Traag et al., 2019) to detect communities within the graph by identifying groups of closely related elements, including nodes, edges, and covariates. By partitioning the graph into these communities, the method can perform parallel summarization and employs a hierarchical structure to provide different levels of detail. It also uses a map-reduce technique to combine the partial answers from the parallel summaries. This approach is designed to handle broad questions and large amounts of text.

Graph Foundation Model for Retrieval Augmented Generation (GFM-RAG) Luo et al. (2025) is a query-aware Graph Neural Network (GNN) pretrained on more than 60 knowledge graphs with over 14M triples and 700k documents. This foundation model is intended to generalize to similar Knowledge Graphs (KGs) independently of the domain. Following Luo et al. (2025), we created our KG by prompting an LLM over the source documents to generate the KG triplets. The pretrained query-aware GFM retriever model is then used to extract relevant entities from the KG with respect to the given query. Based on the relevance scores of the entities, the top entities are selected and then used by a document ranker that retrieves the ranked set of relevant documents. The final top K documents are given as the context for the LLM along with the query to generate the respective answer.

Detailed experimental settings and parameters can be found in Appendix B.

4.3 Evaluation Metrics and Scoring

To robustly assess the quality of the response provided by an architecture, we try to capture distinct dimensions of correctness beyond the exact text overlap by using different metrics.

The most direct evaluation of QA performance is the degree of overlap between the generated answer and the expected reference answer. This surface-level correspondence is captured by traditional text-matching metrics. Following the SQuAD evaluation protocol (Rajpurkar et al., 2016), we report two standard metrics: Exact Match (EM), which assigns a binary score based on exact string equivalence after normalization, and F1 Score, which computes word-level overlap to capture the trade-off between precision and recall.

Alternatively, for a fairer and more robust evaluation, it is key to recognize answers that are semantically equivalent to the reference, even when they differ in phrasing (Bulian et al.; Li et al.; Thakur et al.; Reiter). To capture this aspect, we introduce a second evaluation based on semantic similarity. We use BERTScore (Zhang* et al., 2020), which measures the alignment between predicted and reference answers by computing token-level similarities using contextualized BERT embeddings. The metric performs greedy matching between the tokens in both texts, aligning each token with its most similar counterpart based on cosine similarity. From these alignments, it computes precision, recall, and F1 scores that reflect the degree of semantic correspondence between the two answers, even when their surface forms differ.

We also include an LLM-based metric, which is more sensitive to semantic meaning that depends on subtle contextual cues. This approach, inspired by Zheng et al. (2023); Friel et al. (2024), uses an LLM to evaluate the semantic equivalence between predicted and reference answers within the context of the question. The LLM is prompted to determine whether the candidate's answer accurately preserves the meaning of the ground truth. To improve reliability and better reflect the model's confidence, we frame the evaluation as a factual correctness task with a binary classification—labeling answers as either *CORRECT* or *INCORRECT*. The LLM provides a brief explanation for its judgment while applying semantic flexibility for minor phrasing differences that preserve core meaning, tolerating reasonable omissions that do not introduce ambiguity, and ignoring stylistic differences unless they impact clarity. The answers are marked *INCORRECT* if they contain factual errors, false claims, significant omissions, or distortions of the core meaning. Each question was evaluated by the LLM judge across 3 independent runs. We computed accuracy as the mean proportion of *CORRECT* responses per question (ranging from 0 to 1), then averaged across all questions per system.

To capture a more fine-grained measure of answer quality, we finally adopt the *LLMLogScore* (L3Score) metric introduced by Pramanick et al. (2024). This approach leverages the log-likelihood probabilities generated by an LLM when prompted to evaluate semantic similarity between a candidate answer and the ground truth. By comparing the model's predicted probabilities of "yes" and "no" responses, L3Score computes a continuous similarity score normalized between 0 and 1, enabling a more sensitive and graded evaluation without relying on arbitrary predefined scales.

Detailed evaluation settings and parameters can be found in Appendix B.

5 Results & Analysis

The observed performance patterns should be interpreted as indicators of the benchmark's inherent challenges across question complexity levels rather than definitive assessments of the approaches' capabilities.

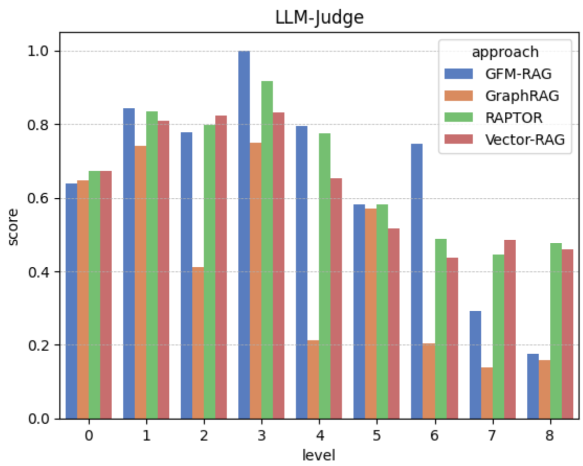


Figure 2: RAG approaches across question levels measured by LLM-Judge.

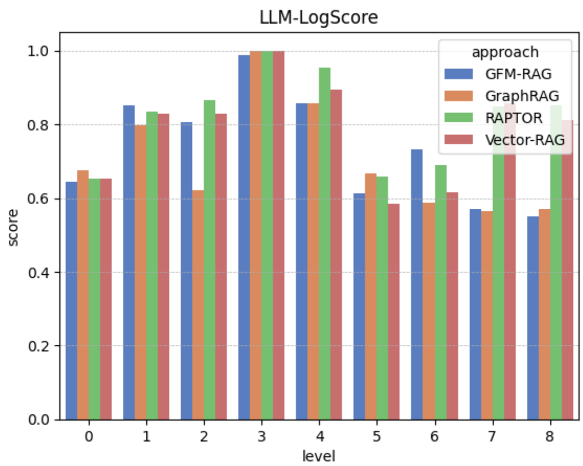


Figure 3: RAG approaches across question levels measured by LLMLogScore.

5.1 Benchmark complexity analysis

Task type analysis reveals distinct performance patterns across RAG architectures. Factual extraction tasks (L0, L1, L4, L5) have relatively stable performance with accuracy ranging from 0.51 to 0.84 0.51-0.84 across systems –except for a significant drop by GraphRAG on L4 (see details in subsection 5.2)– demonstrating suitability for production deployment in basic extraction tasks. Information integration tasks (L2, L6) show substantial performance variability (0.20-0.82 range) with GFM-RAG excelling at L6 (0.75) and GraphRAG struggling in both levels, suggesting system-dependent capabilities for intra-document synthesis. Cross-Document Synthesis tasks (L3, L7, L8) demonstrate extreme performance ranges (0.14-1.00), highlighting fundamental architectural differences in multi-document reasoning capabilities. In the individual-level difficulty analysis, L1 is the easiest level with an average score (across systems) of 0.81 and, counterintuitively, L3 achieved the highest individual system performance (GFM-RAG: 1.00). Levels L6-L8 show substantial performance degradation across most systems, L8 is the one with the lowest average score (0.32) but, L4 presents the highest variance across systems.

5.2 System performance analysis

Vector-RAG exhibits competitive baseline performance, particularly excelling at L2 and L3 (0.82-0.83), but its performance drops on synthesis tasks. RAPTOR, the other top performer, shows consistent mid-to-high performance across all complexity levels, achieving the second-highest scores on L3 (0.92) and maintaining stability across diverse question types. With the highest average score across levels (0.66) with low standard deviation ($\sigma = 0.17$), RAPTOR is a robust general purpose approach. GraphRAG demonstrates mixed performance patterns: strong capability on basic tasks (L1: 0.742) but severe degradation on complex synthesis (L7-L8: 0.14-0.16), which suggests architectural limitations in multi-document reasoning. GFM-RAG is one of the strongest performers, achieving perfect accuracy on L3 (1.00) and leading performance on L6 (0.74), demonstrating good capabilities for multi-document reasoning tasks.

System Selection Guidance For basic document extraction, all systems except GraphRAG achieve adequate performance ($\geq$ 0.65), indicating reliable potential for factual retrieval tasks. Complex synthesis scenarios require targeted system selection: GFM-RAG for multi-document reasoning (L3: 1.00, L6: 0.74), RAPTOR for consistent cross-level performance, and Vector-RAG for mid-complexity applications (L2-L3: 0.82-0.83).

Approach	L0	L1	L2	L3	L4	L5	L6	L7	L8	All
Vector-RAG	**0.673**	0.808	**0.824**	0.833	0.653	0.516	0.449	**0.486**	0.460	0.691
RAPTOR	**0.673**	0.835	0.797	0.917	0.776	**0.581**	0.500	0.444	**0.476**	**0.711**
GraphRAG	0.647	0.742	0.410	0.750	0.211	0.570	0.203	0.139	0.159	0.542
GFM-RAG	0.638	**0.844**	0.779	**1.000**	**0.796**	**0.581**	**0.746**	0.292	0.175	0.705

Table 2: LLM-as-Judge score for each approach across question levels.

GraphRAG demonstrates limited utility beyond basic extraction tasks.

5.3 Metrics comparative analysis

Multi-metric evaluation (details are available in Table 6 of Appendix E) reveals significant measurement discrepancies across assessment approaches. BERTScore maintains consistently high scores (0.86-0.91) across all systems and levels, suggesting preservation of semantic similarity even when factual accuracy suffers. SQUAD Exact Match demonstrates extremely low performance (0.0-0.07) across all systems, indicating minimal exact string matching between generated and reference answers. SQUAD F1 shows moderate performance (0.3-0.5) with significant fluctuations, suggesting partial word overlap between predictions and references.

LLM-based metrics (Figures 13, 14) provide more nuanced assessment. LLMLogScore shows convergence at L3-L4 (0.95-1.00 across systems), then diverges substantially, with some systems recovering at L7-L8 while others decline. LLM-Judge shows varying patterns by system, with performance peaking at level 3 then generally declining, though with significant differences between systems at higher levels.

GraphRAG shows a significant discrepancy between LLMLogScore and LLM-Judge, particularly at higher difficulty levels. Both metrics peak at Level 3, suggesting this is GraphRAG's optimal complexity zone. However, at levels 6-8, LLMLogScore remains moderate (~ 0.6) while LLM-Judge drops severely (~ 0.2). This suggests GraphRAG retrieves semantically relevant information but fails to synthesize factually correct answers at high complexity. The system appears to hit a complexity ceiling beyond Level 3-4, where it likely produces semantically similar but structurally different answers that fall into a "gray area" - good enough for high semantic similarity scores but not meeting the binary judge's correctness threshold.

The benchmark appears to pose increasingly challenging questions at higher levels, as evidenced by the decline in performance in LLM-based metrics. The contrast between BERTScore (consistently high) and SQUAD metrics (consistently low) suggests that responses maintain word-level semantic similarity to references, without exact matching. In combination, this indicates the challenging nature of the benchmark, where similar but incorrectly retrieved context can lead to responses with good token-level semantic similarity, but where nuances in compositionality significantly impact the conveyed meaning.

Our multi-metric evaluation approach uses BERTScore and LLM-Judge to assess semantic correctness while maintaining awareness of formatting precision. The universally low EM scores across all systems and complexity levels suggest that reference answer formatting, rather than content accuracy, drives these results. Moreover, FinDoc-RAG evaluates general financial document understanding—spanning numerical data, legal terms, and marketing content—rather than specialized numerical reasoning tasks. In deployment scenarios, post-processing can standardize formatting, making semantic accuracy the primary criterion for RAG system selection. Developing unified metrics encompassing both semantic understanding and numerical precision represents important future work beyond this paper's scope.

6 Conclusion

Financial institutions increasingly rely on RAG systems for document processing, yet systematic evaluation on industry-specific content has remained limited. We assessed four state-of-the-art RAG architectures using *FinDoc-RAG*, a benchmark comprising 600+ question-answer pairs from real financial documents across nine complexity levels targeting factual extraction, information integration, and cross-document synthesis.

Our evaluation reveals that architectural choice impacts performance on different question types: while leading systems achieve 0.84 accuracy on basic extraction, performance drops substantially for complex synthesis tasks (0.31 average), with architectural differences amplifying at higher complexity levels. Semi-structured approaches (RAPTOR) provide the most consistent performance across complexity levels, while knowledge graph augmentation (GFM-RAG) excels at complex reasoning but shows variable baseline performance. Our analysis reveals that no single architecture dominates across all task types.

These findings highlight the need of benchmarks like *FinDoc-RAG* for measuring progress toward reliable financial document understanding systems.

Limitations

We acknowledge several limitations of *FinDoc-RAG*. First, while our approach to generating QA pairs using LLMs across different complexity levels provides a comprehensive evaluation framework, automatically generated questions may occasionally lack the depth that human-crafted questions might offer. Despite our manual validation showing high correctness rates, there remains inherent variability in the LLM output that could affect the quality of the question. Second, while our collection of documents spans multiple types of financial documents, it still represents a subset of the vast landscape of financial documentation. Finally, our metrics comparison reveals challenges in accurately measuring RAG system performance, suggesting that even our multifaceted evaluation approach may not capture all dimensions of answer quality relevant to financial domain experts.

It is important to note that our evaluation focuses on RAG architectures, which currently represent the most prevalent and practical approach for incorporating external knowledge into LLM-based question answering. This is due to their ability to maintain data locality and provide retrieval transparency - critical requirements in regulated financial environments. In contrast, long-context LLMs face significant deployment challenges in financial institutions, including prohibitive high computational costs for inference over a large document collections and challenges in explaining retrieval decisions.

Acknowledgements

This work has been supported by UBS Switzerland AG and its affiliates.

References

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. Tomayto, Tomahto. Beyond Token-level Answer Equivalence for Question Answering Evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305. Association for Computational Linguistics.

Jian Chen, Peilin Zhou, Yining Hua, Loh Xin, Kehui Chen, Ziyuan Li, Bing Zhu, and Junwei Liang. 2024a. FinTextQA: A dataset for long-form financial question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6025–6047, Bangkok, Thailand. Association for Computational Linguistics.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. 2022. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chanyeol Choi, Jihoon Kwon, Jaeseon Ha, Hojun Choi, Chaewoon Kim, Yongjae Lee, Jy yong Sohn, and Alejandro Lopez-Lira. 2025. Finder: Financial dataset for question answering and evaluating retrieval-augmented generation. *Preprint*, arXiv:2504.15800.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization. *Preprint*, arXiv:2404.16130.

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.

Wenbo Hu, Jia-Chen Gu, Zi-Yi Dou, Mohsen Fayyaz, Pan Lu, Kai-Wei Chang, and Nanyun Peng. 2025. MRAG-bench: Vision-centric evaluation for retrieval-augmented multimodal models. In *The Thirteenth International Conference on Learning Representations*.

Shantanu Jain. 2025. [link].

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Kuan Li, Liwen Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Shuai Wang, and Minhao Cheng. 2025. Lara: Benchmarking retrieval-augmented generation and long-context llms – no silver bullet for lc or rag routing. *Preprint*, arXiv:2502.09977.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. Leveraging Large Language Models for NLG Evaluation: A Survey. *Preprint*, arXiv:2401.07103.

Linhao Luo, Zicheng Zhao, Gholamreza Haffari, Dinh Phung, Chen Gong, and Shirui Pan. 2025. Gfm-rag: Graph foundation model for retrieval augmented generation. *Preprint*, arXiv:2502.01113.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems*, volume 37, pages 95963–96010. Curran Associates, Inc.

Nghia Trung Ngo, Chien Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2024. Comprehensive and practical evaluation of retrieval-augmented generation systems for medical question answering. *Preprint*, arXiv:2411.09213.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *Preprint*, arXiv:2408.10343.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. SPIQA: A dataset for multimodal question answering on scientific papers. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Zafaryab Rasool, Stefanus Kurniawan, Sherwin Balugo, Scott Barnett, Rajesh Vasa, Courtney Chesser, Benjamin M. Hampstead, Sylvie Belleville, Kon Mouzakis, and Alex Bahar-Fuchs. 2024. Evaluating llms on document-based qa: Exact answer selection and numerical extraction using cogtale dataset. *Natural Language Processing Journal*, 8:100083.

Ehud Reiter. *Natural Language Generation*.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. 2022. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States. Association for Computational Linguistics.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*.

Kyunghwan Sohn, Sunjae Kwon, and Jaesik Choi. 2021. The global banking standards QA dataset (GBS-QA). In *Proceedings of the Third Workshop on Economics and Natural Language Processing*, pages 19–25, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A. Rossi, and Dinesh Manocha. 2025. Visdom: Multi-document qa with visually rich elements using multimodal retrieval-augmented generation. *Preprint*, arXiv:2412.10704.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *Preprint*, arXiv:2401.15391.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges. *Preprint*, arXiv:2406.12624.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Dataset generation details

A.1 Converting documents to Markdown text

The original documents are provided in PDF[2] and are professionally typeset for human consumption. As a result, they feature multi-column layouts, text flowing around tables and figures, and a variety of typographical elements, including bold headers, bullet points, footnotes, superscripts, and subscripts. Since PDF documents are designed for visual presentation, text elements are absolutely positioned. As such, they may not be stored in the same order as they appear on the page. Further, headings and body text are kept in separate text boxes that differ only in font size. Without any additional distinguishing features, this poses a challenge for machines. Consequently, many PDF software packages rely on a complex set of heuristics that are prone to error and may introduce artifacts during content extraction. To generate a faithful Markdown representation from PDF documents, we developed a three-step conversion pipeline:

1. **Image Conversion:** Each PDF page is converted into a flat image that captures its complete visual layout, including multi-column arrangements, figures, and other graphical elements, which helps preserve the original presentation.

2. **Text Extraction:** In parallel, we programmatically extract the text content from each page. Although this extraction is a best-effort process that likely omits layout details or includes minor artifacts, it produces a reliable reference of the page's content.

3. **Markdown Generation:** Both the page image and the extracted text are supplied to a vision-enabled Large Language Model (LLM). The text acts as guidance to reduce hallucinations while the image provides visual context. The LLM generates a Markdown output that preserves key stylistic and structural elements such as headers, bold and italic text, bullet points, tables, and hyperlinks. Further, explicit rules are applied to handle layout features that do not have direct Markdown counterparts (e.g., footnotes, super, and subscripts).

A key challenge in our pipeline is maintaining continuity across pages, especially given variations in header levels and layout elements that span multiple pages, such as tables without repeated headers. To address these issues, we implemented a rolling *'continuity bridge'* text (see algorithm 1 in Appendix A). The process is repeated for each subsequent page until the entire document is processed. No additional cleaning was applied beyond the structured conversion process described above. All original features were preserved exactly as they appear in the source documents, including spellings, product names or hyphenated words at page breaks. This ensures that the Markdown representation most accurately mirrors the original PDF documents.

A.2 Document Processing

The pseudo-code 1 describes the algorithm used to process a document.

[2] Portable Document Format

Cluster	Field	Content
0	Filenames	95be, 91f6, f059, 095f
	Main product or service	UBS duo Saving; UBS Investment Fund Account; UBS Fixed Term Deposit; Foreign Exchange (FX) & Precious Metal (PM) Spot, Forward & Swaps
	Coherence	95be, 91f6 and f059 cover UBS retail savings/investment products with similar terms, while 095f lists FX & PM mark-ups for the same clientele, making it a mild outlier.
1	Filenames	3996, b7e1, 3000, e3da, 6592, 6253, 2f9d, 8b09
	Main product or service	UBS Visa Corporate Card; UBS Commercial Credit Cards; Power of Attorney for UBS Commercial Cards; UBS Platinum Credit Card; UBS Travel Insurance Plus; UBS Gold Credit Card
	Coherence	All files concern UBS credit cards: offering, product sheets, insurance add-ons, and legal/administrative details.
2	Filenames	eec1, 9082, a02d, 7f44
	Main product or service	UBS Vitainvest Funds Sustainable; UBS Vitainvest World 25 Sustainable U; UBS Vitainvest Swiss 75 Sustainable U; BVG 21 Reform
	Coherence	eec1, 9082 and a02d are Vitainvest Sustainable fund sheets sharing Swiss-pension and ESG themes; 7f44 adds broader BVG reform context. All fit a "Swiss sustainable retirement investing" topic.
3	Filenames	29e9, 9542, fde3, 873f, 6c47
	Main product or service	UBS Investment Fund Account; UBS key4 smart investing; UBS Personal Account; UBS Manage [CH]; UBS key4 Banking; UBS me Banking Package; UBS Fisca Account; UBS Vested Benefits Account; UBS Investment Funds
	Coherence	29e9, 9542 and 873f focus on investment or discretionary-management offers; fde3 and 6c47 outline the core account and fee framework. Common threads are low entry thresholds, digital access and sustainability, with the payments documents forming the loosest link.

Table 3: Overview of document clusters generated during creating of L5-L8 QAs.

Algorithm 1 Document Processing Across Pages

For Page 1:
Input: Image and extracted text of page 1
Output:
    Markdown text for page 1
    Continuity bridge text describing the structural context on page 1

For Page k+1:
Input:
    Image and extracted text of page k+1
    Continuity bridge text from page k
Output:
    Markdown text for page k+1
    Updated continuity bridge text for page k+1

B Settings for Question Generation, Experiments and Evaluation

B.1 Question generation Settings

QA pairs of levels L0-L3 were generated with the OpenAI model `gpt-4o`, whereas levels L4-L8 were generated using `gpt-4o-mini`. For generating embeddings of Levels L5-L8, the OpenAI `text-embedding-ada-002` model was used.

B.2 RAG-QA Settings

General setup: All algorithm use OpenAI's `gpt-4o-mini` as completion model LLM and `text-embedding-ada-002` as embedding type.

Vector-based: This is an ad hoc implementation of the standard RAG pipeline (Lewis et al., 2021). The preprocessing step includes a chunking of each document using Tiktoken (Jain, 2025)'s tokenizer with the `text-embedding-ada-002` encoding. Each chunk has a size of 100 tokens. The selection of the best chunks is made by minimizing the cosine distance between the input query and all the chunks available in the embedding space.

RAPTOR: (Sarthi et al., 2024) Using the RAPTOR's building algorithm, we created a tree for each document, then all the trees have been merged together (this is a custom change since the available implementation does not support multiple documents). RAPTOR retrieval process has been used with the collapsed tree parameter set to `TRUE`. All other parameters have been kept with default values.

Graph-RAG: (Edge et al., 2025) Despide the changes to use the standard models, this experiment has been run with default parameters using local search mode.

KG-RAG: For KG-RAG, we followed the baseline settings from (Luo et al., 2025) with the 8M-pretrained model [3]. Using the default LLM prompting set-ups, we create the KG from the document sets. For the entity-linking module, the ColBERTv2 model (Santhanam et al., 2022) was employed with a baseline cosine similarity of 0.8 and a maximum default of 100 similar neighbours. This controls the number of synonymous edges to be added between similar entities during the entity-linking phase.

B.3 Evaluation settings

BERTScore evaluations were run with an off-the-shelf `roberta-large` model trained on English texts. LLMJudge and LLMLOgScore were both run using the OpenAI model gpt-4o-mini.

C Dataset Analysis

C.1 Dataset composition

Dataset composition analysis is presented in Figures 4, 5 and 6.

C.2 Complexity and Diversity

Figures 7, 8, 9, and 10 depict the statistical analysis of the dataset in terms of complexity and diversity.

C.3 Information analysis

The information analysis is depicted in Figures 11 and 12.

[3] https://huggingface.co/rmanluo/GFM-RAG-8M

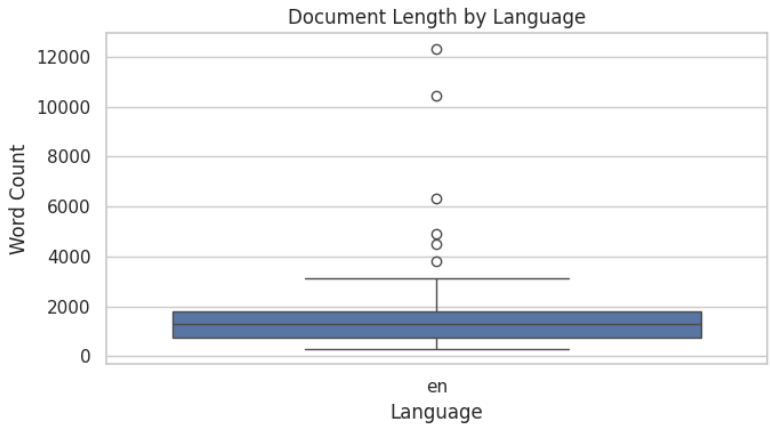


Figure 4: Document length in words.

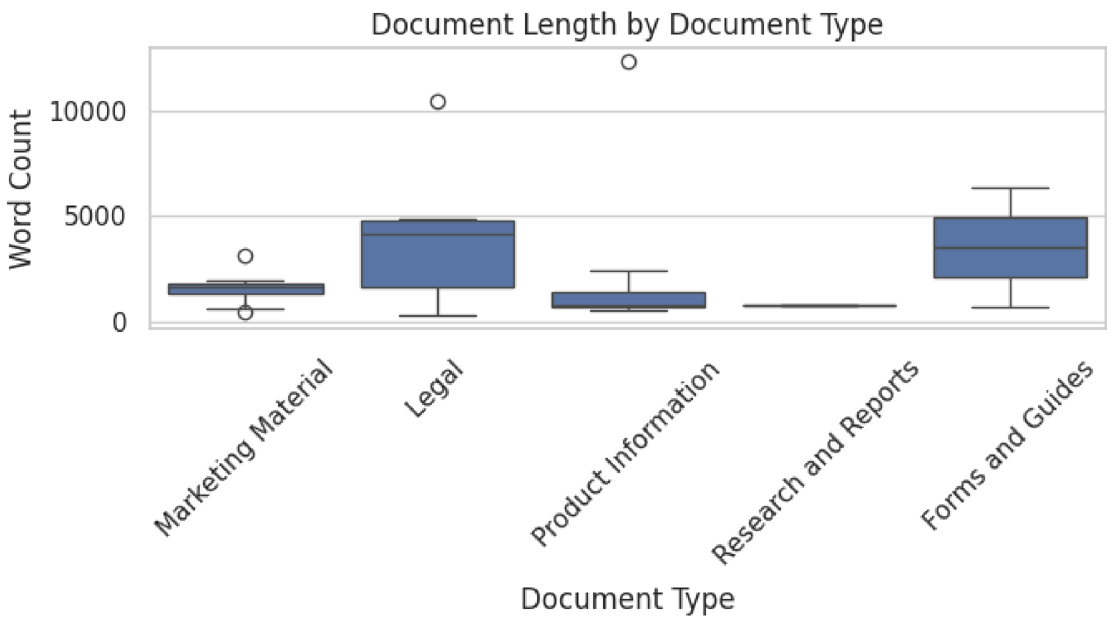


Figure 5: Document length in words by document type.

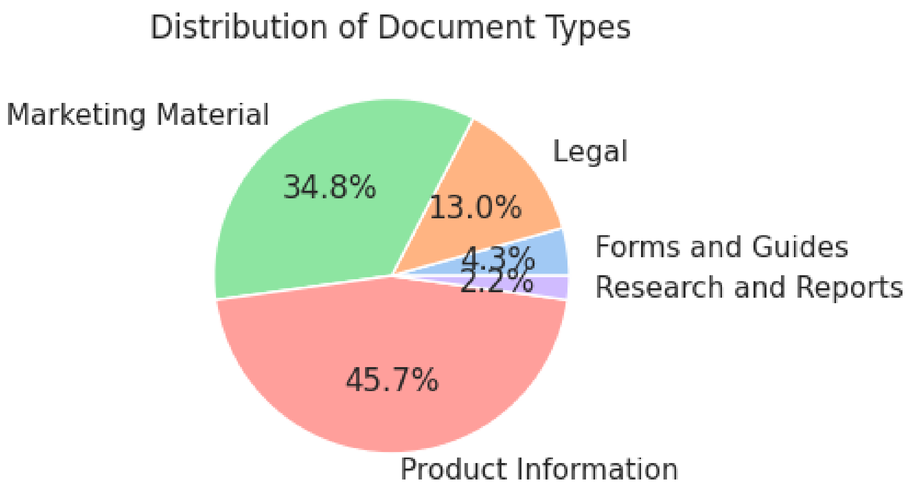


Figure 6: Distribution by document types.

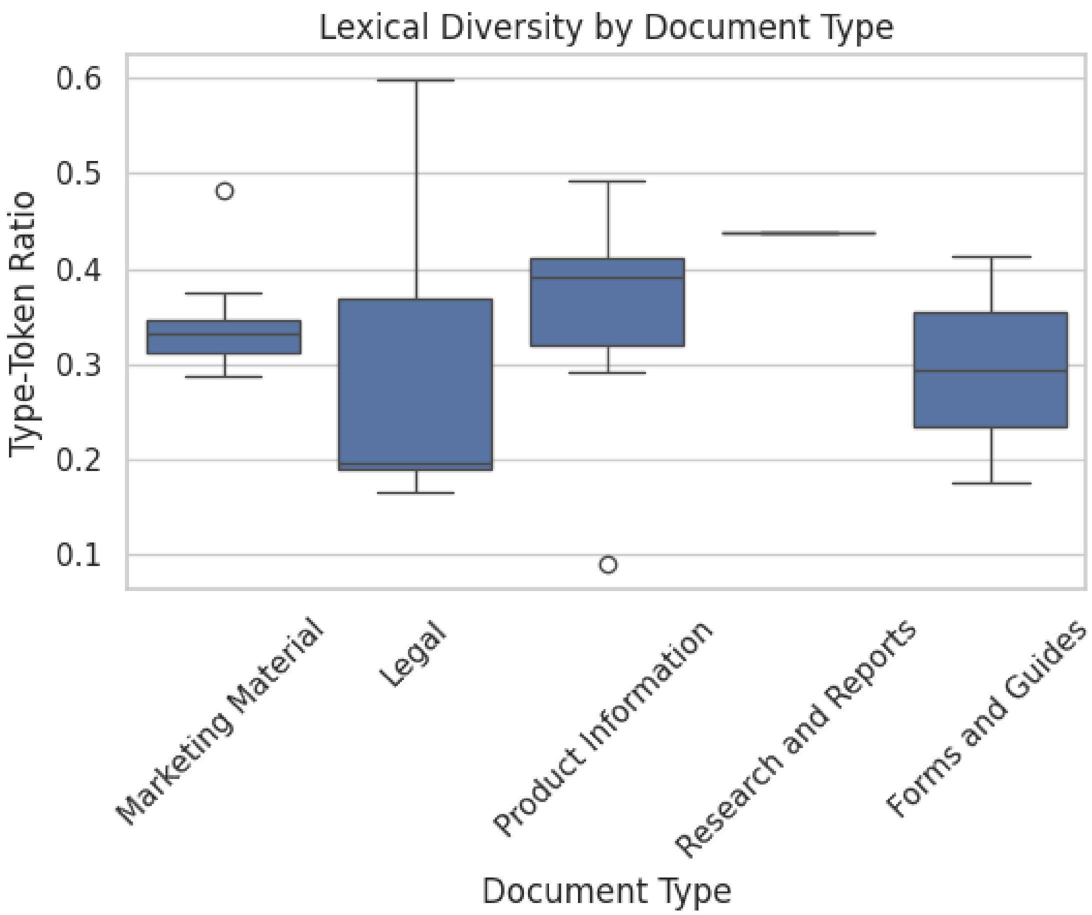


Figure 7: Lexical diversity (i.e., type-token ratio - TTR) by type.

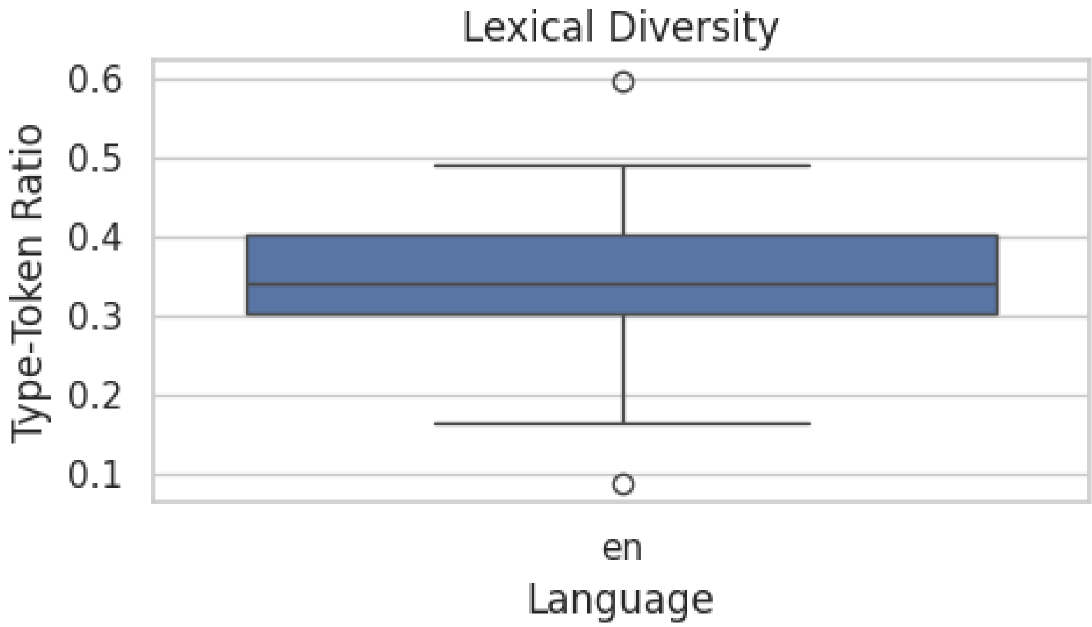


Figure 8: Lexical diversity (i.e., type-token ratio - TTR) by language.

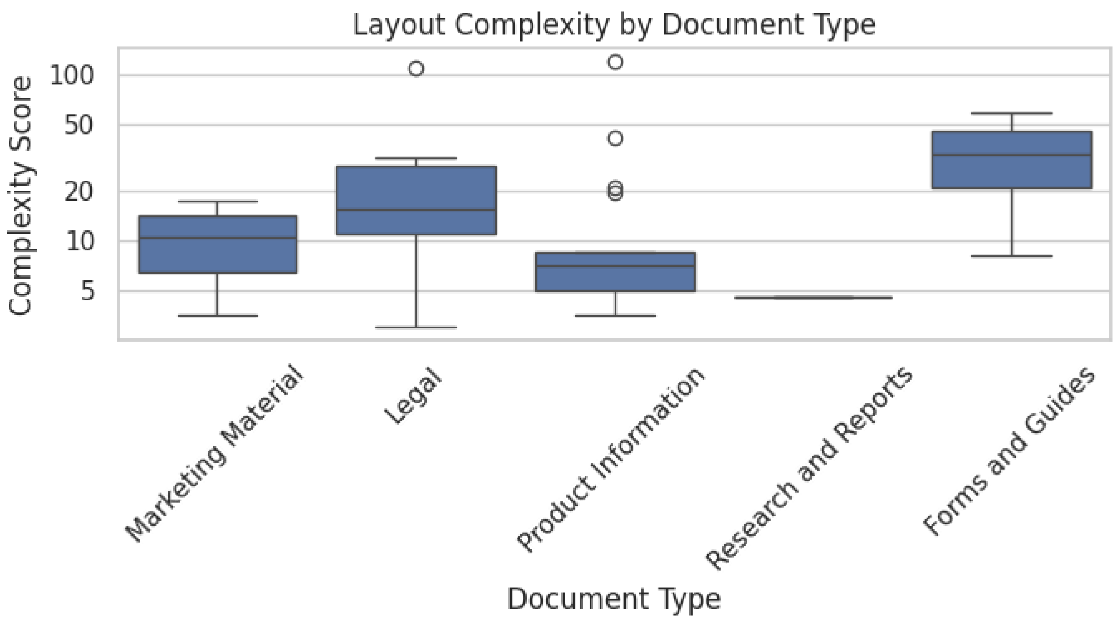


Figure 9: Layout complexity consists in an heuristic based on number of tables, section and images and, the maximum nesting level of sections.

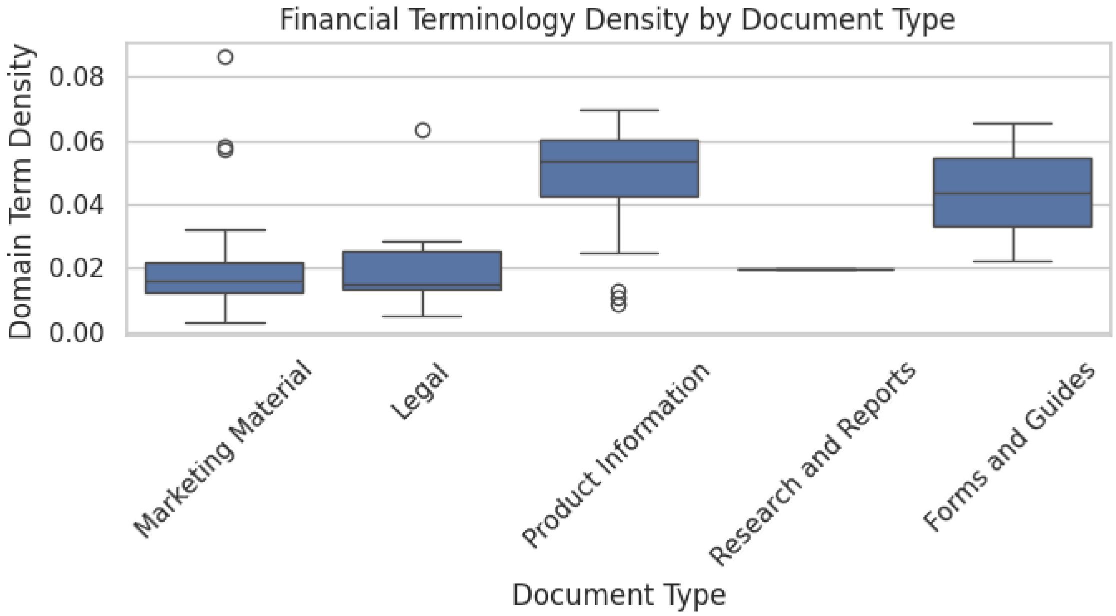


Figure 10: Density of financial terms (account, asset, balance, bond, capital, credit, debt, dividend, equity, fund, interest, investment, liability, mortgage, portfolio, risk, share, stock, tax, yield) by document type.

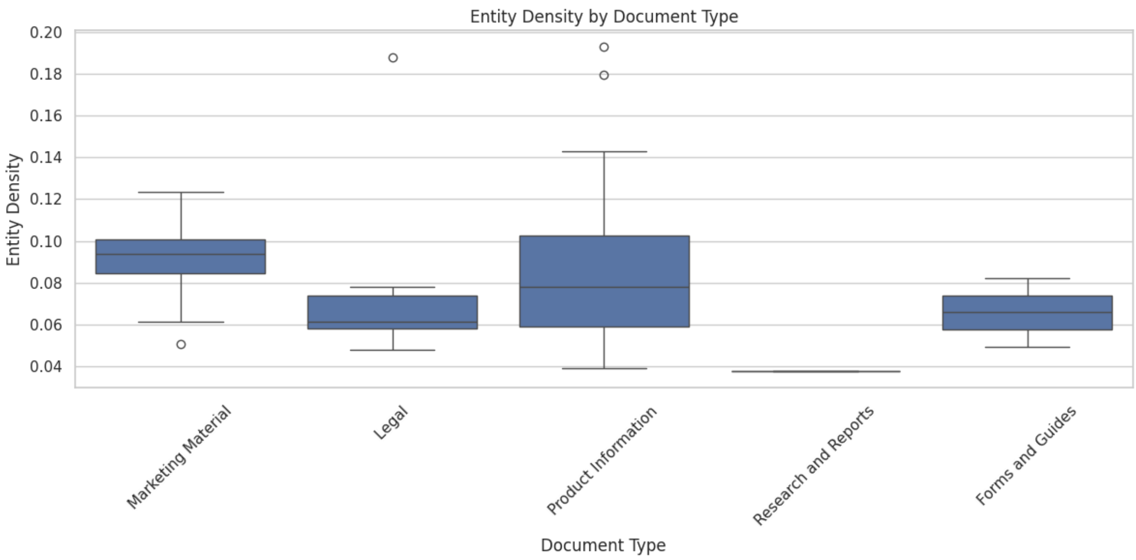


Figure 11: Measure information density using entity density as a proxy (i.e., ratio of entities by tokens).

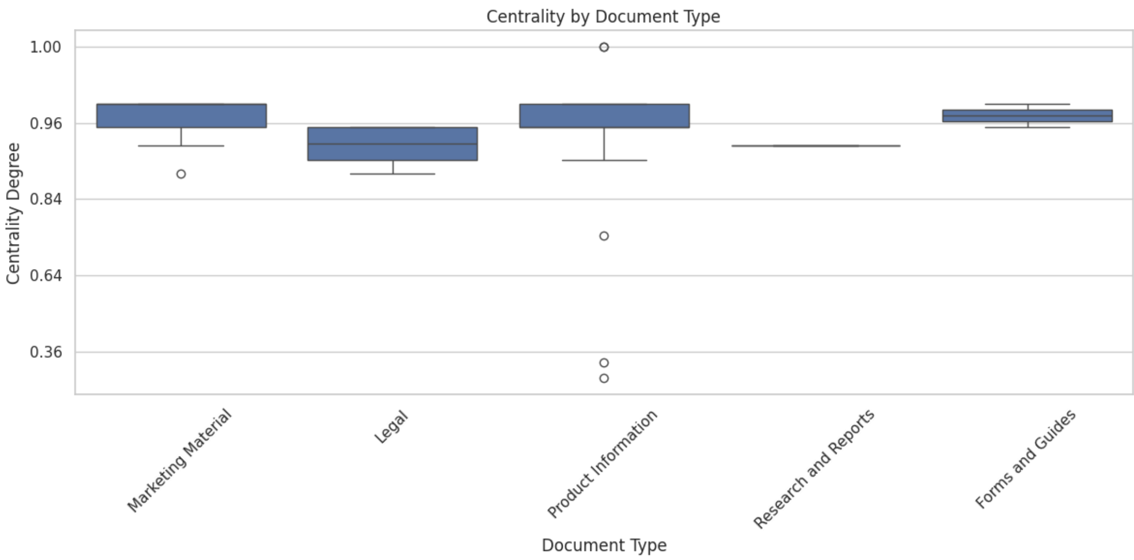


Figure 12: Document's centrality degree by document type.

D Human Validation Details

To evaluate the quality of the Q&A dataset we constructed, we conducted a human evaluation focusing on two key aspects: **Ambiguity** and **Correctness**.

For this evaluation, a randomly sampled representative subset of Q&A pairs was manually reviewed for each level. Correct and non-ambiguous examples are shown in table 5, meanwhile incorrect and/or ambiguous examples are shown in table 4. The evaluation criteria were defined as follows:

- **Ambiguity (Question-level)**: A question was marked as ambiguous if it met **any** of the following conditions:
  1. The question was unclear or poorly formulated.
  2. The question was too general, allowing multiple distinct answers to be considered valid.
  3. The question included vague or forbidden referents (e.g., "in this story"), which are not self-contained or interpretable without external context.
- **Correctness (Answer-level)**: An answer was marked as correct if it satisfied **all** of the following conditions:
  1. It directly addressed the question being asked.
  2. It was factually accurate and faithful to the content of the source document(s).
  3. It was complete, providing all relevant and necessary details to fully answer the question.

For levels **4 through 8**, the Q&A generation process follows a slightly different paradigm: questions at these levels are intended to reflect the perspective of users with limited prior knowledge of the subject matter. As a result, the evaluation criteria were applied with a more relaxed interpretation. Specifically:

- Questions were allowed to be more general, provided they remained clear and self-contained.
- Answers were not required to include every possible detail, as long as they remained accurate and sufficiently informative given the context and intent.

E Benchmark evaluation

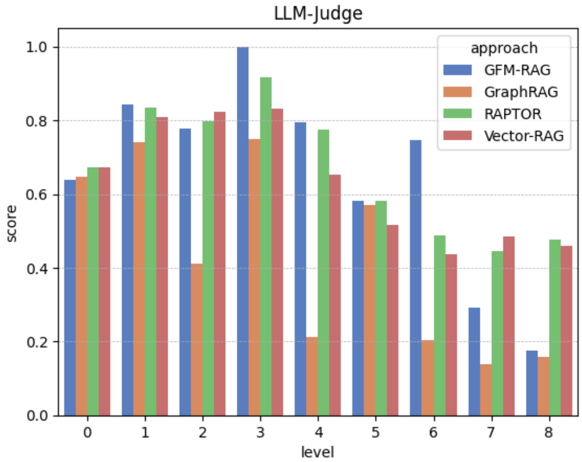


Figure 13: LLM-as-Judge comparison of RAG-QA approaches across question levels.

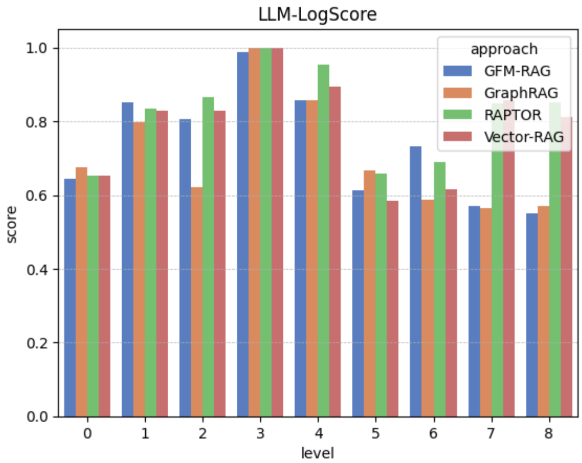


Figure 14: LLMLogScore comparison of RAG-QA approaches across question levels.

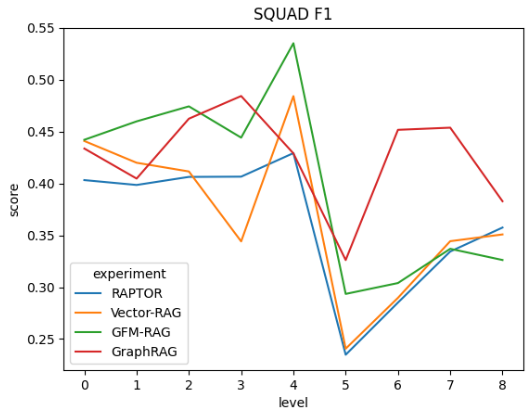


Figure 15: SQUAD-F1 comparison of RAG-QA approaches across question levels.

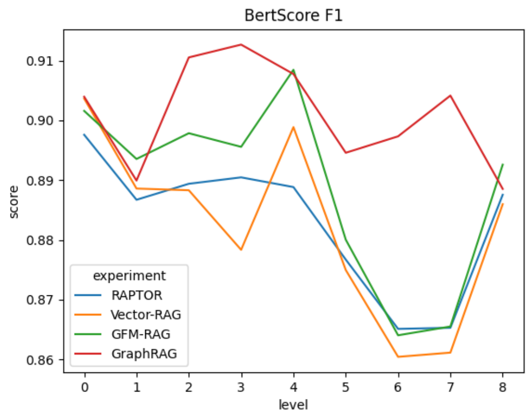


Figure 16: BERTScore-F1 comparison of RAG-QA approaches across question levels.

Group: Incorrect and/or non-ambiguous examples
Example 1 **Q:** What does the UBS Gold Credit Card offer for airport lounge access? **A:** The card provides membership to Priority Pass for access to over 1300 lounges worldwide. **LEVEL:** 5 **ID:** 99288f2fd571eb545bb18ba4b33caa31b6adb8eb7f6c92953b7665fae34fdc48 **Ambiguity:** No **Correctness:** No - Answer is missing key detials, and therefore is uncomplete. **Source:** 8b09 **Example 2** **Q:** What is this "infrastructure investment" that you are talking about? **A:** Infrastructure investment refers to investments in large-scale assets that are critical to economic productivity, such as regulated utilities, transportation systems, power stations, and telecommunications capabilities. These investments are expected to provide a combination of steady income and potential for capital appreciation. They are seen as resilient to macroeconomic and interest rate pressures, offering diversification benefits and stable long-term cash flows. The document highlights that infrastructure investments could be particularly timely due to fiscal initiatives like the Inflation Reduction Act in the U.S. and the Green Deal in Europe, which may funnel investments into infrastructure assets for years to come. **LEVEL:** 4 **ID:** 70b7b8445c454663b7a9abb5420495910155266f6ed035657e9e823d93108c13 **Ambiguity:** Yes - the term "you are talking about" is ambiguous. The question includes a vague or forbidden referent. **Correctness:** Yes **Source:** 2642 **Example 3** **Q:** For UBS Commercial Credit Cards, what happens if the card is used for illegal purposes? **A:** The use of the card for illegal purposes is prohibited. **LEVEL:** 0 **ID:** 42142c0776f5e06328d75874d498712e1740d9f8f35fbcfd1c81cf03eeaaf70e **Ambiguity:** Yes - The question is poorly formulated. It is unclear whether it refers to (1) the actions a cardholder must take in the event of illegal card use, (2) UBS's or holder response or liability in such a case, or (3) the legal consequences for the cardholder. As written, it leaves room for multiple interpretations. **Correctness:** No - The answer do not address the question and it is not faithful to the content of the source document, nor presents all the necessary detials. **Source:** 2f9d

Table 4: Incorrect and/or non-ambiguous examples in our human evaluation.

Group: Correct and non-ambiguous Examples
Example 1 **Q:** What is included in the UBS key4 Banking package? **A:** The UBS key4 Banking package includes personal and savings accounts, debit and credit cards, and digital banking services. **LEVEL:** 5 **ID:** 0d43c3a2df5f78dc8704238b5e43c093e502be215e8704dd88decde92ed62ab1 **Ambiguity:** No **Correctness:** Yes **Source:** 6c47
Example 2 **Q:** How do the investment strategies of the UBS Duo Saving and the UBS Fixed Term Deposit cater to different investor needs? **A:** The UBS Duo Saving caters to investors looking for both stability and growth by splitting investments between a fixed-term deposit and an investment component, while the UBS Fixed Term Deposit is tailored for risk-averse investors seeking security with a fixed interest rate. The Duo Saving states, the product aims to balance stability and growth, while the Fixed Term Deposit is described as designed for risk-averse investors. **LEVEL:** 6 **ID:** d975a031c131f7a4b050e84f42acf45f6fd27cfc6f3ae027a263f768073060de **Ambiguity:** No **Correctness:** Yes **Source:** f059, 95be
Example 2 **Q:** What is the management fee per annum for the UBS Vitainvest Swiss 75 Sustainable U fund? **A:** 1.20% **LEVEL:** 0 **ID:** e88bf375f47e9bb251dcdeefc5eba2a0682b7f0cb4e1e477bd5426991d7c69bf **Ambiguity:** No **Correctness:** Yes **Source:** eec1

Table 5: Correct and non-ambiguous examples in our human evaluation.

Approach	level	Q_count	llmjudge (std)	llmlogscore	bertscore_F1	squad_EM	squad_F1
GFM-RAG	0	201	0.638 (0.010)	0.644	0.902	0.070	0.442
GFM-RAG	1	182	0.844 (0.005)	0.852	0.894	0.033	0.460
GFM-RAG	2	74	0.779 (0.023)	0.806	0.898	0.000	0.474
GFM-RAG	3	4	1.000 (0.000)	0.988	0.896	0.000	0.444
GFM-RAG	4	49	0.796 (0.033)	0.858	0.908	0.020	0.535
GFM-RAG	5	31	0.581 (0.000)	0.613	0.880	0.000	0.294
GFM-RAG	6	46	0.746 (0.037)	0.732	0.864	0.000	0.304
GFM-RAG	7	24	0.292 (0.000)	0.569	0.866	0.000	0.337
GFM-RAG	8	21	0.175 (0.059)	0.551	0.893	0.000	0.326
GraphRAG	0	201	0.647 (0.015)	0.676	0.904	0.035	0.434
GraphRAG	1	182	0.742 (0.027)	0.797	0.890	0.000	0.405
GraphRAG	2	74	0.410 (0.006)	0.622	0.911	0.000	0.463
GraphRAG	3	4	0.750 (0.204)	1.000	0.913	0.000	0.484
GraphRAG	4	49	0.211 (0.019)	0.856	0.908	0.000	0.429
GraphRAG	5	31	0.570 (0.040)	0.667	0.895	0.000	0.326
GraphRAG	6	46	0.203 (0.027)	0.588	0.897	0.000	0.452
GraphRAG	7	24	0.139 (0.052)	0.565	0.904	0.000	0.454
GraphRAG	8	21	0.159 (0.022)	0.570	0.889	0.000	0.383
RAPTOR	0	201	0.673 (0.009)	0.654	0.898	0.025	0.403
RAPTOR	1	182	0.835 (0.008)	0.835	0.887	0.027	0.399
RAPTOR	2	74	0.797 (0.029)	0.866	0.889	0.014	0.406
RAPTOR	3	4	0.917 (0.118)	1.000	0.890	0.000	0.407
RAPTOR	4	49	0.776 (0.044)	0.954	0.889	0.000	0.429
RAPTOR	5	31	0.581 (0.026)	0.659	0.877	0.000	0.235
RAPTOR	6	46	0.500 (0.031)	0.690	0.865	0.000	0.285
RAPTOR	7	24	0.444 (0.104)	0.848	0.865	0.000	0.334
RAPTOR	8	21	0.476 (0.000)	0.851	0.888	0.000	0.358
Vector-RAG	0	201	0.673 (0.006)	0.652	0.904	0.035	0.441
Vector-RAG	1	182	0.808 (0.009)	0.830	0.889	0.027	0.420
Vector-RAG	2	74	0.824 (0.022)	0.828	0.888	0.000	0.412
Vector-RAG	3	4	0.833 (0.118)	0.998	0.878	0.000	0.344
Vector-RAG	4	49	0.653 (0.050)	0.893	0.899	0.000	0.484
Vector-RAG	5	31	0.516 (0.000)	0.584	0.875	0.000	0.241
Vector-RAG	6	46	0.449 (0.041)	0.617	0.860	0.000	0.290
Vector-RAG	7	24	0.486 (0.071)	0.858	0.861	0.000	0.344
Vector-RAG	8	21	0.460 (0.045)	0.811	0.886	0.000	0.351

Table 6: System performance across question difficulty levels. Q_count indicates number of questions per level. Evaluation metrics: llmjudge (LLM-as-judge accuracy), llmlogscore (log probability scores), bertscore_F1 (semantic similarity), squad_EM (exact match), squad_F1 (token-level F1).

F Prompts for generating level 4 questions

Prompt 1 : question generation prompt

system : You are client_profile_name: Here is the description of your profile: client_profile Ensure that you always write in the style associated with your assigned profile.
user: Read the following markdown document describing a banking product and generate a **simple, naïve question** about it. The question should be something a person with **no prior knowledge of banking** might ask when encountering this product for the first time. Assume the person has **little to no financial expertise** and is genuinely curious about basic concepts. **Guidelines for the question:** - It should be **basic and straightforward**, avoiding complex financial terminology. - It should reflect **genuine curiosity**, as if someone is trying to understand the very basics. - The question **must explicitly reference the banking product** (e.g., *"a savings account"*, *"this type of loan"*, *"this investment plan"*) instead of using vague words like "this" or "it." - The answer **must be found within the document**—do not ask questions unrelated to the content. - Do **not** add explanations or extra context—**just generate the question**. Wrap the question with the «Q» and «/Q» tags. ### Banking Product Description (Markdown Format): “‘markdown banking_markdown “‘

Prompt 2 : answer generation prompt

system: Your task is to: 1. Read the provided markdown document describing a banking product, and the provided question. 2. First, answer the question using the information from the markdown document. Wrap the answer with the «A» and «/A» tags. 3. If it’s not possible to answer given the document, answer with «A» No answer «/A».
user: ### Banking Product Description (Markdown Format): “‘markdown banking_markdown “‘ ### Question: question

Prompt 3 : quotation generation prompt

system: Your task is to: 1. Read the provided markdown document describing a banking product, and the provided question. 2. Provide a quotation from the document that answers the provided question. When quoting, wrap the quotation with «Quot» and «/Quot» tags. 3. If no quotation answers the question, answer with «Quot» No quotation «/Quot».
user:### Banking Product Description (Markdown Format): “‘markdown banking_markdown “‘ ### Question: question

Do Companies Reveal Their Own Fraud? – A Novel Data Set for Fraud Detection Based on 10-K Reports

Moustafa Amin[1] and Matthias Aßenmacher[1,2]

[1]Department of Statistics, LMU Munich,
[2]Munich Center for Machine Learning (MCML), LMU Munich
Correspondence: matthias@stat.uni-muenchen.de

Abstract

This work aims to gather and analyze data for text-based fraud detection using data from financial disclosures – specifically, the Management's Discussion and Analysis (MDA) sections of 10-K reports submitted to the US Securities and Exchange Commission. We provide a comprehensive overview of the process for creating the data set and introduce the resulting data set as an open-source resource for future research in the financial natural language processing domain. We subsequently train a range of machine learning and deep learning classifiers on the MDA text, intending to provide reasonable baselines for future researchers and to offer insight into the nature of fraudulent disclosures and how such data can be effectively used for uncovering fraud.

1 Introduction

Getting involved in financial crimes might be one of the most lucrative propositions, financially speaking. In 2024 alone, around $3.1 trillion circulated through the global economy, with $782.9 billion being used in drug trafficking, $346.7 billion in human trafficking, and $11.5 billion in terrorism financing due to financial crimes (Nasdaq Verafin, 2024). Fraud, of course, comprised a notable share of that money, with an estimated loss of $485.6 billion due to fraud in 2023 alone (Nasdaq Verafin, 2024). Naturally, with the amount of money implicated in financial crimes, the number of stakeholders is not scarce – be it regulators, law enforcement, firms, or companies. It is well-documented that more often than not, the more synergistically stakeholders work to prevent financial crimes, the more likely they succeed (Nasdaq Verafin, 2024).

The focus of this research is not financial crimes as a whole but rather fraud, which is a complex phenomenon with different facets. According to the Association of Certified Fraud Examiners' (ACFE) *Report to the Nations* (Association of Certified Fraud Examiners, 2024), most fraudulent activities were uncovered by tips from individuals involved or adjacent to the fraud. Considering the financial damage caused by it, the necessity to improve mechanisms for the detection of financial crimes – and with it, fraud – is obvious. According to the ACFE and *Black's Law Dictionary*, it can be broadly defined as "any activity that relies on deception to achieve a gain" and becomes a crime when, in layman's terms: "you lie to deprive a person or organization of their money or property.". For the purposes of understanding the concept further, the definitions and categorizations of the ACFE are being adopted here.

Contributions. The main contribution of this work is to provide (to the best of our knowledge) the first publicly available data set of (specific sections of) 10-K reports alongside labels indicating fraudulent behavior. To enable other researchers to replicate or extend our work, we provide transparent descriptions of the data scraping and labeling processes, release our source code, and make the data available. We provide baseline results for a given data split motivated by specific temporal characteristics of fraud. Our entire code and the created data set are publicly available:

- **Scraper:** `https://github.com/aminmous/fraud-webscraper`
- **Code:** `https://github.com/aminmous/fraud-analysis`
- **Data:** `https://doi.org/10.5281/zenodo.17121948`

2 Related Work

The Management Discussion and Analysis (MDA), which is the 7th section of an annual report submitted to the U.S. Securities and Exchange Commission (SEC), is unique in that the information

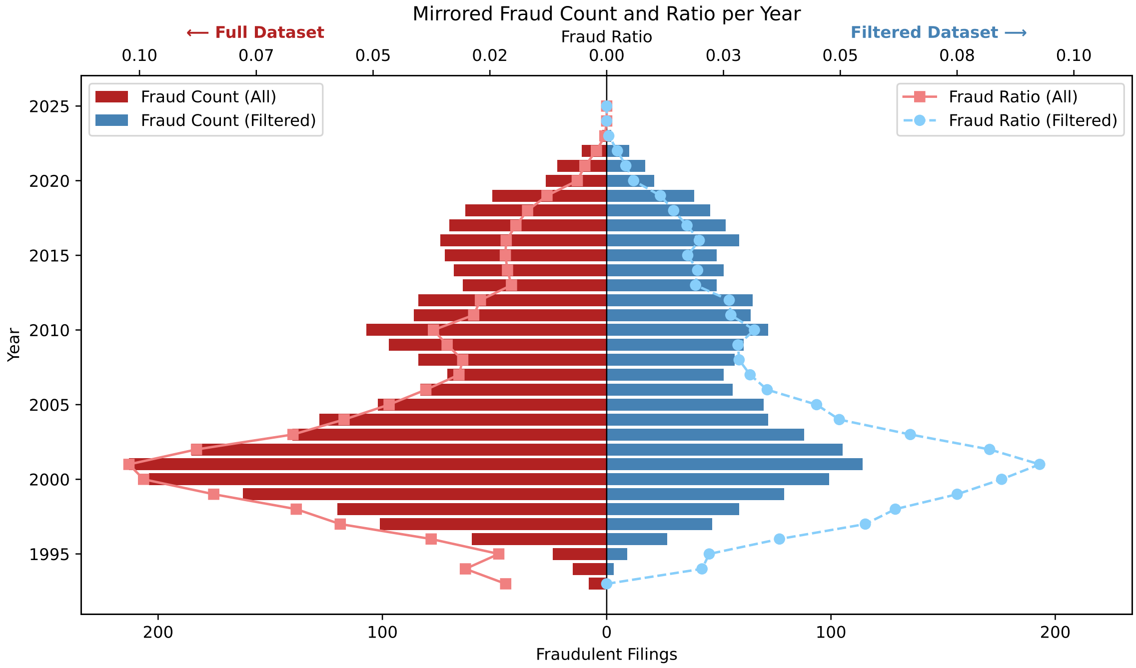


Figure 1: Visualizing the significance of fraud in 10-K reports: Development of the share of fraudulent reports in our newly curated data set over time, both in the raw (left) and in the final, filtered version (right).

contained in it is up to management's discretion and that it is subjective. Unlike other sections, it is not standardized in its structure and content and has thus been studied and scrutinized as a source of information – either by investors to gain an advantage over just evaluating tabular financial data or by researchers to evaluate, whether it can provide insights beyond what is concretely put to paper. Ultimately, firms are run by humans, humans are subject to their own biases and emotions, and why shouldn't this be reflected in MDAs?

Classical Machine Learning. The potential information content of MDAs is supported by work such as by Feldman et al. (2010), where the authors conclude that tone changes in MDA sections in both yearly (10-K) and quarterly (10-Q) reports are associated with market reactions in the short term. Hence, market participants can gauge short-term fluctuations based on the "nonfinancial" content of a report. (Durnev and Mangen, 2020) reveal that the MDA section of the 10-K report has influence over investment and disclosure decisions of other companies, especially in related industries, which suggests that the market not only affects but is affected by MDAs. (Holder-Webb and Cohen, 2007) determine that the quality of the disclosure is influenced by the level of stress that the companies are experiencing. It is therefore plausible to assume that the MDA section of the 10-K report is a good candidate to analyze and use as a basis to gain insights into the state of a firm and its management.

Our work is primarily inspired by Hoberg and Lewis (2017), whose work represents one of the pioneering works in the field of textual analysis of non-financial data and its relationship to fraud. The authors show that MDAs are an "informative setting for understanding fraud" by first showing that firms produce abnormal MDAs compared to their ISA (short for "*Industry, Size and Age*") peers[1], and secondly, by showing that fraudulent firms produce abnormal disclosures compared to their own in years where they did not commit fraud. They also used (tabular) accounting data from COMPUSTAT in conjunction with the MDA as text input and the fact that an AAER (short for "*Accounting and Auditing Enforcement Release*") was filed against the firm as a label for fraud. The AAER is a public document issued by the SEC that details the illegal findings of an investigation concerning a civil lawsuit brought by the SEC against a company, public or private, and/or individuals. AAERs contain valuable information about the nature of the fraud, such

[1] Industry peers are defined as firms with the same two-digit Standard Industrial Classification (SIC) codes: The first two digits define the major industry group (e.g., 25 = Furniture and Fixtures), the first three digits define the industry group (e.g., 252 = Office Furniture), and the full four-digit code specifies the detailed industry (e.g., 2521 = Wood Office Furniture).

as the involved parties, the scope of misconduct, the duration of the fraud, and the violated statutes.

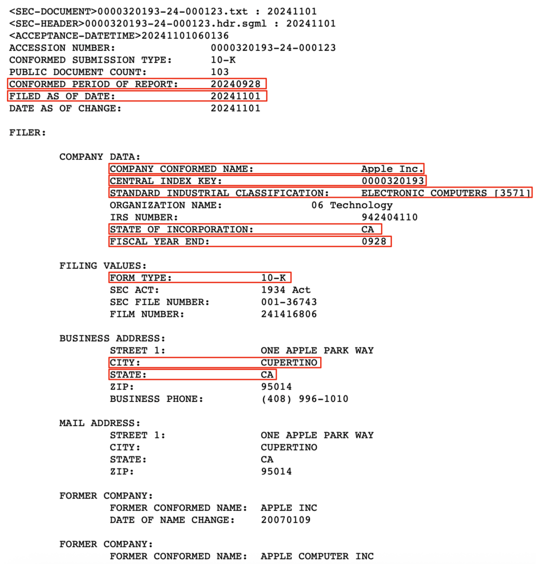
LLM-based Approaches. A methodologically more advanced, albeit theoretically not as extensive, paper that starts to bridge the gap between the work of Hoberg and Lewis (2017) and the modern NLP methods available today for textual analysis is the work by Bhattacharya and Mickovic (2024). The main difference here is the use of BERT (Devlin et al., 2019), instead of topic modeling or sentiment analysis, to analyze MDAs and classify firms as fraudulent or not. The authors also use a data set that combines 28 quantitative financial features from COMPUSTAT with textual data from MDA sections of 10-K filings, and identify fraudulent outcomes using AAER enforcement data. The data set spans the years 1994 to 2013 and focuses on detecting a single category (accounting fraud). Besides BERT, the authors use LDA with 78 topics, selected by maximizing the AUC on their validation set over a range of 10 to 150 topics.

Shortcomings. What unifies all these different works is their lack of publicly available code and data, serving as the main motivation for our work of transparently constructing a benchmark data set for further experimentation in financial NLP.

3 Data Set Construction

3.1 Creating Firm-Year Observations

Firm-years are the unit of observation in the work of (Hoberg and Lewis, 2017); they represent the MDA sections of 10-K reports along the time dimension. We use the same nomenclature to describe the observations in our data. Although no open-source data set is available, the SEC provides a public database called the EDGAR system, which contains all filings submitted to the SEC dating back to 1994, with full coverage starting in 1997 (Hoberg and Lewis, 2017). It can be accessed through web scraping, albeit with some limitations.[2] Each company is assigned its own unique ten-digit SEC identifier, known as the *Central Index Key* (CIK), with a specific form type in mind, a URL form can be submitted via `https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany&CIK=cik&type=formtype&dateb=&owner=exclude&count=40&search_text=`. By replacing the `formtype` with "10-K" and `cik` with the CIKs from the JSON list of CIKs from the SEC's website[3], access to all types of 10-K forms was enabled.[4] We developed a crawler based on the Scrapy library.[5] The crawler extracted the `.txt` version of the filing, as this format was consistently available in a structured format for all periods. It has a standardized header (cf. Figure 2, further details on the extracted variables are provided in Appendix A), providing useful additional information.



```
<SEC-DOCUMENT>0000320193-24-000123.txt : 20241101
<SEC-HEADER>0000320193-24-000123.hdr.sgml : 20241101
<ACCEPTANCE-DATETIME>20241101060136
ACCESSION NUMBER:              0000320193-24-000123
CONFORMED SUBMISSION TYPE:     10-K
PUBLIC DOCUMENT COUNT:         103
CONFORMED PERIOD OF REPORT:    20240928
FILED AS OF DATE:              20241101
DATE AS OF CHANGE:             20241101

FILER:

        COMPANY DATA:
                COMPANY CONFORMED NAME:                 Apple Inc.
                CENTRAL INDEX KEY:                      0000320193
                STANDARD INDUSTRIAL CLASSIFICATION:     ELECTRONIC COMPUTERS [3571]
                ORGANIZATION NAME:              06 Technology
                IRS NUMBER:                             942404110
                STATE OF INCORPORATION:                 CA
                FISCAL YEAR END:                        0928

        FILING VALUES:
                FORM TYPE:              10-K
                SEC ACT:                1934 Act
                SEC FILE NUMBER:        001-36743
                FILM NUMBER:            241416806

        BUSINESS ADDRESS:
                STREET 1:               ONE APPLE PARK WAY
                CITY:                   CUPERTINO
                STATE:                  CA
                ZIP:                    95014
                BUSINESS PHONE:         (408) 996-1010

        MAIL ADDRESS:
                STREET 1:               ONE APPLE PARK WAY
                CITY:                   CUPERTINO
                STATE:                  CA
                ZIP:                    95014

        FORMER COMPANY:
                FORMER CONFORMED NAME:  APPLE INC
                DATE OF NAME CHANGE:    20070109

        FORMER COMPANY:
                FORMER CONFORMED NAME:  APPLE COMPUTER INC
```

Figure 2: 10-K header in `.txt` filing (U.S. Securities and Exchange Commission, 2024)

Regex extraction was not feasible due to inconsistencies in the filings and structural changes, such as the introduction of item 7(a) in 1997, despite improvements to prior regex-based methods used in the work of Bhattacharya and Mickovic (2024). As a result, we turned to the SEC API (SEC API, 2025), a multifaceted (commercial) tool that operates similarly to the official SEC's API in that it allows users to access filing metadata. However, it also offers the additional capability of extracting specific sections from a set of filings.[6]

3.2 Labels

In most contemporary research, the labels are taken from the (closed-source) USC Marshall School of

[2]Requests are capped at 10 per second (U.S. Securities and Exchange Commission, 2023). Violating this policy results in a 10-minute ban from the server.

[3]https://www.sec.gov/files/company_tickers.json

[4]Crawling in this manner also yields various other types of 10-K forms (cf. Appendix B), as they share the same prefix.

[5]https://docs.scrapy.org/en/latest/

[6]Filings before 2002 are less standardized due to the absence of Sarbanes-Oxley (SOX), making MDA extraction via the SEC API less reliable for those years.

Business AAER data set, which can be purchased at several price points depending on the type of user. The data set contains 4,278 AAERs, with 1,816 cases of firm misstatements issued from 1982 to 2021 (University of Southern California, 2021). A non-exhaustive list of AAERs on the SEC's website[7]. Each AAER is numbered, with the latest one posted on the website (as of May 19, 2025) being AAER-4568. AAERs can be issued against all types of entities, even individuals, and 3,317 were available as of May 19, 2025. After filtering to include only enforcement actions taken against companies using the SEC API, the number of relevant AAERs was reduced to a reasonable starting point of 1,223. As only public companies are required to file 10-K reports, we had to apply an additional review and filtering.

Inclusion Criteria for Fraudulent Firms and Fraud Periods. We separated public from non-public companies by applying two main checks to each case: first, whether the company had a CIK; and second, if so, whether it had ever submitted a periodic filing required of public companies (10-K or 10-Q). Companies that had only submitted 10KSB (Deloitte Development LLC, 2008) forms before 2007 were not accepted as valid cases. If no 10-K or 10-Q was filed during the fraud period, the firm was not accepted as a valid case. For all valid cases, we recorded the CIK to link firm-years to corresponding labels and to enable integration with other datasets. As CIKs may vary over time, often due to a corporate split or restructuring, we use the CIK that corresponded to the periodic filings during the fraud period. Furthermore, we require the accounting enforcement action to specifically mention the firm as the perpetrator, meaning the firm had to be listed as a respondent and identified in the legal violations.

In rare cases, it was difficult to determine which specific firm was involved in an enforcement action due to ambiguous names and overlapping corporate structures. These ambiguities were resolved by examining filing patterns, such as joint submissions and matching CIKs in the 10-K headers. As mentioned, the fraud period often had to be deduced, and in some cases, the exact period could not be determined. Expecting the precise start and end dates of the fraud to be stated is often unrealistic and adds additional complexity to the requirements. Trading off precision against practicality, we approached the timeline in terms of quarters and fiscal years[8], ensuring our data also supports research involving 10-Q reports. We further distinguish between cases where the fraud period was mentioned explicitly and those where it was not.

Vague Cases. If the AAER stated that the fraud began at the "beginning" of a year, it was marked as starting in the first quarter. If it said "middle", the fraud began at the halfway point of the year. If it said "end", it began in the final quarter. If the AAER referred to a "fiscal year" rather than a calendar year, the same rules applied, but based on the company's fiscal quarters. If no specific timing was mentioned rather just the years where it had been committed, the fraud start was approximated as the beginning of the fiscal year, and the end was set to the end of that fiscal year. This approach allowed aligning fraud periods with the reporting periods in periodic filings. All such cases were classified as vague in terms of identifying the start and end dates.

Specific Cases. In some AAERs, specific quarters were mentioned as the fraud start or end period. In others, the AAER stated that the fraud began in the "period ending" a specific quarter – these entire quarters were marked as fraudulent. If an AAER said that the company "was fraudulently reporting" for a specific year or fiscal year, the entire fiscal year was marked as fraudulent. To distinguish these more precisely defined cases and also those mentioning a specific month as start and end date from vague ones, we included a certainty indicator in the data set (`certainty_start` and `certainty_end`). Cases with vague timing were marked with a 0, and specific cases with a 1. As with all binary variables in the label data set, 1 corresponds to the affirmative, and 0 to the negative. To ensure proper integration with the firm-year data set, the start and end dates of fraud were aligned with the **reporting** dates, not the **filing** dates. This decision supported the goal of aligning fraud periods with the timing of financial reports. In some rare cases, the AAERs mentioned more than one fraud period. If the periods did not overlap, two separate fraud cases were created – even if they originated from the same AAER. If the periods overlapped but involved distinct types of fraudulent activity,

[7] https://www.sec.gov/enforcement-litigation/accounting-auditing-enforcement-releases

[8] The fiscal year refers to the 12-month reporting cycle and may not align with the calendar year.

the case was likewise split into two cases based on the statutory violations cited in the AAER.

Violations. The SEC, as a regulatory body, derives most of its authority from the Securities Act of 1933 and the Securities Exchange Act of 1934.[9] It has the authority to enforce these acts and to take action against firms that violate them. The legal violations detailed in the AAERs are also included in our data set to provide information that may be useful for other purposes. In doing so, we do not record all violations mentioned in the AAERs, but only those specifically perpetrated by the firm in question – excluding those against individuals or unrelated entities named in the same AAER. Moreover, if a firm persistently fails to meet its periodic filing requirements, the SEC may revoke its Exchange Act registration under Section 12(j) of the Exchange Act (U.S. Congress, 1934). This is a rare occurrence and is recorded in the data set via the variable `revoked`, which notes the date (month and year) the registration was revoked. To enrich the data set further with a more structured understanding of fraud, the ACFE fraud classification introduced in Section 1 is also included. This categorization includes corruption, asset misappropriation, and financial statement fraud.

CIK Discrepancies. While merging the data sets, our earlier suspicion was confirmed: the CIK list provided on the SEC website is neither exhaustive nor immutable. The number of available CIKs varies depending on when the JSON file (company_tickers.json) is accessed. The CIK list used for crawling was obtained on May 8, 2025, selected solely because it contained the largest number of keys compared to the other lists at our disposal. This list includes 7,900 unique CIKs out of 10,132 total entries. In contrast, the labels data set contains 534 unique CIKs across 570 observations, 342 of which were not present in the CIK list used to crawl the firm-year data. This discrepancy increases by eight when comparing the CIKs in the firm-year data set with those in the labels data set, as the number of unique CIKs in the firm-year data set shrinks to 5,169 due to crawling issues. As a result, we decided to separately crawl firm years using the CIKs from the labels data set, yielding 10,764 firm-year observations. After merging both sources and removing duplicates, the final data set comprised 94,922 firm-year records. For the sake of reproducibility, the list of CIKs used to crawl the firm-year data set is included in the GitHub.

Data Merging. After generating firm-year records and compiling fraud labels, we were left with two distinct data sets: one containing 84,203 firm-years crawled using the SEC's CIK list, and another with 10,764 firm-years crawled using CIKs from the labels data set. Additionally, the labels data set contained 570 fraud-labeled observations. Merging these data sets was made significantly easier by relying on the Central Index Key (CIK), as merging based on company names would have required a fuzzy matching algorithm to resolve inconsistencies. The purpose of the labels data set was to add a binary fraud indicator to the firm-years, which could then be expanded with associated metadata variables. The `reporting_date` denotes the end of the fiscal year to which a report pertains, and it should correspond to the `fiscal_year_end` field included in the firm-year data set. A firm-year was labeled as fraudulent if its `reporting_date` fell within a fraud period listed in the labels data set. However, using this criterion alone would overlook cases in which fraudulent activity extended beyond the end of a fiscal year. To account for ongoing irregularities, any fiscal year in which the fraud period ended was also labeled as fraudulent. It is worth noting that a more precise labeling procedure could be achieved by merging the labels with a "firm-quarter" data set. To support future refinements, the individual raw data sets (the labels, firm-years from the CIK list, and firm-years from the labeled data set) are provided along with this thesis. The script used to merge these data sets is also included in the electronic appendix. In instances where a firm had overlapping or multiple fraud periods (as indicated by multiple entries in the labels data set), corresponding firm-year entries were duplicated, each reflecting distinct fraud-related metadata.

4 Descriptive Analysis

4.1 Full Data Set

The raw data set comprises $n = 89,453$ observations and $p = 57$ variables. A full description of all variables is provided in Table 2 in Appendix D. It includes filings from 5,508 unique CIKs, which

[9]The purpose of the Securities Act is primarily to ensure transparency and fairness before the initial issuance of securities, while the Exchange Act is more concerned with regulating the trading of securities in the secondary market.

approximately correspond to the number of distinct firms contained in the data set. Some CIKs have been dropped when extracting firm-years due to their integration into other filings, such as in cases where they are subsidiaries. These filings span 33 years, from 1992 to 2025, with an average of 16 and a median of 13 filings per company. It is important to note that the data set includes not only standard 10-K filings but also amended filings and late filings. As a result, some firms may appear overrepresented due to multiple amendments or corrections, a pattern that is common among the most frequently appearing ones. The most frequently occurring firm in the data set is *"Old Republic International Corporation"*, headquartered in Chicago, with a total of 81 filings (due to numerous amended reports).

Geographical Distribution. The data set contains firms located in 2,157 different cities, covering all 50 US states, D.C., three US territories (Guam, Puerto Rico, the Virgin Islands), Canadian provinces, and several other countries. In total, these firms span 158 unique jurisdictions, including both US states and international regions. In terms of legal incorporation, firms are registered across 91 different jurisdictions. Notably, in 61,881 instances, the state of incorporation differs from the state in which the firm is headquartered, with Delaware being by far the most common state of incorporation (cf. Figures 4a and 4b, Appendix E).

New York City has the highest number of filings and also hosts the greatest number of distinct firms. It accounts for more than twice the number of filings as the second-ranked city, Houston, and over three times as many unique firms (cf. Figures 5a and 5b, Appendix E). Interestingly, although Chicago and Atlanta are among the top cities by total filings, but not in terms of the number of distinct firms, suggesting a higher turnover of firms or a smaller presence of legacy corporations. At the state level, California surpasses New York in both total filings and the number of distinct firms (cf. Figures 6a and 6b, Appendix E). This is likely attributable to California's large population size, GDP, economic diversity, and concentration of large corporations spread across numerous cities. As previously mentioned, Delaware dominates as the primary state of incorporation, both in terms of the total number of filings and the number of incorporated firms.

Industry Distribution. The data set spans 412 distinct industries, classified by four-digit Standard Industrial Classification (SIC) codes. The most common industry by number of filings is *Pharmaceutical Preparations* (SIC 2834), with a total of 5,265 filings. This industry also ranks highest in terms of the number of distinct companies (cf. Figures 8a and 8b, Appendix E). It is important to note that the SEC does not always provide a SIC code in the header of each filing, not even the SIC code 9999, which represents a general-purpose category for firms that do not fit into any other industry. Filings with missing SIC codes span a wide variety of industries and account for 1405 filings, ranking 11th in group size. The smallest fraction of the data pertains to *Wholesale Trade - Furniture and Home Furnishings* (SIC 5020), with only 2 filings across 2 distinct companies (cf. Figure 8c, Appendix E). Aggregating the data by major industry groups reveals further insights. Consistent with the top specific industry, *Major Group 28* (*Chemicals and Allied Products*) ranks highest by both filings and number of companies. However, *Business Services* (Major Group 73) emerges as the second most common major group in both dimensions, indicating its significance across the corporate filing landscape. When aggregated even further to the division level, *Division D* (*Manufacturing*) dominates the data set in both total number of filings and companies. This division includes Major Group 28 and captures a wide range of manufacturing-related industries (cf. Figures 7a, 7b, 9a, and 9b, Appendix E).

Finally, several data quality issues were identified in several filings, such as a filing by *Coeur D'Alene Mines Corporation*, which operates in the *Gold and Silver Ores* industry. This filing was incorrectly tagged with SIC code 1044, which does not correspond to any valid industry classification. Four filings were listed with SIC Code 0, which does not exist, including two filings from *Enron Oil & Gas Company* which should correspond to SIC 1311, one from *BP Prudhoe Bay Royalty Trust* which should correspond to 2911 and one filing from *National Health Laboratories Holdings Inc.* which should correspond to sic 8071. These examples suggest the presence of additional misclassifications in the data, some of which may be undetectable unless the SIC code is invalid or missing.

Filing Types and MDAs. Among all filing types present in the data, the standard 10-K form is by far the most prevalent (77,851 filings), followed by the amended 10-K/A version (12,642 filings)[10] and other smaller, negligible categories (cf. Figure 10, Appendix E). Across all filings, the MDA sections have an average word count[11] of 8,340 and a median of 6,981, indicating a right-skewed distribution. Some outliers are substantially longer, with the longest MDA comprising 188,443 words. As shown in Figure 3, not only does the number of filings in the data set increase over time, but also the average and median word counts. This trend is likely driven by enhanced standardization, improved digitization of EDGAR filings, and the SEC's continued efforts to refine filing procedures and document formatting. This enhanced filing quality also allows the parser to work more effectively and extract MDA sections more reliably.

Another important consideration is the presence of filings that do not contain substantive content, e.g., only the word "omitted" in the MDA section, while others refer the reader to other sections or separate documents. After manually inspecting a sample of such cases, filings with MDAs below a threshold of 200 words were considered *non-substantive* and were excluded from the analysis.[12] After filtering, the data set containing the substantive MDAs comprises 68,894 MDAs. The distributions of word and character counts for these texts are visualized in Figure 13 (Appendix E).

4.2 Fraudulent Cases

Turning to the subject of fraud, approximately 2.9% of all filings in the data set are labeled as fraudulent. This corresponds to 2,598 fraudulent filings out of a total of 89,453 (cf. Table 3, Appendix F). After filtering out the illegitimate cases (as described above), the proportion of fraudulent filings slightly decreases to 2.3%, amounting to 1,596 fraudulent entries (cf. Table 4, Appendix F). The distribution of word counts in these legitimate fraudulent filings is illustrated in Figure 14 (Appendix E). The highest number of fraudulent filings occurred around the year 2001, peaking at 213 filings (cf. Figure 1). This spike coincides with the aftermath of the *dot-com bubble*, a period of excessive speculation in internet-related companies during the late 1990s. A second notable peak is observed around 2010, corresponding to the fallout of the *2008 global financial crisis*, which led to the failure of major financial institutions and exposed widespread corporate malfeasance. During this period, fraudulent activity again surged, with regulators uncovering misconduct across various industries. Following the 2010 peak, the number of detected fraud cases declined steadily.

Time Delay and Duration. This decline, however, is likely not indicative of an actual reduction in fraud, but rather reflects the inherent lag in detection and enforcement. Regulatory bodies such as the SEC typically take several years to investigate and build cases against firms. As such, any recent misconduct, such as potential fraudulent activity during the COVID-19 pandemic, may not yet be reflected in the data. This time lag between fraudulent activity and its detection is evident in the distribution of detection delays. As shown in Figures 11a and 11b (Appendix E), the average time from the start of fraud to its detection is approximately 6.5 years. Further, the average time from the end of the fraudulent activity to its public exposure is about 3.5 years. This delay arises not only because fraud is difficult to detect, but also due to the time-consuming nature of building a legal case. On average, fraudulent activity spans approximately three years, as illustrated in Figure 12 (Appendix E).

Fraud by Industry. When analyzing fraud across industries, the ones with the highest rates of fraudulent filings are not necessarily the most common. As illustrated in Figure 15 (Appendix E), SIC 7372 (*Services—Prepackaged Software*)ranks fourth in terms of overall filing frequency, yet it accounts for the highest numbers of fraudulent filings. This aligns with historical trends, as the *dot-com bubble* era (late 1990s) was characterized by significant corporate fraud in the technology and software sectors. The industry with the highest proportion of fraudulent filings relative to total filings is SIC 2020 (*Dairy Products*). While the absolute number of fraud cases in this industry is low, its relative fraud rate is the highest among all SIC codes. This highlights how less visible or prominent sectors can still exhibit high relative risk. The major group with the highest fraud proportion is *Major Group 51: Wholesale Trade—Nondurable Goods* (cf. Fig-

[10] Amended filings (10-K/A) typically result from the need to correct errors, clarify previously misstated information, or respond to regulatory or legal issues. Although not all amendments are associated with fraud, they often reflect irregularities in the original reports. For this reason, MDA sections from amended filings are excluded from the subsequent analysis.

[11] Word counts were computed after the following preprocessing steps: Lower-casing, removal of stopwords, URLs, HTML tags, extraneous whitespace or non-textual symbols.

[12] This results in two versions of the data set, one with all the observations and one with only the substantive MDAs.

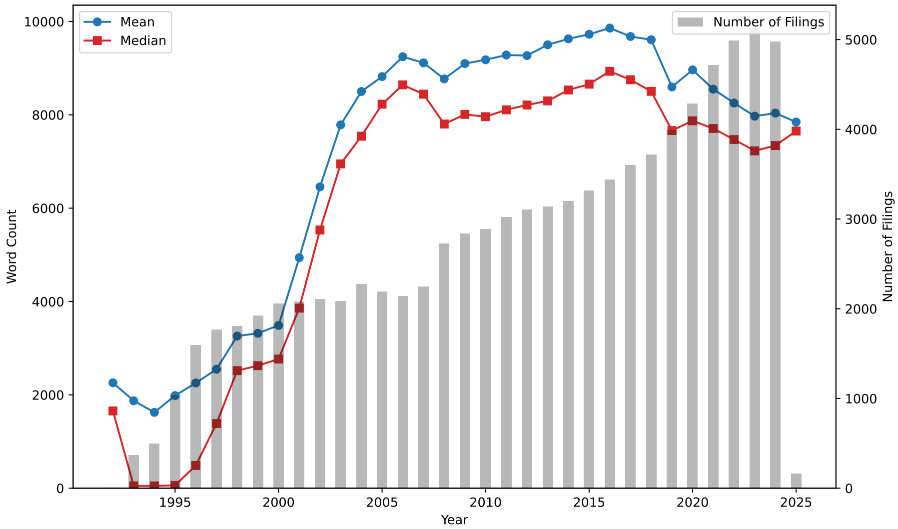


Figure 3: Mean and Median MDA Word Count Over Time with Filing Count

ure 16, Appendix E), while *Business Services* is the group with the highest absolute number of fraud cases, consistent with its high representation across the data set. At the division level, a similar discrepancy appears. *Division D (Manufacturing)* contains the most fraudulent cases in absolute terms[13], while the division with the highest fraud rate is *Division F (Wholesale Trade)*, which includes Major Group 51. These patterns suggest that while some sectors are more prone to frequent fraud due to their size, others exhibit disproportionately high risk relative to their footprint in the data.

Fraud Geography. When examining the geographical distribution of fraud in terms of absolute counts, the most fraudulent state is California, which aligns with its overall dominance in the number of total filings. However, when measured by fraud rate, Luxembourg stands out (4 out of 23 filings; cf. Figure 18, Appendix E). At the city level, the highest fraud rate is found in Pembroke, Bermuda, a notable offshore financial center,[14] while New York City leads in terms of absolute numbers, which aligns with the high density of financial institutions headquartered there – institutions that have historically been implicated in numerous financial misconduct cases. The most common state of incorporation for fraudulent firms is Delaware, which should not come as a surprise given that it is the most common state of incorporation overall. New Brunswick (Canada) exhibits the highest rate, with 5 out of 27 filings flagged as fraudulent (cf. Figure 20, Appendix E). Taken together, these findings provide a nuanced geographical portrait of corporate fraud in the data. While certain regions naturally have higher counts due to their economic prominence, some lesser-known jurisdictions display disproportionately high fraud rates. Nevertheless, these results must be interpreted with caution, as fraud overall remains a relatively rare event in the data, and high fraud rates in small regions are often based on few observations.

5 Experimental Results

Exemplarily, we use all data until 2008 for training and test on all data from the year 2011. We did abstain from using more current data due to delays in fraud detection and the comparably low amount of data in current years. Given that the median post-fraud detection delay is approximately 3.2 years, all models are tested on data that is at least three years in the future relative to the training data. A further reason for separating training and testing data temporally is that disclosures inherently have temporal characteristics. Extracting the MDA texts works increasingly well over time due to the enhanced standardization, and the rate of non-substantive fraudulent MDA sections is higher in earlier filings. We consider standard metrics such as accuracy, precision, recall, F1, and the AUC[15] to provide a point of reference – es-

[13]Excluding unknowns, which also might be an indicator of the inconsistencies in the fraudulent filings compiled.

[14]This ranking includes only cities with at least 10 filings, as smaller sample sizes prohibit robust conclusions.

[15]AUC does not adequately reflect performance in imbalanced settings; Our main evaluation metric is the F1-Macro.

Input Features	Random Forest					XGBoost					No Fraud / Fraud	
	Precision	Recall	F1-Macro	AUC	Accuracy	Precision	Recall	F1-Macro	AUC	Accuracy	Train Set	Test Set
Tabular	0.79	0.64	0.68	0.71	0.97	0.61	0.65	0.63	0.72	0.95	27226 / 1700	2935 / 86
Word Count	0.51	0.52	0.51	0.52	0.97	0.50	0.54	0.31	0.55	0.41	16695 / 937	2344 / 64
Sentence Embeddings	0.49	0.50	0.49	0.67	0.97	0.54	0.52	0.53	0.65	0.96	16695 / 937	2344 / 64
ModernBERT	0.49	0.50	0.49	0.62	0.97	0.50	0.50	0.50	0.61	0.96	16695 / 937	2344 / 64
29 LDA Topics	0.49	0.50	0.49	0.68	0.97	0.53	0.59	0.53	0.66	0.89	8032 / 307	2344 / 64
75 LDA Topics	0.49	0.50	0.49	0.64	0.97	0.53	0.55	0.54	0.64	0.94	8032 / 307	2344 / 64
100 LDA Topics	0.49	0.50	0.49	0.59	0.97	0.54	0.56	0.55	0.62	0.94	8032 / 307	2344 / 64

Table 1: Macro-Averaged Classification Metrics for Random Forest vs. XGBoost Across Input Representations. Differences in the sizes of the training sets result from the exclusion of the invalid cases, which are only present for the model trained on tabular data. For LDA, only 2004 to 2008 were used for training (due to the amount of data).

pecially for comparison with the work of (Bhattacharya and Mickovic, 2024). An overview of the results is presented in Table 1, where we compare Random Forest (Breiman, 2001) to XGBoost (Chen and Guestrin, 2016)) trained on different input features: tabular data only, word counts, sentence embeddings (`all-MiniLM-L6-v2`; Reimers and Gurevych, 2019), ModernBERT embeddings (`answerdotai/ModernBERT-base`; Warner et al., 2025), and LDA topics (inspired by Hoberg and Lewis, 2017). Overall, the performance across all evaluation metrics is worse than that achieved with the tabular data baseline, showing that classifiers based on only text-based inputs struggle severely. This impression is further supported by the full classification reports in Table 5 (Appendix G).

6 Conclusion

The Achilles heel of any fraud analysis is the scarcity of fraud cases, which significantly hampers efforts to gain meaningful insights. Therefore, it is essential not only to aggregate extensive, high-quality data but also to involve the right competence regarding analytical methods and domain knowledge. We see our work as an important auxiliary means for providing domain experts with high-quality data. Innovative analytical methods are vital as tools to assist experts in combating fraud and broader financial crime effectively. Hence, using the latest technology, such as large language models, is an important step in advancing this endeavor. Fraud detection is a balancing act, often involving inherently imbalanced data where invasive monitoring is difficult without compelling evidence, making false positive claims is particularly costly, and misallocating resources due to incorrect fraud predictions can negatively affect firms, employees, and the fraud detection process itself. Ultimately, this work aims to provide valuable insights and especially resources to anyone interested in understanding and analyzing the complexities of fraud detection.

Limitations

Despite we hope to have provided a valuable resource for fellow researchers working in the area of financial natural language processing, we are aware that our data set does not come without shortcomings: **(1)** As already mentioned a few times throughout the main part of this paper, we can not be entirely sure about the correctness of all CIKs; while some errors can be (relatively) easily be detected, further error correction would require domain knowledge beyond our expertise. **(2)** Given that only *detected* cases of fraud (positives) can be labeled as such, this positions this data set in the realm of positive-unlabeled (PU) learning, as we can not be sure whether negatives (cases labeled as *no fraud*) are actually not fraudulent or whether they were just not detected. This shortcoming has neither been addressed in previous work nor is it reflected in our baseline results; we leave the application of PU learning techniques to this data for future work.

The exclusive reliance on SEC enforcement reports raises another concern regarding common source bias. Both the texts and the metadata, meaning all inputs, are collected from a single source and have not been supplemented with data from elsewhere. Moreover, since none of us has had direct contact with the SEC, we cannot assess the systematic deficiencies over time that may have influenced the data available on EDGAR and the SEC website. Nevertheless, these limitations still deserve to be acknowledged.

Acknowledgments

Matthias Aßenmacher received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research

Data Infrastructure – NFDI 27/1 - 460037581 - BERD@NFDI. This work was partially supported by Henan Provincial Center for Outstanding Overseas Scientists (No. GZS2025004).

References

Association of Certified Fraud Examiners. 2024. 2024 report to the nations: Global study on occupational fraud and abuse. Technical report, Association of Certified Fraud Examiners.

Indranil Bhattacharya and Ana Mickovic. 2024. Accounting fraud detection using contextual language learning. *International Journal of Accounting Information Systems*, 53:100682.

Leo Breiman. 2001. Random forests. *Machine learning*, 45:5–32.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. ACM.

Deloitte Development LLC. 2008. Changeover to the sec's new smaller reporting company system. Accessed: 2025-05-18.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Art Durnev and Claudine Mangen. 2020. The spillover effects of md&a disclosures for real investment: The role of industry competition. *Journal of Accounting and Economics*, 70(1):101299.

Ronen Feldman, Suresh Govindaraj, Joshua Livnat, and Benjamin Segal. 2010. Management's tone change, post earnings announcement drift and accruals. *Review of Accounting Studies*, 15(4):915–953.

Gerard Hoberg and Craig Lewis. 2017. Do fraudulent firms produce abnormal disclosure? *Journal of Corporate Finance*, 43:58–85.

Lori Holder-Webb and Jaffrey R. Cohen. 2007. The association between disclosure, distress, and failure. *Journal of Business Ethics*, 75(3):301–314.

Nasdaq Verafin. 2024. 2024 global financial crime report. Technical report, Nasdaq Verafin. Accessed: 2025-03-11.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

SEC API. 2025. Sec edgar api service. Accessed: 2025-05-18.

University of Southern California. 2021. Aaer dataset. Accessed: 2025-05-18.

U.S. Congress. 1933. Securities act of 1933. https://www.govinfo.gov/content/pkg/COMPS-1884/pdf/COMPS-1884.pdf. Codified at 15 U.S.C. §§ 77a-77aa. Accessed: 2025-05-18.

U.S. Congress. 1934. Securities exchange act of 1934. https://www.govinfo.gov/content/pkg/COMPS-1885/pdf/COMPS-1885.pdf. Codified at 15 U.S.C. §§ 78a-78qq. Accessed: 2025-05-18.

U.S. Securities and Exchange Commission. 2023. Privacy and security. Accessed: 2025-05-17.

U.S. Securities and Exchange Commission. 2024. Apple inc. form 10-k filing (2024). Accessed: 2025-05-17.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Appendix

A Format of the 10-K filings

Information in the 10-K headers (highlighted in red, Figure 2), in the following order of appearance:

- **Conformed Period of Report** - Report period
- **Filed as of Date** - Date the report was filed
- **Company Conformed Name**
- **Central Index Key** - Unique identifier
- **Standard Industrial Classification (SIC)**
- **State of Incorporation**
- **Fiscal Year End** - format: mmdd
- **Form Type** - Type of filing (cf. Appendix B)
- **City** - City of the company
- **State** - State/foreign country of the company

B Further 10-K Variants

The different types of annual reports are as follows:

- **10-K** - Standard annual report
- **10-K/A** - Amended annual report
- **10-K405** - Late annual report
- **10-K405/A** - Amended late annual report
- **10-KT**[16] - Transition annual report
- **10-KT/A** - Amended transition annual report

[16] Transitional reports are filed in cases where, e.g., firms merge and the first report after the merger is submitted outside the required reporting period of one of the merging firms.

C Regex Pattern for MDA-Extraction

Item 7 Start Pattern

- `r"it[\s]*em[\s]*7[\.\s]*manag[\s]*e?[\s]*ment[\s\'-]*[\w\s\'-]{0,10}(discussion[\s]*and[\s]*analysis|narrative[\s]*analysis)"`
  Matches various flexible formats of the section header for Item 7, it allows for:
  - Optional whitespace or punctuation between characters (e.g., "Item 7." or "Item 7 Management's")
  - Variations like "management" or misspelled/malformed variants
  - Up to 10 characters between "management" and "discussion" to tolerate OCR noise/alternative phrasings
  - Matches either "discussion and analysis" or "narrative analysis"

Search Phrases

- `the following discussion`
- `this discussion and analysis`
- `should be read in conjunction`
- `should be read together with`
- `r"the following management[\s\'-]*s discussion and analysis"`
  Flexible punctuation or possessive formatting.
- `r"(?:\"[^\"]+?\")([^"]*\"[^\"]+?\"){4,}"`
  A complex pattern that identifies sequences of four or more quoted strings.
- `r"\b(?:\w+)(?:\s*,\s*\w+){3,}\b"`
  Matches sequences of at least four comma-separated words.

Item 8 End Pattern

- `r"item[\s]*8[\.\s]*financial statements and supplementary data"`
  Allows optional whitespace or punctuation between "Item 8" and the section title.

D Variable Overview

Table 2: Variable Descriptions in the MDA-Fraud Dataset

Variable	Description
Firm-Years: Variables Scraped & Parsed from Annual Reports (10-K)	
`cik`	Central Index Key (unique company identifier)
`name`	Company name
`city`	City
`state`	State
`sic`	Standard Industry Classification number
`incorp_state`	State of incorporation
`filing_type`	Filing type
`fye`	Fiscal year end
`filing_date`	Date the 10-K was filed
`reporting_date`	Period the 10-K reports on
`url`	URL to the filing
`mda`	Management Discussion and Analysis section (text)
`late_filing`	Indicates a 10-K405 (late filing)
`transition_filing`	Indicates a 10-KT (transition report)
`amend_filing`	Indicates amended 10-K (any 10-K ending in /A)
Labels: Variables from Labels Dataset	
`dateTime`	Date and time of the AAER
`respondents`	Names of respondents in the AAER
`fraud_start`	Beginning date of the fraudulent period (mm-yyyy)
`fraud_end`	End date of the fraudulent period (mm-yyyy)
`revoked`	Revokation date of Exchange Act registration (mm-yyyy)
`certainty_start`	Binary indicator: certainty regarding fraud start date
`certainty_end`	Binary indicator: certainty regarding fraud end date
`17a`	17(a) Securities Act violation
`17a2`	17(a)(2) Securities Act violation
`17a3`	17(a)(3) Securities Act violation
`17b`	17(b) Securities Act violation
`5a`	5(a) Securities Act violation
`5b1`	5(b)(1) Securities Act violation
`5c`	5(c) Securities Act violation
`10b`	10(b) Securities Exchange Act violation
`13a`	13(a) Securities Exchange Act violation
`12b20`	Section 12b rule 12b-20 Securities Exchange Act violation
`12b25`	Section 12b rule 12b-25 Securities Exchange Act violation
`13a1`	Section 13a rule 13a-1 Securities Exchange Act violation
`13a10`	Section 13a rule 13a-10 Securities Exchange Act violation
`13a11`	Section 13a rule 13a-11 Securities Exchange Act violation
`13a13`	Section 13a rule 13a-13 Securities Exchange Act violation
`13a14`	Section 13a rule 13a-14 Securities Exchange Act violation
`13a15`	Section 13a rule 13a-15 Securities Exchange Act violation
`13a16`	Section 13a rule 13a-16 Securities Exchange Act violation
`13b2A`	13(b)(2)(A) Securities Exchange Act violation
`13b2B`	13(b)(2)(B) Securities Exchange Act violation
`13b5`	13(b)(5) Securities Exchange Act violation
`14a`	14(a) Securities Exchange Act violation
`14c`	14(c) Securities Exchange Act violation
`30A`	Foreign Corrupt Practices Act violation

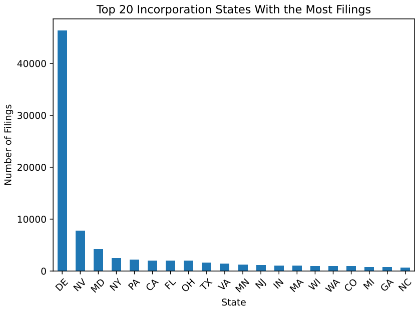
(Continued on next page)

Variable	Description
100a2	100(a)(2) Regulation G of Securities Act violation
100b	100(b) Regulation G of Securities Act violation
19a	19(a) violation under Investment Company Act
105c7B	105(c)(7)(B) violation under SOX
corruption	Binary indicator of corruption
amis	Binary indicator of asset misappropriation
fsf	Binary indicator of financial statement fraud
fraudulent	Binary indicator of fraud
MDA counts	
char_count	Character count of the MDA text
word_count	Word count of the MDA text
word_density	Number of characters per word

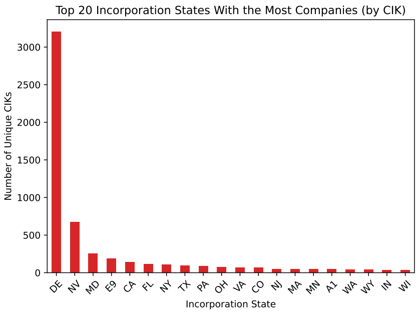
E Descriptive Analysis – Figures and Tables

E.1 Whole Data Set

Geographical Distribution

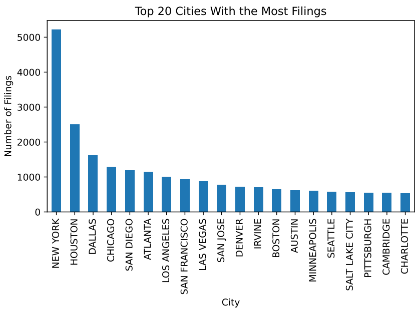


(a) Top 20 States of Incorporation by Filings

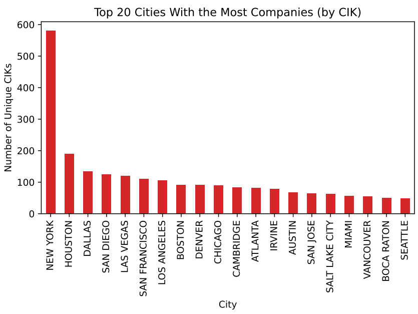


(b) Top 20 States of Incorporation by Number of Companies

Figure 4: Top 20 States of Incorporation

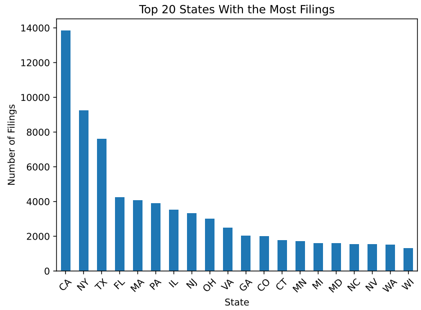


(a) Top 20 Cities by Total Filings

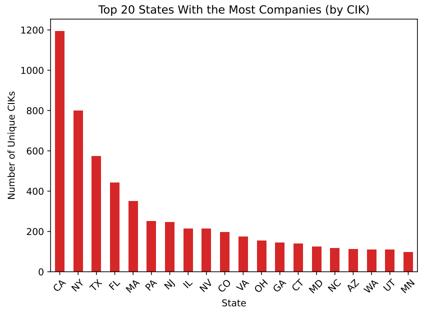


(b) Top 20 Cities by Number of Companies

Figure 5: Top 20 Cities



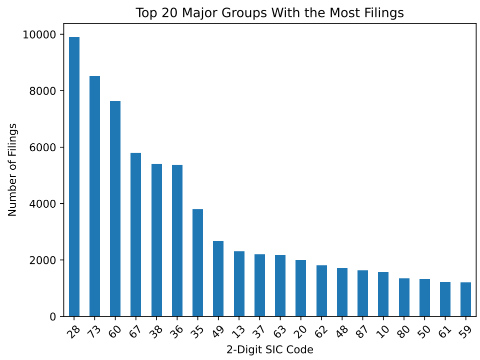
(a) Top 20 States by Total Filings



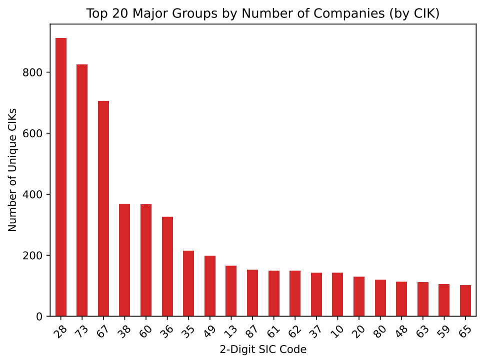
(b) Top 20 States by Number of Companies

Figure 6: Top 20 States

Industry Distribution

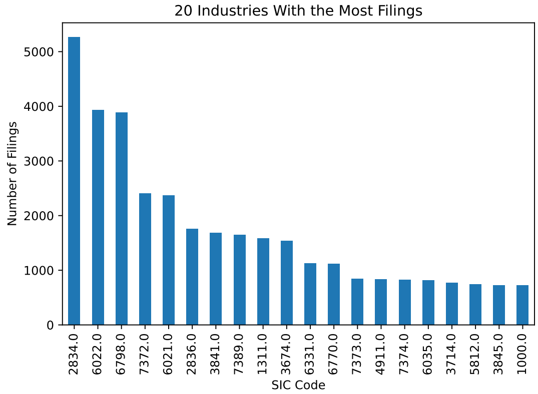


(a) Top 20 Major Groups by Filings

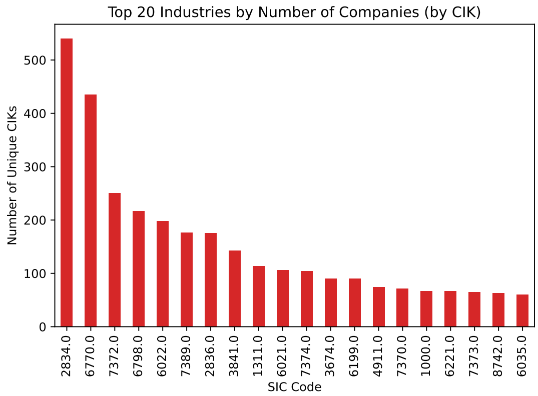


(b) Top 20 Major Groups by Number of Companies

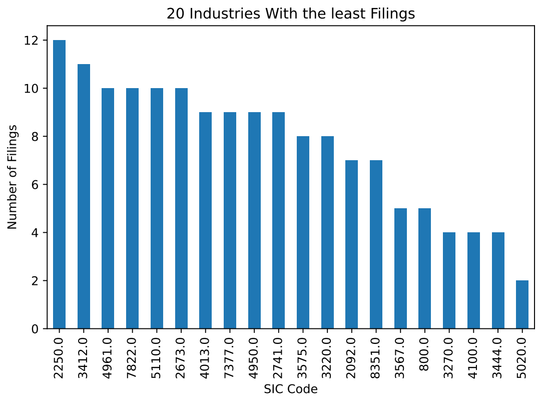
Figure 7: Top 20 Major Groups



(a) Top 20 Industries by Filings

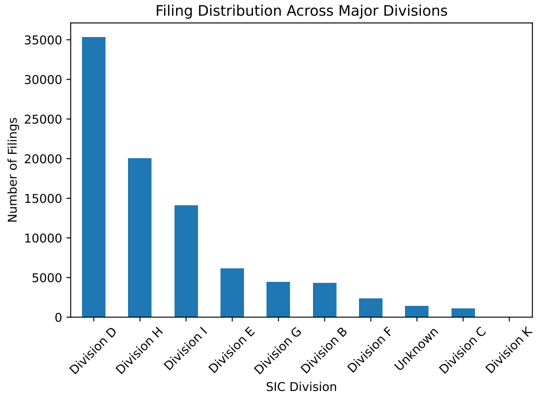


(b) Top 20 Industries by Number of Companies

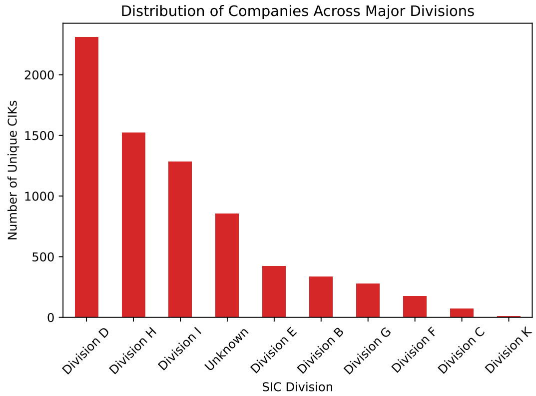


(c) Top 20 Industries by Number of Companies

Figure 8: Top and bottom 20 industries by the frequency of SIC codes in the data set.



(a) Top 20 Divisions by Filings



(b) Top 20 Divisions by Number of Companies

Figure 9: Top Major Divisions

Distribution of Filings

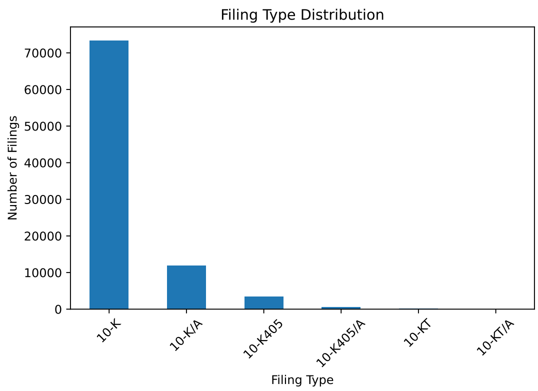
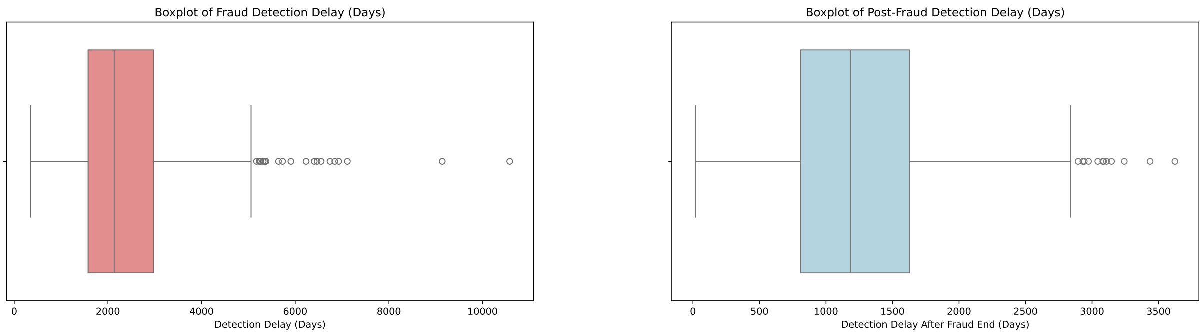


Figure 10: Distribution of 10-K Filing Types

Time Distribution



(a) Boxplot of Fraud Detection Delay

(b) Boxplot of Post-Fraud Detection Delay

Figure 11: Fraud Detection and Post-Fraud Detection Delays

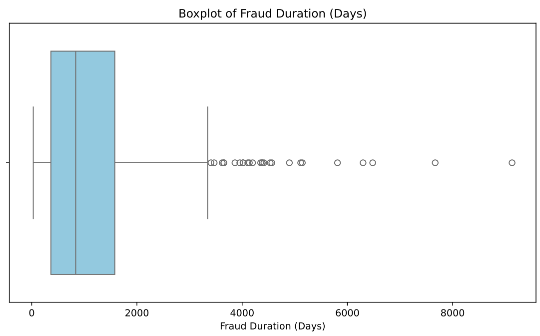


Figure 12: Boxplot of Fraud Duration

E.2 Text Length Distributions

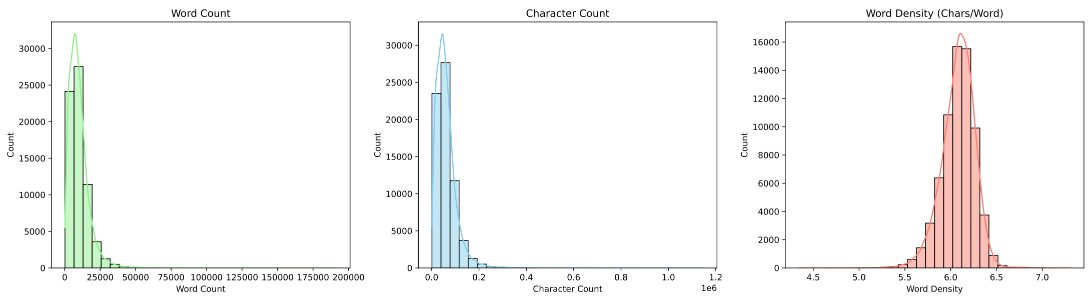


Figure 13: Text Length Distributions of Substantive MDA Sections

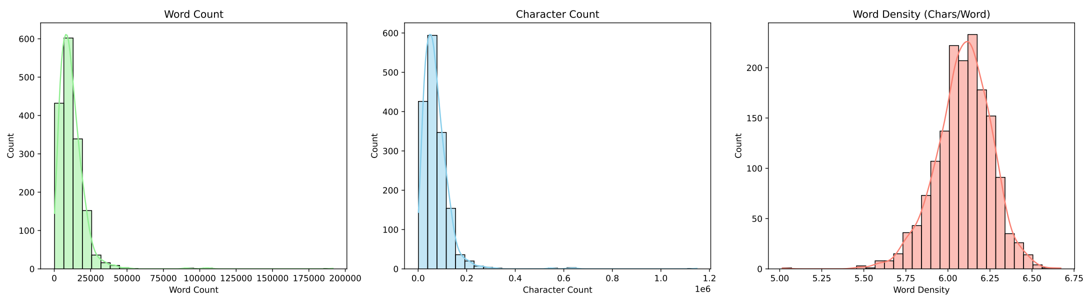
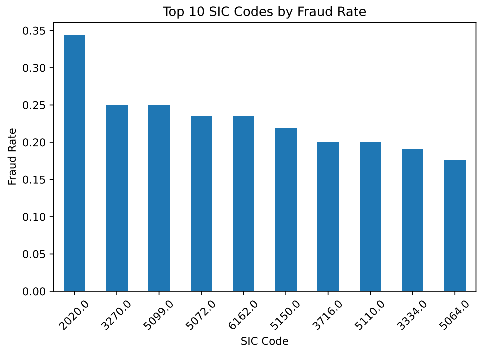
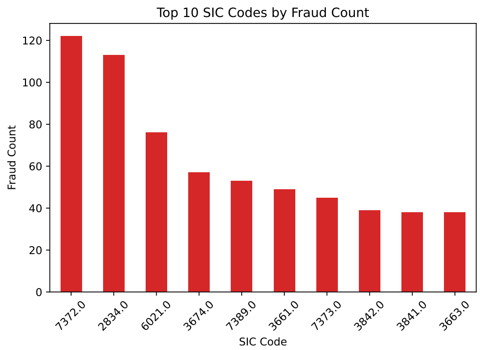


Figure 14: Text Length Distributions of Substantive Fraudulent MDA Sections

E.3 Fraud

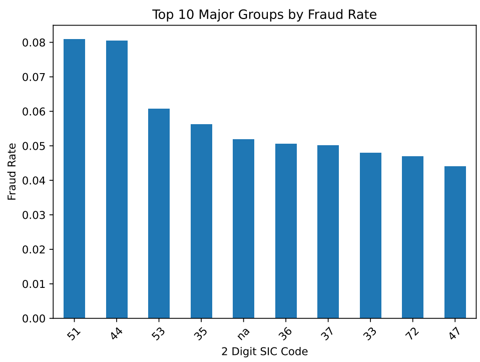


(a) Top 10 SIC Codes by Fraud Rate

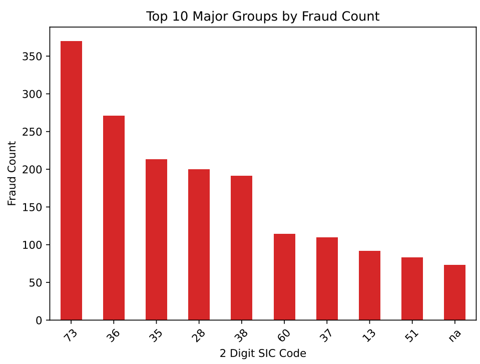


(b) Top 10 SIC Codes by Fraud Count

Figure 15: Top Fraud Industries

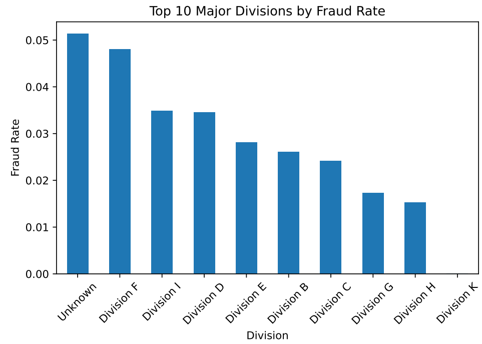


(a) Top 10 Major Groups by Fraud Rate

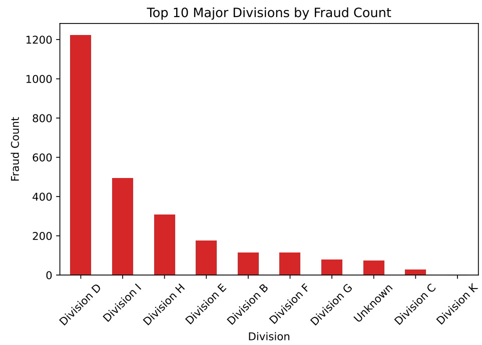


(b) Top 10 Major Groups by Fraud Count

Figure 16: Top Fraud Major Groups

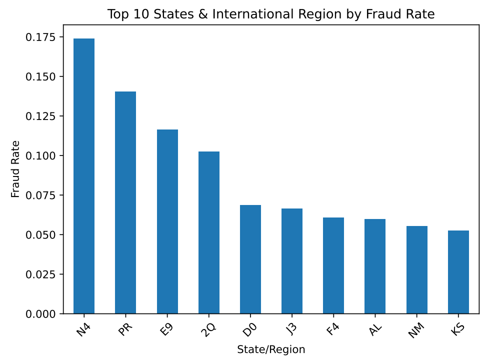


(a) Top 10 Major Divisions by Fraud Rate

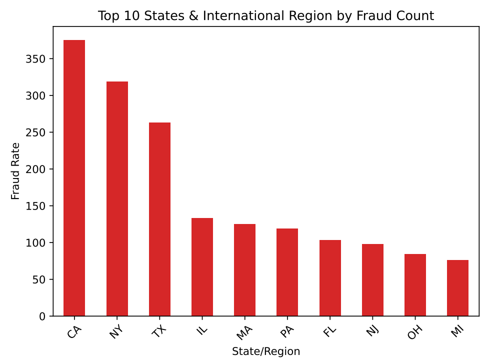


(b) Top 10 Major Divisions by Fraud Count

Figure 17: Top Fraud Major Divisions

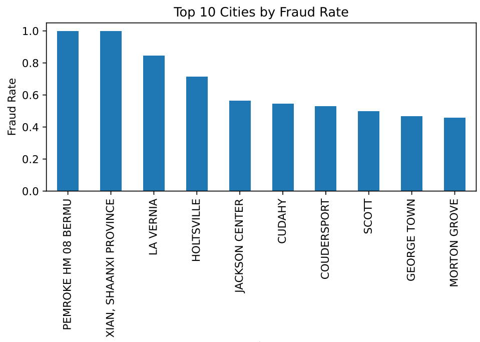


(a) Top 10 States by Fraud Rate

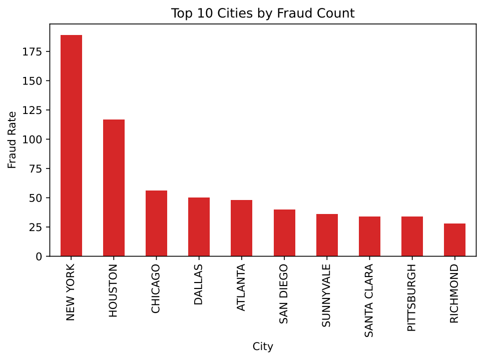


(b) Top 10 States by Fraud Count

Figure 18: Top Fraud US States and International Regions



(a) Top 10 Cities by Fraud Rate



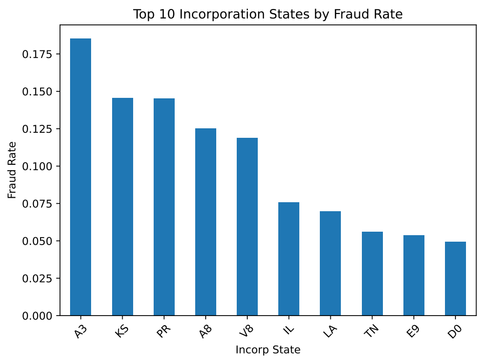
(b) Top 10 Cities by Fraud Count

Figure 19: Top Fraud Cities

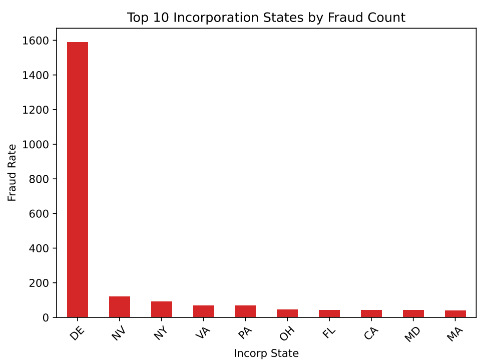
F Data Set Statistics

Year	Fraudulent Count	Filings Count	Fraudulent Fraction
1992	0	6	0.000
1993	8	370	0.022
1994	15	496	0.030
1995	24	1039	0.023
1996	60	1597	0.038
1997	101	1771	0.057
1998	120	1806	0.066
1999	162	1926	0.084
2000	204	2058	0.099
2001	213	2083	0.102
2002	185	2107	0.088
2003	140	2085	0.067
2004	128	2277	0.056
2005	102	2190	0.047
2006	83	2144	0.039
2007	71	2245	0.032
2008	84	2726	0.031
2009	97	2840	0.034
2010	107	2886	0.037
2011	86	3021	0.028
2012	84	3108	0.027
2013	64	3140	0.020
2014	68	3203	0.021
2015	72	3319	0.022
2016	74	3440	0.022
2017	70	3603	0.019
2018	63	3720	0.017
2019	51	3990	0.013
2020	27	4285	0.006
2021	22	4714	0.005
2022	11	4991	0.002
2023	2	5127	0.000
2024	0	4978	0.000
2025	0	162	0.000

Table 3: Fraudulent Cases by Year (Full Dataset)



(a) Top 10 Incorp States by Fraud Rate



(b) Top 10 Incorp States by Fraud Count

Figure 20: Top Fraud States of Incorporation

Year	Fraudulent Count	Filings Count	Fraudulent Fraction
1993	0	113	0.000
1994	3	147	0.020
1995	9	410	0.022
1996	27	730	0.037
1997	47	849	0.055
1998	59	955	0.062
1999	79	1053	0.075
2000	99	1171	0.085
2001	114	1230	0.093
2002	105	1281	0.082
2003	88	1354	0.065
2004	72	1446	0.050
2005	70	1558	0.045
2006	56	1630	0.034
2007	52	1694	0.031
2008	57	2011	0.028
2009	61	2171	0.028
2010	72	2277	0.032
2011	64	2408	0.027
2012	65	2478	0.026
2013	49	2576	0.019
2014	52	2672	0.019
2015	49	2825	0.017
2016	59	2977	0.020
2017	53	3083	0.017
2018	46	3210	0.014
2019	39	3392	0.011
2020	21	3644	0.006
2021	17	4152	0.004
2022	10	4364	0.002
2023	2	4432	0.000
2024	0	4444	0.000
2025	0	157	0.000

Table 4: Fraudulent Cases by Year (Filtered Dataset)

G Full Results

Input Type	Class	Random Forest					XGBoost					Support
		Precision	Recall	F1-Score	AUC	Accuracy	Precision	Recall	F1-Score	AUC	Accuracy	
Tabular	0 (Non-Fraud)	0.98	0.99	0.99	–	–	0.98	0.97	0.98	–	–	2935
	1 (Fraud)	0.60	0.28	0.38	–	–	0.25	0.33	0.28	–	–	86
		–	–	–	0.71	0.97	–	–	–	0.72	0.95	3021
	Macro Avg	0.79	0.64	0.68	–	–	0.61	0.65	0.63	–	–	–
	Weighted Avg	0.97	0.97	0.97	–	–	0.96	0.95	0.96	–	–	–
Word Count	0 (Non-Fraud)	0.97	0.92	0.95	–	–	0.98	0.40	0.57	–	–	2344
	1 (Fraud)	0.04	0.12	0.06	–	–	0.03	0.69	0.06	–	–	64
		–	–	–	0.52	0.97	–	–	–	0.55	0.41	2408
	Macro Avg	0.51	0.52	0.51	–	–	0.50	0.54	0.31	–	–	–
	Weighted Avg	0.95	0.90	0.93	–	–	0.95	0.41	0.55	–	–	–
Sentence Embeddings	0 (Non-Fraud)	0.97	1.00	0.99	–	–	0.97	0.99	0.98	–	–	2344
	1 (Fraud)	0.00	0.00	0.00	–	–	0.11	0.06	0.08	–	–	64
		–	–	–	0.67	0.97	–	–	–	0.65	0.96	2408
	Macro Avg	0.49	0.50	0.49	–	–	0.54	0.52	0.53	–	–	–
	Weighted Avg	0.95	0.97	0.96	–	–	0.95	0.96	0.96	–	–	–
ModernBERT	0 (Non-Fraud)	0.97	1.00	0.99	–	–	0.97	0.99	0.98	–	–	2344
	1 (Fraud)	0.00	0.00	0.00	–	–	0.03	0.02	0.02	–	–	64
		–	–	–	0.62	0.97	–	–	–	0.61	0.96	2408
	Macro Avg	0.49	0.50	0.49	–	–	0.50	0.50	0.50	–	–	–
	Weighted Avg	0.95	0.97	0.96	–	–	0.95	0.96	0.95	–	–	–
29 LDA Topics	0 (Non-Fraud)	0.97	1.00	0.99	–	–	0.98	0.90	0.94	–	–	2344
	1 (Fraud)	0.00	0.00	0.00	–	–	0.07	0.28	0.12	–	–	64
		–	–	–	0.68	0.97	–	–	–	0.66	0.89	2408
	Macro Avg	0.49	0.50	0.49	–	–	0.53	0.59	0.53	–	–	–
	Weighted Avg	0.95	0.97	0.96	–	–	0.95	0.89	0.92	–	–	–
75 LDA Topics	0 (Non-Fraud)	0.97	1.00	0.99	–	–	0.98	0.97	0.97	–	–	2344
	1 (Fraud)	0.00	0.00	0.00	–	–	0.09	0.12	0.10	–	–	64
		–	–	–	0.64	0.97	–	–	–	0.64	0.94	2408
	Macro Avg	0.49	0.50	0.49	–	–	0.53	0.55	0.54	–	–	–
	Weighted Avg	0.95	0.97	0.96	–	–	0.95	0.94	0.95	–	–	–
100 LDA Topics	0 (Non-Fraud)	0.97	1.00	0.99	–	–	0.98	0.96	0.97	–	–	2344
	1 (Fraud)	0.00	0.00	0.00	–	–	0.11	0.16	0.13	–	–	64
		–	–	–	0.59	0.97	–	–	–	0.62	0.94	2408
	Macro Avg	0.49	0.50	0.49	–	–	0.54	0.56	0.55	–	–	–
	Weighted Avg	0.95	0.97	0.96	–	–	0.95	0.94	0.95	–	–	–

Table 5: Results Comparison: Random Forest vs XGBoost across Different Input Representations

Synthesizing Behaviorally-Grounded Reasoning Chains: A Data-Generation Framework for Personal Finance LLMs

Akhil Theerthala
Perfios Software Solutions

Abstract

Personalized financial advice requires consideration of user goals, constraints, risk tolerance, and jurisdiction. Prior LLM work has focused on support systems for investors and financial planners. Simultaneously, numerous recent studies examine broader personal finance tasks, including budgeting, debt management, retirement, and estate planning, through agentic pipelines that incur high maintenance costs, yielding less than 25% of their expected financial returns. In this study, we introduce a novel and reproducible framework that integrates relevant financial context with behavioral finance studies to construct supervision data for end-to-end advisors. Using this framework, we create a 19k sample reasoning dataset and conduct a comprehensive fine-tuning of the Qwen-3-8B model on the dataset. Through a held-out test split and a blind LLM-jury study, we demonstrate that through careful data curation and behavioral integration, our 8B model achieves performance comparable to significantly larger baselines (14-32B parameters) across factual accuracy, fluency, and personalization metrics while incurring 80% lower costs than the larger counterparts.
Keywords: *Financial Datasets; Personal Finance; Reasoning Models; Large Language Models*

1 Introduction

Legal counseling, healthcare, and finance are among the numerous high-stakes domains in which personalized advice is essential. However, the development of this personalized advice is fraught with obstacles, requiring substantial investments and years of human expertise. Recent research efforts have thoroughly investigated automated decision support systems in various areas, emphasizing their cost-effectiveness. In the financial sector, a variety of support systems have been investigated, with a particular emphasis on asset recommendations and investment predictions. (Sanz-Cruzado et al., 2024; Luo et al., 2025; Takayanagi et al., 2023)

Recent advances in large language models (LLMs) have shown effective performance in acting as decision support systems for investors (Gupta, 2023) and financial planners (Huang et al., 2024). The core advantage of natural language generation presents these automated support systems with a unique advantage that was never available in previous applications. This advantage has repeatedly shown its power in linguistic tasks such as streamlining complex financial narratives from extensive documents, corporate discourses, news sources, and social media.(Gueta et al., 2025; Lee and Lay-Ki, 2024) The utility of these models is also being explored in Time series (Liu and Jia, 2025) and Financial reasoning applications (Liu et al., 2025).

Notwithstanding this capability, recent research indicates that no model excels across all financial task categories, which include text summarization, sentiment analysis, causal analysis, forecasting, and text classification (Matlin et al., 2025). It has been demonstrated that attaining robust performance frequently necessitates the utilization of large, expensive models, thereby constraining the practicality of these solutions. Due to these inherent limitations and the complexity of financial advisory, many studies focusing on broader financial decision systems have preferred an agentic approach over training financial domain-specific language models. (Okpala et al., 2025; Joshi, 2025; Takayanagi et al., 2025a)

Although the initial agentic frameworks focused on answering simple inquiries,(Lakkaraju et al., 2023) recent studies have accelerated the development of these systems to provide practical and actionable advice to the end user (Takayanagi et al., 2025b; Okpala et al., 2025). These agents can now dynamically interact with users and can assist in various tasks such as recommendation, question an-

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

swering, search, and customer profiling. (Li et al., 2024; Takayanagi et al., 2025a; Han et al., 2024)

Although agentic systems demonstrate potential in providing tailored financial advice, their efficacy is hindered by considerable constraints, including the integration with legacy systems, compliance with data security regulations, and high inference costs. (Cemri et al., 2025; Wang et al., 2025). In support of these concerns, a recent study by (Meimandi et al., 2025) illustrates that a confluence of technical and cost-related factors hinders these applications from realizing even 25% of their anticipated returns. This research also establishes an important differentiation: success in benchmarks does not necessarily equate to success in deployment. In practical terms, these proactive financial advisors frequently encounter a swift deterioration in performance within a matter of months following their implementation, attributable to the inherent volatility of real-world conditions. Concurrently, studies show that the extent of personalization is often limited by the volume of context and information that can be supplied to an agent, impacting the overall performance. (Zhou et al., 2025; Winder et al., 2024)

One of the direct ways to address these limitations is to tune a model with a domain-specific context that integrates financial, behavioral, and psychological information. This work aims to close this gap by providing a reproducible framework to generate financial advice through a well-structured chain-of-thought. In particular, the framework constructs supervision data to train models to (a) provide personalized guidance for users' financial dilemmas, (b) reliably apply core financial knowledge, and (c) recognize and mitigate user-side behavioral biases by integrating behavioral and historical evidence.

To address these limitations, we propose a novel, data-centric framework for synthesising behaviorally-grounded reasoning chains. Rather than relying on complex agentic architectures, our approach directly bakes financial, behavioural, and psychological knowledge into the training data itself. Crucially, we treat the inference of the user's psychological state not as an afterthought, but as a standalone, foundational phase in the reasoning chain. This design choice is directly motivated by recent findings that users' trust and engagement are heavily influenced by the persona of the advisor (Takayanagi et al., 2025a), not just the raw accuracy of its advice. By isolating and explicitly modelling this psychological dimension, our framework ensures that personalisation and empathetic framing are intrinsic to the model's reasoning process, leading to more effective and trustworthy financial guidance.

It should also be considered that although recent agentic frameworks respond based on real-time knowledge; most of these knowledge sources need to be manually curated (Aggarwal and Singh, 2024). In addition to this, we should note that most of the recommendations needed for general financial advice do not require real-time financial knowledge. Instead, this advice needs an agent that can inherently retrieve the relevant information from its memory. We address this problem by carefully crafting a chain-of-thought section to retrieve the financial context relevant to the query.

Recent studies have shown that inherent biases often limit users' ability to make many wealth-making financial decisions. (Baker et al., 2017; Agrawal, 2012) These biases are highly variable and often depend on the age, experience and location of the user. Many financial agents do not directly address these biases when providing financial advice to the user. In this study, we have tried to integrate these biases into the reasoning model's natural chain-of-thought to tune the final responses towards acknowledging and addressing these biases.

Each stage of chain-of-thought generation is verified by a set of Large Language Model juries that rank various generations and pick the best version suitable for the user queries. We used this framework to generate a 19k sample dataset, which is used to finetune a Qwen-3-8B model. This model is then compared to models of similar sizes to determine the impact of this framework.

This paper introduces a principled, data-centric framework as a step toward smaller, more trustworthy personal finance LLMs, and we outline its use as a backbone policy within agentic workflows to thin planning chains and lower orchestration cost—an evaluation we defer to future work.

2 Related Works

The application of automated systems to financial advice is not a new undertaking. Prior to the widespread adoption of large language models, research focused on applying classic techniques such as collaborative filtering and case-based reasoning to well-defined domains such as loan and insurance policy recommendation, as surveyed by Zibriczky (2016). However, the advent of powerful LLMs has opened new frontiers and presented a distinct set of challenges and approaches.

Much of the recent literature has focused on benchmarking the capabilities of general-purpose LLMs on a range of isolated financial tasks. For instance, a comprehensive study by Hean et al. (2025) evaluated leading models such as ChatGPT and Claude against standardized financial literacy questionnaires covering diverse topics from mortgages to taxes. While their findings show that newer models are consistently improving and can achieve high accuracy on specific topics, they also reveal significant limitations, concluding that LLMs still struggle to provide accurate responses for complex financial queries. This highlights a critical performance gap: off-the-shelf models are often insufficient for the nuanced demands of holistic financial advice.

To overcome the limitations of single models and address more complex, multi-step planning, a significant body of research has shifted towards developing sophisticated agentic workflows. A recent survey by Ding et al. (2024) provides a comprehensive overview of this landscape, categorizing these systems into distinct architectural patterns such as reflection-driven and debate-driven agents. A clear example is the work of Okpala et al. (2025), who designed "agentic crews" composed of multiple specialized LLM agents, such as data scientists and compliance checkers, to automate the entire financial modelling and risk management pipeline. While powerful, such multi-agent systems demonstrate significant architectural complexity and high maintenance costs. Furthermore, research into these conversational agents has revealed significant risks; Takayanagi et al. (2025a) found in a user study that participants often placed more trust in a confident, "extroverted" agent even when it provided lower-quality advice, highlighting the potential for these complex systems to mislead inexpert users.

We argue, however, that the primary bottleneck is not architectural complexity, but the inherent irrationality of the models themselves, necessitating a data-centric approach. This need is rooted in the tendency of LLMs to amplify human cognitive biases. The groundbreaking work of Zhou et al. (2025) introduced a comprehensive framework based on behavioral finance to demonstrate that LLMs exhibit significant financial biases, such as anchoring and overconfidence. Their crucial finding that fine-tuning on financial data can sometimes exacerbate these irrational tendencies underscores the profound risks of using uncurated data. This is supported by empirical studies exposing a significant "product bias" in leading LLMs (Zhi et al., 2025) and by findings that LLM-generated advice systematically increases portfolio risk by reinforcing investment biases such as geographical concentration and trend chasing (Winder et al., 2024). Taken together, these findings reveal that a model's pre-trained knowledge is an unreliable and potentially risky foundation for financial advice.

Therefore, our work addresses a critical gap. While large-scale financial language models like FinGPT, which continuously ingest real-time market data to update and adapt the underlying model (Yang et al., 2023; Wang et al., 2023; Zhang et al., 2023; Liu et al., 2023), have been proposed, our approach differs fundamentally in its core contribution. Whereas such work focuses on scaling model capacity and live data ingestion, our work introduces a novel and reproducible methodology for creating the supervision data itself. By integrating the relevant financial context with behavioral finance studies, we construct a high-quality reasoning dataset designed to train smaller, more efficient end-to-end advisors that are grounded in sound, unbiased principles from their inception.

3 Dataset construction

3.1 Data Collection and Processing

Our first step was to collect a large pool of real-world finance questions. Reddit (Reddit, [2025]) proved ideal as a source of complex scenarios that span the breadth of personal finance domains—from debt consolidation and retirement planning to tax optimization and insurance decisions. The platform's subreddits, particularly r/personalfinance, which receives hundreds of thousands to millions of user queries, contain authentic scenarios that capture the intricate, multi-faceted nature of real financial decision-making, providing the scenario diversity essential for training comprehensive advisory models.

To comply with Reddit's terms and conditions, we exclusively utilized publicly available archived data from posts prior to June 2023, ensuring all collected queries were ethically sourced and properly de-identified.

After ingestion, we filtered the raw corpus in two

Table 1: **A detailed breakdown of the dataset generated via our proposed framework**. This table presents the distribution of approximately 19k samples across eight distinct categories of personal finance. Each category includes key metrics, such as the average token count for the initial query, the generated chain-of-thought delineating the reasoning steps, and the final answer.

Category	Description	Count	Avg. Query Tokens	Avg. CoT Tokens	Avg. Response Tokens
Debt Management & Credit	Strategies for debt reduction (e.g. snowball, avalanche), credit-score improvement, and loan analysis.	5175	215.76	628.30	393.69
Retirement Planning	Strategies, income-needs analysis, benefits optimization (e.g. 401(k), pensions) and withdrawal strategies.	3286	198.10	648.28	407.02
Tax Planning & Optimization	Tax-minimization strategies, understanding deductions and credits, and investment-tax implications.	3019	182.96	630.20	397.81
Investing & Wealth Building	Investment strategies based on risk tolerance, diversification, asset allocation, and long-term growth.	2994	200.16	653.54	402.98
Budgeting & Cash-Flow Management	Creating budgets, tracking expenses, managing income streams, and improving cash flow.	2503	221.53	628.71	394.47
Insurance & Risk Management	Assessing insurance needs (life, health, property), understanding policies, and managing financial risks.	1035	213.86	621.53	389.65
Savings & Emergency Funds	Strategies for building savings, establishing emergency funds, and goal-based saving.	638	177.18	652.25	382.95
Estate Planning & Legacy	Wills, trusts, inheritance considerations, and minimising estate taxes (accounting for regional variations).	196	216.90	653.47	409.06

stages:

- **Topical validity** – retained posts that contained an explicit, answerable personal finance question (e.g., budgeting, credit, retirement), discarding generic news, advertisements, or off-topic commentary.
- **Contextual clustering** – grouped semantically similar posts and removed near-duplicates to reduce noise.

This pipeline yielded 405k unique questions. We sampled 19k representative queries that span eight thematic categories. Table 1 contains the detailed description of the final dataset generated using the framework. The entire 405k-item corpus remains available for future scaling. Details about prompt templates and specific instructions used in each phase of the generation framework are presented in **Appendix A.1**.

3.2 Generation methodology

On a high level, the dataset generation framework can be divided into two parts: (i) chain-of-thought generation and (ii) response generation.

Our chain-of-thought generation is divided into four major phases, as illustrated in **Fig. 1**. This modular approach helps us focus on developing an independent rubric for each phase while giving the ability to stitch them together as a coherent chain-of-thought.

3.2.1 Query Analysis

The issue with natural language inquiries is the potential inconsistency of the information supplied to the model. There may be significant redundancy, or essential information may be hidden at times. Thus, the initial stage of answer creation, the question analysis phase, serves as a fundamental step in which the user's question is deconstructed into its essential components. This is required to ascertain the (i) primary conflict from the user's input; (ii) the principal players in the dilemma; and (iii) the essential financial facts to address the inquiry. This facilitates the optimization of subsequent cognitive processes while remaining aligned with the user's inquiry.

3.2.2 Context Analysis

Context analysis (Modular RAG). After intent parsing, we assemble a compact evidence pack via a modular RAG framework (Gao et al., 2024) built on two self-curated corpora snapshotted through **February 2025**: (i) a **financial corpus** of ∼600k

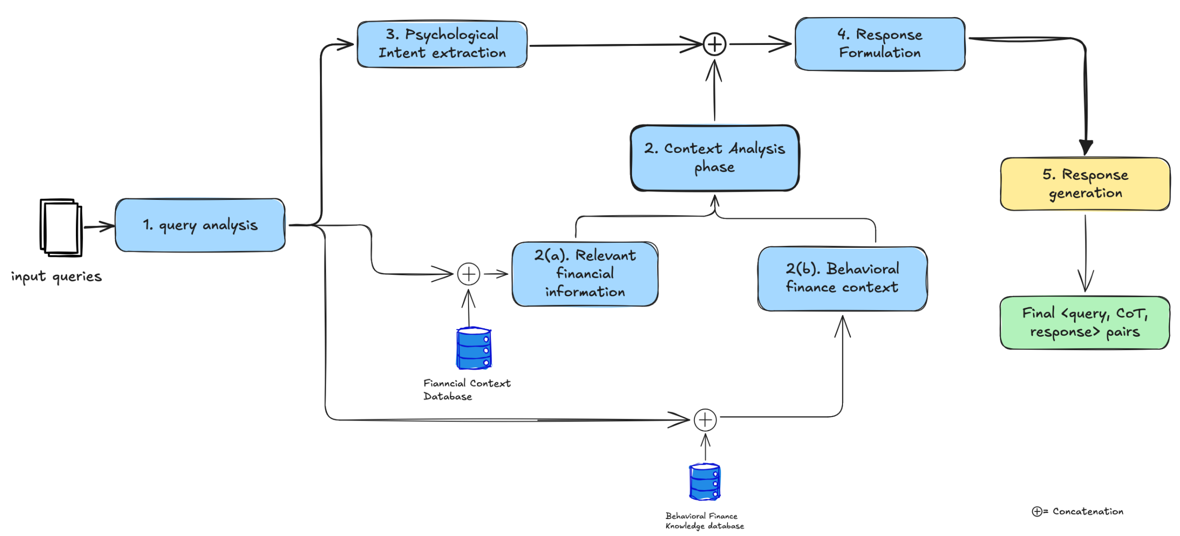


Figure 1: Dataset generation pipeline. Four modular chain-of-thought phases feed into final response generation. Each phase includes LLM-jury validation (not shown) to ensure quality.

tokens—practical sources such as Investopedia and a Bogleheads snapshot (Investopedia, 2025; Bogleheads, 2025) covering core concepts (e.g., retirement accounts, debt-repayment strategies), plus curated summaries of policy changes for major U.S. credit-card products and other consumer-policy/market updates; and (ii) a **behavioral corpus** of ∼300k tokens—research and practitioner write-ups spanning psychology of risk, investor behavior, behavioral portfolio theory, behavioral asset pricing, psychological effects of debt, and generational differences.

Candidate chunks are retrieved with `text-embeddings-3-large` (OpenAI, 2025b) (top-25), re-ranked with `all-MiniLM-L12-v2` (Sentence-Transformers, 2021), and the top-15 are condensed by gemini-2.0-flash (Google, 2025; Team et al., 2025a) to remove residual noise and unify terminology. The streamlined context and the user query then feed the downstream reasoning stage. Further details are provided in Appendix B.

3.2.3 Psychological Cue identification

In parallel to context identification, a psychological cue identification module is run to identify cues from the text. We extract the overall sentiment of the text, the primary emotions identifiable from the choice of words in the query, and the level of certainty present in the information. Using these cues, we try to generalize a set of communicative intents that might be behind the user's query. By breaking down the assessment into four distinct categories, the process ensures a comprehensive evaluation of the user's intent. This intent is utilized to direct the final response into a tone that is most suitable for the user, rather than directly providing them a monotonous response.

To operate the cue-identification at scale, and in line with the prior studies which demonstrate that state-of-the-art large language models outperform human annotators in judgment tasks(Bojić et al., 2025; O'Leary, 2025), we adopt an LLM-based framework for cue identification similar to the other stages in the framework.

3.2.4 Response Formulation

The final phase of the chain-of-thought is a distinct response formulation phase, in which we synthesize a set of instructions, consolidating information from all preceding phases. This produces a set of directives that must be adhered to throughout the response-generation phase.

3.3 Response generation

A conclusive response is formulated to address the user's inquiry, utilizing the previously optimized stages of information. This concluding comment is based on the financial context presented and is articulated in a suitable tone for the user.

3.4 Data Validation

Given that various open and proprietary LLMs automate numerous generations, there is a clear necessity to assess and authenticate their outputs. We employed a series of juries, specifically gemini-2.0-flash and o4-mini (OpenAI, 2025a), to evaluate and rank various generations for each phase. Each juror assessed the created information within a three-shot evaluation framework, ultimately selecting the highest-ranked response for subsequent generation jobs.

4 Evaluation

To test whether our dataset enables practical decision support, we fine-tune Qwen-3-8B (Yang et al., 2025) for five epochs and compare it with baselines of similar size.

We perform an additional assessment of the performance using two separate held-out datasets. We employ these methods to assess the quality of the responses through both quantitative and qualitative measures.

4.1 Quantitative Evaluation

To assess the quantitative performance of the models, we utilize a held-out dataset comprising 500 distinct queries across various categories of personal finance. Ground truths were produced by the generation framework presented in **Section 3.2** (not the fine-tuned model) prior to training and validated by independent jurors. Following the ground-truth generation, we calculate the BERTScore (Zhang et al., 2020) using the Qwen-3-8B-embeddings (Zhang et al., 2025) model to assess the semantic accuracy of the responses. We also calculate the BLEURT (Sellam et al., 2020) score to assess the fluency (or) human-likeness of the responses, respectively. The quantitative scores of various models utilized in this evaluation are detailed in **Table 2**.

Our 8B model achieves semantic accuracy comparable to leading baselines, including Gemma3-27B/12B and Mistral-24B. In particular, our model surpasses these larger models by approximately 3–5% in human-likeness and fluency. This indicates a reduced deviation from ground-truth data and enhanced fluency signals compared to models twice its size.

Model	BERTScore ↑	BLEURT
Gemma3-27B-IT (Team et al., 2025b)	**0.7142**	0.4374
Gemma3-12B-IT	0.7139	0.4390
Mistral-24B-2501 (MistralAI, 2025b)	0.7133	0.4464
QWQ-32B (Qwen, 2025) (reasoning)	0.7069	0.4452
DeepSeek-Qwen-14B (reasoning) (DeepSeekAI, 2025)	0.7069	0.4513
Ours (8 B)	0.7000	**0.4600**
Llama-3 8B (Meta, 2024)	0.6881	0.4547
Mistral-7B v0.3 (MistralAI, 2025a)	0.6650	0.4501

Table 2: Automatic evaluation on the 500-query test set. Bold marks the best score in each column; higher is better.

4.2 Qualitative Evaluation

To complement reference-based metrics and, critically, to assess the model's generalization capabilities, we run a list-wise blind LLM-jury ranking on 504 queries that were entirely held out and unseen during the training phase. These test queries were collected from a subsequent time period to ensure no data contamination. Meanwhile, all the candidates were zero-shot generated in their respective default inference settings to get their best performance. This setup allows us to evaluate whether our fine-tuned model has merely learned to mimic the training data or if it has successfully internalized a generalizable framework for the response generation that can be applied to novel user problems.

To mitigate familial bias and leakage, we excluded judges from model families used anywhere in our pipeline. In particular, Gemini models were omitted because they were used during dataset generation/validation, and Qwen-family judges were omitted because the system under test is Qwen-8B. A few otherwise suitable judges were also excluded for cost reasons. The final judge pool comprises models from unrelated families; none overlapped with training or data-creation components.

For each query, every judge sees all k anonymized candidates simultaneously (no ground truth and no model identities) and returns a full ranking; candidate order is uniformly randomized per replicate. We use two main judges, namely DeepSeek-V3-0324 (DeepSeek-AI et al., 2025) and Kimi-k2 (AI, 2025). Kimi-k2 is run three times, and DeepSeek-v3-0324 is run five times on independently shuffled anonymized candidate orders for each query to reduce possible biases. These judges were chosen in order to avoid same-family bias prevalent in modern LLM-judge studies.

Table 3: Rank correlations between judge sets (higher is better). τ measures how often the judges agree with A > B, and ρ measures how closely the full rank lists track.

Metric	Kendall's τ	Spearman's ρ
Plausibility	0.6183	0.7711
Accuracy	0.6183	0.7635
Relevance	**0.6910**	**0.8264**
Overall	0.6429	0.7904

The rankings are converted to Borda points(Saari, 2023) and averaged across judges and replicates to obtain the representative score of a response. We receive the ranking judgments according to three criteria, namely their financial accuracy, plausibility, and relevance to the query, and report the aggregate Borda scores in **Fig.2**. Whereas **Appendix C.1** presents the in-depth analysis of the evaluation results.

To examine rank consistency between the judge sets, we compute Kendall's τ and Spearman's ρ over per-query model ranks. Kendall's τ assesses pairwise order agreement (do both judges prioritize model A above model B?). Spearman's ρ assesses how closely the complete ranked lists move together and penalizes significant rank differences. We observe $\tau \approx$ 0.62-0.69 and $\rho \approx$ 0.76-0.83 (overall $\tau = 0.64$, $\rho = 0.79$), indicating substantial agreement. The consistently higher ρ than τ suggests disagreements are mostly local swaps rather than wholesale reorderings. Relevance demonstrates the strongest alignment ($\tau = 0.691$, $\rho = 0.826$). **Table 3** shows τ and ρ for each metric and overall.

Our experimental results demonstrate that a well-curated, behavior-tuned finance dataset can elevate an 8B open model to achieve performance parity with models two to three times its size, thus validating the practical utility of our framework. Details about the entire training environment and settings are presented in **Appendix D**.

4.3 Qualitative Analysis and Error Patterns

Analysis of the 504 held-out responses reveals consistent patterns across the three evaluation dimensions. When models produce inaccurate responses, they typically also exhibit degraded reasoning quality—accuracy and plausibility failures often co-occur. However, relevance remains relatively independent; responses can stay on-topic and address

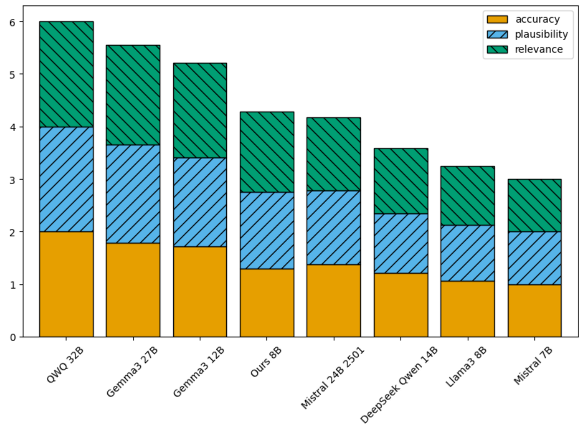


Figure 2: LLM-jury evaluation on 504 unseen subreddit queries: stacked bars show Borda-average scores for accuracy (blue), plausibility (orange), and relevance (green); taller bars indicate stronger overall preference. Our 8 B system (fourth from left) outperforms all other sub-14 B models and approaches the 27 B–32 B leaders. The y-axis represents the average Borda points a model has received.

user constraints even when containing factual errors or poor reasoning.

Strengths. The model consistently produces well-structured responses with clear headers, sequential action steps, and appropriate empathetic framing. It reliably extracts user-specific details (monetary amounts, timelines, constraints) and incorporates them into tailored advice. Responses typically acknowledge emotional context before providing practical guidance—a pattern that enhances perceived helpfulness.

Failure Modes. The primary weakness is factual hallucination, particularly for jurisdiction-specific regulations and tax details. The model occasionally generates plausible-sounding but incorrect specifics (e.g., non-existent grant programs, outdated tax brackets). These errors are most frequent in regulation-heavy domains (taxes, insurance) and least common in general planning tasks (budgeting, debt management).

Implications. While the model maintains strong structural and empathetic qualities across all responses, factual grounding remains the key bottleneck. This suggests that adding targeted retrieval for regulatory information and calculation verification would yield the highest marginal improvement. Even with current limitations, the model's consistent task alignment and user-responsive framing provide practical utility for non-critical advisory scenarios.

4.4 Cost Analysis

Beyond performance metrics, practical deployment requires careful cost consideration. **Table 4** presents a comprehensive cost analysis of the model produced by our framework against several baselines, comparing hosting infrastructure, inference latency, and total operational expenses.

Our data-centric approach delivers exceptional cost efficiency in the personal finance domain. By enabling a compact 8B model to achieve performance competitive with much larger systems, our method facilitates at least an 80% reduction in operational costs when compared to baselines with over 12B parameters. This dramatic cost reduction stems from targeted behavioral integration and principled data construction, rather than sheer computational scale.

The efficiency translates to practical deployment advantages: at a hosting cost of $0.8 per hour and an average inference time of 34.15 seconds, our model enables responsive financial advisory services without prohibitive infrastructure requirements. These results validate the effectiveness of our novel data generation framework. They demonstrate that by carefully integrating financial and behavioral signals into training data, it is possible to create competent, domain-specific models that are also economically viable. This presents a compelling approach for developing production-ready financial advisory tools that do not rely solely on expensive, large-scale models.

5 Future Works

We will advance this research by first determining the optimal path for global scaling: either finalizing a US-optimized pipeline for systematic market porting or—contingent on high-precision detection of regional signals (e.g., currency symbols, policy terminology, and spelling conventions)—implementing a Mixture-of-Experts (MoE) framework. In the latter case, a shared backbone model will process universal financial logic while lightweight regional experts handle localized nuances. This core model will deploy as a backbone policy within a thin agentic stack, minimizing latency and cost by resolving queries internally and invoking external tools (e.g., regulatory databases or fact-checking APIs) only for uncertainty resolution. We will rigorously measure resulting cost-latency trade-offs across regions. Rather than additional supervised fine-tuning, we will treat financial advice generation as an alignment problem, testing preference-based optimization (e.g., DPO/IPO) to refine outputs and deploying rule-based compliance layers to enforce regulatory fidelity, bias mitigation, and tone consistency. Success will be quantified through targeted evaluations of safety, compliance adherence, and user trust metrics.

6 Conclusion

Our research establishes a data-centric framework that enables an 8B-parameter model to achieve semantic fidelity and human-likeness on par with, and sometimes exceeding, 27–32B baselines in our held-out evaluations and blind LLM-jury study. On a 500-query test, the model outperforms Gemma3-27B by 5% on BLEURT and is competitive on BERTScore, with only a 2% difference; jury rankings show the 8B system approaching the 27–32B leaders. These gains stem from three synergistic components: explicit psychological cues, retrieval-augmented grounding, and a thin agentic execution layer. The modular design supports incremental extension (e.g., regional experts with minimal retraining). While geographic scope, behavioral depth, and privacy safeguards remain limitations, this work offers a cost-aware backbone for standalone personal-finance assistants and a viable alternative to monolithic cloud deployments—leaving a precise cost/latency audit to future work.

Limitations

Several aspects of our work leave room for future improvements. First, our study is limited to inquiries sourced solely from Reddit, which may overlook other demographics and query formats, suggesting a need for more diverse data sources. Second, our 19k sample dataset, though sufficient for proof-of-concept, lacks the scale and diversity needed to cover the full spectrum of real-world personal finance scenarios. Future work should expand the corpus with varied sources beyond Reddit to improve generalization. Third, our psychological analysis remains rudimentary, deriving only basic sentiment from phrases rather than incorporating enhanced psychological indicators such as risk tolerance or financial stress through specialized surveys or transfer learning from clinical datasets. Finally, our framework's scope excludes tasks beyond core natural language processing, particularly multi-modal data processing and reasoning capabilities, which represent critical areas for future

Table 4: Cost and Inference Performance Analysis for Deployment. Total costs reflect the expense to infer 504 queries from the test set, with each model benchmarked using four concurrent requests.

Model	Size (GB)	Endpoint Cost ($/h)	GPU	Inference Time (s/query)	Total Time (h)	Total Cost ($)
QWQ-32B	65.0	3.8	4xL4	167.86	5.82	22.33
Gemma3-27B	46.4	2.5	1xA100	64.34	2.23	5.63
Gemma3-12B	20.0	1.8	1xL40S	58.26	2.02	3.67
Ours (8B)	**16.4**	**0.8**	**1xL4**	**34.15**	**1.19**	**0.96**
Mistral-24B-2501	46.1	3.8	1xA100	37.99	1.32	5.05
DeepSeek-Qwen-14B	29.5	1.8	1xL40S	54.18	1.88	3.41
Llama3-8B	16.1	0.8	1xL4	33.58	1.17	0.94
Mistral-7B	14.5	0.8	1xL4	29.15	1.01	0.82

research expansion.

Acknowledgements

I want to express my sincere gratitude to **Raghu Ram Theerthala (KPIT Technologies)** for his valuable contributions to the related works section and insightful discussions during the brainstorming sessions that helped shape this research. I am grateful to Prathyusha Akundi, Syed Md. Bilal, Ashish Kubade, and Sai Narayan for their careful review of the manuscript and constructive feedback that improved the clarity and quality of this work. This research was supported by Perfios Software Solutions, which sponsored the computational costs and infrastructure required for model training and evaluation.

Data & Code Availability

The dataset, model, and code artifacts described in this paper are publicly available on Hugging Face. All data has been de-identified following the ethical guidelines described in Section 6, with personally identifiable information removed from Reddit sources. The resources are released under the Apache 2.0 license to facilitate reproducibility and future research in behavioral finance and LLM applications.

The following resources are available:

- **Model**: Fine-tuned Qwen-3-8B model at https://huggingface.co/Akhil-Theerthala/Kuvera-8B-qwen3-v0.2.1
- **Dataset**: 19k sample reasoning dataset at https://huggingface.co/datasets/Akhil-Theerthala/Kuvera-PersonalFinance-V2.1

Ethical Considerations

We curate data from publicly available Reddit posts and aggressively de-identify them: usernames/links/metadata are removed, PII (e.g., names, emails, phone numbers, addresses, IDs) is scrubbed, and queries are lightly rephrased so only the financial situation remains; no raw identifiers are stored or released. The system is for educational use only—not fiduciary or personalized financial advice—and our prompts/filters forbid unsafe guidance (e.g., evasion, "guaranteed returns"). Evaluation uses multiple LLM judges; we report inter-judge agreement and run judge-swap checks to limit model-family bias.

References

Rohit Aggarwal and Harpreet Singh. 2024. Overcoming limitations of ai agents: Integrating tacit knowledge through inferred latent themes. *Available at SSRN 4843878*.

Khushbu Agrawal. 2012. A conceptual framework of behavioral biases in finance. *IUP Journal of Behavioral Finance*.

Moonshot AI. 2025. Kimi-k2-instruct (revision 2f7e011).

H Kent Baker, Greg Filbeck, and Victor Ricciardi. 2017. How behavioural biases affect finance professionals. *The European Financial Review*, pages 25–29.

Bogleheads. 2025. Bogleheads - investing advice inspired by john bogle.

Ljubiša Bojić, Olga Zagovora, Asta Zelenkauskaite, Vuk Vuković, Milan Čabarkapa, Selma Veseljević Jerković, and Ana Jovančević. 2025. Comparing large language models and human annotators in latent content analysis of sentiment, political leaning, emotional intensity and sarcasm. *nature briefing*.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. 2025. Why do multi-agent llm systems fail? *Preprint*, arXiv:2503.13657.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

DeepSeekAI. 2025. deepseek-ai/deepseek-r1-distill-qwen-14b.

Han Ding, Yinheng Li, Junhao Wang, and Hang Chen. 2024. Large language model agent in financial trading: A survey. *Preprint*, arXiv:2408.06361.

Yunfan Gao, Yun Xiong, Meng Wang, and Haofen Wang. 2024. Modular rag: Transforming rag systems into lego-like reconfigurable frameworks. *Preprint*, arXiv:2407.21059.

Google. 2025. Gemini 2.0 flash. `https://cloud.google.com/vertex-ai/generative-ai/docs/model-reference/inference`.

Almog Gueta, Amir Feder, Zorik Gekhman, Ariel Goldstein, and Roi Reichart. 2025. Can llms learn macroeconomic narratives from social media? *Preprint*, arXiv:2406.12109.

Udit Gupta. 2023. Gpt-investar: Enhancing stock investment strategies through annual report analysis with large language models. *arXiv preprint arXiv:2309.03079*.

Xuewen Han, Neng Wang, Shangkun Che, Hongyang Yang, Kunpeng Zhang, and Sean Xin Xu. 2024. Enhancing investment analysis: Optimizing ai-agent collaboration in financial research. *Preprint*, arXiv:2411.04788.

Oudom Hean, Utsha Saha, and Binita Saha. 2025. Can ai help with your personal finances? *Applied Economics*, page 1–9.

Zengyi Huang, Chang Che, Haotian Zheng, and Chen Li. 2024. Research on generative artificial intelligence for virtual financial robo-advisor. *Academic Journal of Science and Technology*, 10(1):74–80.

Investopedia. 2025. Investopedia. `https://www.investopedia.com/`.

Satyadhar Joshi. 2025. A comprehensive review of gen ai agents: Applications and frameworks in finance, investments and risk domains. *International Journal of Innovative Science and Research Technology*, pages 1339–1355.

Kausik Lakkaraju, Sara E Jones, Sai Krishna Revanth Vuruma, Vishal Pallagani, Bharath C Muppasani, and Biplav Srivastava. 2023. Llms for financial advisement: A fairness and efficacy study in personal decision making. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 100–107, New York, NY, USA. Association for Computing Machinery.

Meisin Lee and Soon Lay-Ki. 2024. 'finance wizard' at the finllm challenge task: Financial text summarization. *Preprint*, arXiv:2408.03762.

Jinzheng Li, Jingshu Zhang, Hongguang Li, and Yiqing Shen. 2024. An agent framework for real-time financial information searching with large language models. *Preprint*, arXiv:2502.15684.

Xiao-Yang Liu, Guoxuan Wang, Hongyang Yang, and Daochen Zha. 2023. Data-centric fingpt: Democratizing internet-scale data for financial large language models. *NeurIPS Workshop on Instruction Tuning and Instruction Following*.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, Chao Li, Sheng Xu, Dezhi Chen, Yun Chen, Zuo Bai, and Liwen Zhang. 2025. Fin-r1: A large language model for financial reasoning through reinforcement learning. *Preprint*, arXiv:2503.16252.

Zian Liu and Renjun Jia. 2025. Llm4fts: Enhancing large language models for financial time series prediction. *Preprint*, arXiv:2505.02880.

Yichen Luo, Yebo Feng, Jiahua Xu, Paolo Tasca, and Yang Liu. 2025. Llm-powered multi-agent system for automated crypto portfolio management. *arXiv preprint arXiv:2501.00826*.

Glenn Matlin, Mika Okamoto, Huzaifa Pardawala, Yang Yang, and Sudheer Chava. 2025. Finance language model evaluation (flame). *Preprint*, arXiv:2506.15846.

Kiana Jafari Meimandi, Gabriela Aránguiz-Dias, Grace Ra Kim, Lana Saadeddin, and Mykel J. Kochenderfer. 2025. The measurement imbalance in agentic ai evaluation undermines industry productivity claims. *Preprint*, arXiv:2506.02064.

Meta. 2024. meta-llama/llama-3.1-8b-instruct.

MistralAI. 2025a. mistralai/mistral-7b-instruct-v0.3.

MistralAI. 2025b. mistralai/mistral-small-24b-instruct-2501.

Izunna Okpala, Ashkan Golgoon, and Arjun Ravi Kannan. 2025. Agentic ai systems applied to tasks in financial services: Modeling and model risk management crews. *Preprint*, arXiv:2502.05439.

Daniel E. O'Leary. 2025. Editorial: Analysis of sentiment estimates and cognitive fallacies in large language models. *Intelligent Systems in Accounting, Finance and Management*, 32(3):e70010. E70010 9691779.

OpenAI. 2025a. o3 and o4-mini system card. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf.

OpenAI. 2025b. Openai text-embeddings-3.

Qwen. 2025. Qwen/qwq-32b.

Reddit. [2025]. Reddit: The heart of the internet. https://www.reddit.com.

Donald G. Saari. 2023. Selecting a voting method: the case for the borda count. *Constitutional Political Economy*, 34(3):357–366.

Javier Sanz-Cruzado, Edward Richards, and Richard McCreadie. 2024. Far-ai: A modular platform for investment recommendation in the financial domain. In *Advances in Information Retrieval*, pages 267–271, Cham. Springer Nature Switzerland.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. Bleurt: Learning robust metrics for text generation. *Preprint*, arXiv:2004.04696.

Sentence-Transformers. 2021. all-minilm-l12-v2.

Takehiro Takayanagi, Kiyoshi Izumi, Atsuo Kato, Naoyuki Tsunedomi, and Yukina Abe. 2023. Personalized stock recommendation with investors' attention and contextual information. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 3339–3343, New York, NY, USA. Association for Computing Machinery.

Takehiro Takayanagi, Kiyoshi Izumi, Javier Sanz-Cruzado, Richard McCreadie, and Iadh Ounis. 2025a. Are generative ai agents effective personalized financial advisors? *Preprint*, arXiv:2504.05862.

Takehiro Takayanagi, Masahiro Suzuki, Kiyoshi Izumi, Javier Sanz-Cruzado, Richard McCreadie, and Iadh Ounis. 2025b. Finpersona: An llm-driven conversational agent for personalized financial advising. In *Advances in Information Retrieval*, pages 13–18, Cham. Springer Nature Switzerland.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025a. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025b. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Kesen Wang, Daulet Toibazar, Abdulrahman Alfulayt, Abdulaziz S. Albadawi, Ranya A. Alkahtani, Asma A. Ibrahim, Haneen A. Alhomoud, Sherif Mohamed, and Pedro J. Moreno. 2025. Multi-agent interactive question generation framework for long document understanding. *Preprint*, arXiv:2507.20145.

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *NeurIPS Workshop on Instruction Tuning and Instruction Following*.

Philipp Winder, Christian Hildebrand, and Jochen Hartmann. 2024. Biased echoes: Generative ai models reinforce investment biases and increase portfolio risks of private investors.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.

Boyu Zhang, Hongyang Yang, and Xiao-Yang Liu. 2023. Instruct-fingpt: Financial sentiment analysis by instruction tuning of general-purpose large language models. *FinLLM Symposium at IJCAI 2023*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yuhan Zhi, Xiaoyu Zhang, Longtian Wang, Shumin Jiang, Shiqing Ma, Xiaohong Guan, and Chao Shen. 2025. Exposing product bias in llm investment recommendation. *Preprint*, arXiv:2503.08750.

Yuhang Zhou, Yuchen Ni, Zhiheng Xi, Zhangyue Yin, Yu He, Gan Yunhui, Xiang Liu, Zhang Jian, Sen Liu, Xipeng Qiu, Yixin Cao, Guangnan Ye, and Hongfeng Chai. 2025. Are LLMs rational investors? a study on the financial bias in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24139–24173, Vienna, Austria. Association for Computational Linguistics.

David Zibriczky. 2016. Recommender systems meet finance: A literature review. In *International Workshop on Personalization & Recommender Systems in Financial Services*.

Appendices

A Prompting Guidelines followed for the generation and evaluation stages

A.1 Guidelines followed in the generation stage.

This section focuses on outlining the guidelines followed in crafting the prompts for each phase of generating and evaluating the outputs.

A.1.1 Overarching principles

There are three core principles followed for the process of crafting the prompts:

a. Modularity

b. Deconstruction

c. Personification

The goal of the overall prompt crafting is to keep the overall structure of the prompts similar and swappable depending on the task at hand. As with the framework, where the complex task of generating a suitable response is broken down into individual phases, the prompts are broken down to make sure the structure of the instructions given to the model remains the same.

Each stage of the prompting had a unique, suitable persona (e.g., linguistic analysis expert, expert financial reasoning engine). This role-playing technique primes the model to access relevant knowledge, adopt the appropriate tone, and constrain its behavior to the specific requirements of the task.

The generic structure of the prompt is as follows:

```
"""
You are a {persona}, whose task is to
↪ {task_details}.

### INSTRUCTION ###
{instructions_for_the_task}

### Key Points ###
{key_points_to_keep_in_mind}

---
**Inputs**:
{inputs}
---
**Your Response**:"""
```

A.1.2 Individual Phases

1. **Classification:**

   a. The primary goal of this stage is to classify incoming user queries into suitable categories of personal finance. The prompt constrains the model by forcing a single-label classification (ONE of the following) based on PRIMARY INTENT, which prevents ambiguity and ensures a decisive output for downstream routing.

   b. Each category has a Scope and an example that the model uses to make its decisions. If the query does not fall into any of the categories, the query is labeled Not_Applicable.

2. **Query Analysis:**

   a. The primary goal of this prompt is to direct the model to break down the user query into more specific and manageable pieces of information.

   b. Since most of the user queries on Reddit and in general are often filled with unrelated noise, this stage directs the model to distil the user's query into essential semantic elements, eliminating the conversational distractions and concentrating on actionable concerns and their impact on the key stakeholders.

3. **Context Analysis:**

   a. The context analysis is one of the key prompts that influences the quality of the output by the framework. The prompt directs the final model to generate actionable and insightful contextual summaries that are placed into the model's natural chain-of-thought.

b. The prompt explicitly asks for a Concise chain-of-thought Analysis Block and instructs the model that this is an internal reasoning step, not the final answer. This step forces the model to externalise its reasoning process, exploring multiple scenarios and their consequences before concluding.
c. By requiring the model to detail the Stakeholder Impact for each approach, the prompt ensures a holistic analysis that considers the financial and emotional consequences for all relevant parties mentioned in the query. This scenario-based analysis moves the responses beyond simple fact-based analysis to a more human-centred form of reasoning.

4. **Psychological analysis**

   a. The goal of this prompt is to direct the model and extract the key information about the user's state of mind when asking the query.
   b. The prompt demands that every conclusion about sentiment, emotion, or intent be justified by referencing specific words or phrases. This approach grounds the analysis in textual evidence, preventing the model from making unfounded psychological assumptions and improving the explainability of its affective understanding.
   c. This analysis is a separate step from the financial reasoning (Context Analysis). This deliberate separation prevents the user's emotional state from biasing the objective financial analysis, and vice-versa, allowing for a final response that can synthesise both aspects without compromising either.

5. **Response Rubric**

   a. This stage consolidates all the previously collected information and creates a complete rubric that can direct the model into generating the final response.
   b. The key information from the previous stages gets highlighted while being linked to different parts of the user query for easier reference and understanding.

6. **Response Generation**

   a. This final stage synthesises all preceding analyses into a coherent, user-facing response.
   b. The prompt provides the model with all previous outputs (the original query and the comprehensive chain-of-thought) and explicitly instructs it to integrate both factual accuracy and emotional intelligence seamlessly. It acts as a final "assembly" instruction, guiding the model on how to combine the rational and affective components.
   c. The use of clear positive (Do) and negative (Do not) instructions creates strict behavioral boundaries. For instance, "Do not reference the chain-of-thought analysis" ensures the final output is natural and user-friendly, hiding the complex underlying cognitive architecture from the end-user. These instructions create a helpful response without being robotic or transparent about its inner workings.
   d. These responses are generated in a way that ensure the ability to train non-reasoning models from the same dataset.

A.2 Prompt Guidelines for Evaluation through LLM-as-a-Judge

The goal of the evaluation is to determine which responses are naturally ranked better than the others. Since this is a list-wise ranking with a high room for confusion or hallucination, the evaluation criterion are strictly defined.

The overall prompt structure for each of the case are as follows:

```
"""
You are a {persona}. Your task is to rank financial advice responses from best to worst based *solely* on the strict definition of {target_aspect}.

### **Evaluation Criteria**
{Evaluation Criterion}

#### **I. Primary Criteria (What to look for):**
{primary_set_of_instructions}
```

```

#### **II. Explicit Penalties (What to
↪ penalize):**
{penalizing_instructions}

#### ** III. Key Points to note:**
{additional_instructions}
---

**Query:** {query}

**Responses to Rank:**
{anonymized_shuffled_model_responses}
"""
```

1. **Accuracy**:
    a. The goal of this prompt is to direct the model to review the search results and the query to estimate the accuracy of the output.
    b. The responses are penalized if and only if the responses demonstrate wrong/harmful advice (or) inappropriate financial concepts to the query.
    c. The model is specifically instructed not to penalise on the style or relevance of the response and solely focus on the accuracy of the financial concepts provided in the text. This guides the model to rank solely based on the accuracy of the financial concepts present in the response.
2. **Plausibility**:
    a. A response is defined to be plausible if it sounds reasonable and believable to a typical user. Some of the key characteristics include
        - Logical flow and coherent reasoning structure
        - Sensible approach to the problem
    b. A response is penalized if it contains unnecessarily verbose or contain excessive detail. The responses are also penalized if they contain complex or hard-to-follow reasoning.
    c. The model is specifically instructed not to penalise on the accuracy or relevance of the responses.
3. **Relevance**:
    a. A response is considered relevant if it address every component of the user's query. A relevant response should incorporate the specific figures, constraints, and details mentioned in the user's query, and answer the questions immediately without generic introductions.
    b. Any partial relevance or additional context not relevant to the query is penalized.

B Modular RAG for Context Analysis

Goal. Given a user query, the context-analysis phase assembles a compact, high-signal context pack from two specialized corpora: (i) *Behavioral insights* (behavioral economics and psychology) and (ii) *Financial concepts* (mainstream personal finance knowledge). The context pack is then passed to the response generator.

Corpora. **Behavioral insights** are sourced from peer-reviewed research and reputable psychology venues, complemented by carefully selected psychology blogs for practitioner framing. **Financial concepts** are drawn from practical, high-visibility sources such as Investopedia, Bogleheads, and other widely cited personal-finance viewpoints. All raw pages are converted to Markdown with headers and section structure preserved to retain document semantics.

Preprocessing and indexing.

- **Scraping & normalization:** We scrape public pages (respecting robots/terms), remove boilerplate (nav, ads), and normalize to Markdown with stable headings.

- **Semantic chunking:** Documents are segmented into *modular chunks* along header/semantic boundaries to keep each chunk topically coherent; we attach metadata (source, URL or handle, snapshot time, section path, corpus tag: `behavioral` or `financial`).

- **Dense indexing:** Each chunk is embedded with `text-embeddings-large-003` and stored in a vector databsase (ChromaDB).

Retrieval and re-ranking (per query).

1. **Dual retrieval:** From each index, retrieve the top-k candidates (k=25) using the query embedding.

2. **Cross-encoder re-ranking:** Concatenate candidates from both corpora and re-rank with a lightweight cross-encoder (`sentence-transformers/all-minilm-l12-v2`); keep top-m (m=15).

3. **LLM synthesis/filter:** A fast LLM (gemini-2.0-flash) receives {top-m chunks, query} and (a) extracts salient facts, definitions, and decision criteria; (b) discards residual off-topic spans; (c) emits a streamlined, source-attributed context.

Assembly and handoff. The streamlined context (with inline source attributions and corpus tags) is passed, together with the user input, to the final LLM that completes the context-analysis phase.

Behavioral vs. financial module roles. The **behavioral module** surfaces cognitive-bias descriptors, debiasing tactics, and user-state cues (e.g., loss aversion framing, present bias prompts). The **financial module** surfaces actionable rules of thumb, definitions, procedures, and typical constraints (e.g., contribution limits, insurance concepts, payoff ordering heuristics). Both modules contribute to the same context pack; behavioral cues guide *how* advice is framed, while financial chunks ground *what* advice is provided.

Limitations. (1) Coverage and staleness depend on the snapshot of public sources; (2) blogs can introduce style bias despite re-ranking; (3) the synthesis step may over-prioritize well-structured sources. We mitigate these by preserving source attributions, tracking snapshot timestamps, and prompting synthesis to prefer higher-priority sources when conflicts arise.

C Deeper Evaluation Results

C.1 Score Definitions and Rationale

We evaluate responses along three orthogonal axes—*Accuracy*, *Plausibility*, and *Relevance*—to separate factual correctness, reasoning quality, and task alignment. This decomposition avoids a single scalar that can reward fluent but unsafe answers or penalize terse yet correct ones, and it enables targeted error analysis and ablations.

Accuracy (financial correctness).

- **Objective.** Judge reviews the response against the query and retrieved evidence and scores only the *validity of financial concepts, calculations, and advice*.
- **Penalties.** Deductions occur *iff* the answer contains wrong or harmful guidance, or misapplies financial concepts to the user's situation.
- **Non-considerations.** Style, tone, verbosity, and even partial coverage are *not* penalized; the judge is instructed to focus exclusively on correctness.

Plausibility (reasoning quality).

- **Objective.** Assess whether the answer reads as reasonable and believable to a typical user—i.e., it exhibits a clear logical flow and a coherent problem-solving structure.
- **Penalties.** Overly verbose, needlessly complex, or hard-to-follow chains of reasoning are penalized.
- **Non-considerations.** Factual correctness and topical coverage are *not* scored here; the lens is purely rhetorical/structural.

Relevance (task alignment).

- **Objective.** Verify that the response directly addresses *every* component of the user's query, incorporates the user's numbers, constraints, and context, and answers without generic preambles.
- **Penalties.** Partial coverage, tangential content, or extra context not pertinent to the query is penalized.
- **Non-considerations.** Factual accuracy and stylistic polish are ignored for this axis.

C.2 Borda Points

Definition. For a listwise ranking of n systems, the item placed at rank r ($r = 1$ is best) receives a Borda score

$$b = n - r,$$

so the top entry gets $n - 1$ points and the last gets 0.

Motivation. Borda aggregation is well–suited to LLM-as-a-judge experiments where *relative* quality matters more than absolute scores:

- *Full-order utilisation*: every position contributes signal, ensuring that small but consistent advantages are captured rather than discarded by winner-takes-all rules.
- *Cardinal comparability*: with a fixed candidate set, raw points can be averaged across queries and judges without normalisation, giving a stable, interpretable mean.

- *Robustness to mild noise*: swapping adjacent middle ranks changes the total by only ± 1, so individual judge idiosyncrasies exert limited influence on the final average.

Interpretation. Higher mean Borda points indicate that a system outranks its peers more often. The maximum possible mean is $n - 1$; the gap to this ceiling offers an intuitive sense of head-room.

Limitations.

- *Rank-reversal*: inserting or removing a candidate can change every system's score, complicating longitudinal comparisons.
- *Independence of Irrelevant Alternatives (IIA) violation*: a judge's relative preference between two systems can affect, and be affected by, ranks assigned to others.
- *Equal-interval assumption*: the method treats the gap between successive ranks as uniform, ignoring situations where judges perceive larger quality jumps near the top.
- *Strategic susceptibility*: if human judges know what influences the aggregation, they could inflate or deflate lower ranks to benefit a favored system.

C.3 LLM-Jury Protocol

LLM-based judging scales across topics, is inexpensive, and achieves strong agreement with human raters when rubrics are explicit and task context is provided. It also captures holistic qualities (e.g., coherence, task fit) that single-number similarity metrics may miss.

It should be noted that zero-shot judging is vulnerable to *position bias* (earlier items rank higher), *same-family bias* (preference for outputs from the judge's own family), and prompt/leniency variance. We therefore (i) use *multi-shot* prompts to anchor criteria, (ii) evaluate with *listwise* ranking on *independently shuffled* candidate lists, and (iii) diversify judges across model families to minimize correlated bias.

Judge pool and prompting. We employ two main heterogeneous judges: *DeepSeek-v3-0324* (5-shot), *Kimi-k2* (3-shot). For each query and criterion (Accuracy, Plausibility, Relevance), judges rank anonymized model outputs in a single list. Few-shot exemplars are held constant within a run and varied across repeats to reduce overfitting to any one demonstration set. A subsample of these rankings were further validated by *o4-mini* model to consolidate the relative performance.

Scoring and aggregation (per criterion). For each query, judges perform *multi-shot listwise ranking* over anonymized outputs using the rubrics in Sec. C. Ranks are converted to raw Borda points $b = n - r$. We then:

1. average b across shuffles/repeats for each judge;
2. average across the judges to obtain a per-query, per-criterion score for each model;
3. average across all queries within a *category* (e.g., the "overall" set or a PF subcategory) to obtain the model's *criterion-wise mean* in that category.

The stacked bars in Fig. 2 display these criterion-wise means (*Accuracy*, *Plausibility*, *Relevance*) for each model. For a single category-level number, we also report the *unweighted average* of the three criterion-wise means as the model's final representation score in that category.

C.4 Overall Category Scores (Accuracy, Plausibility, Relevance)

We report criterion-wise means derived from the raw Borda points assigned by the LLM jury (Sec. C.3). For each criterion and model, scores are averaged across judges and queries within the overall set. Higher is better.

Accuracy. Figure 3 shows a size-tilted pattern: *QwQ-32B (reasoning)* leads, followed by *Gemma3-27B-it* and *Gemma3-12B-it*. *Mistral-Small-24B* sits between this top cluster and the rest. The proposed *8B* model is mid-pack—behind the leaders and the 24B baseline, but ahead of several 7–14B baselines. This points to factual calibration and retrieval/verification as the primary levers to close the gap, rather than rewriting or stylistic tuning.

Plausibility. As shown in Fig. 4, *QwQ-32B* ranks first, with *Gemma3-27B-it* next. The proposed *8B* clusters near the front: it exceeds the *Mistral-Small-24B* baseline but trails *Gemma3-12B-it*. This suggests that the dataset structure and few-shot conditioning induce coherent reasoning steps and a sensible flow even at mid scale.

Relevance. Figure 5 indicates strong task alignment at the top end (*QwQ-32B*, *Gemma3-27B-it*,

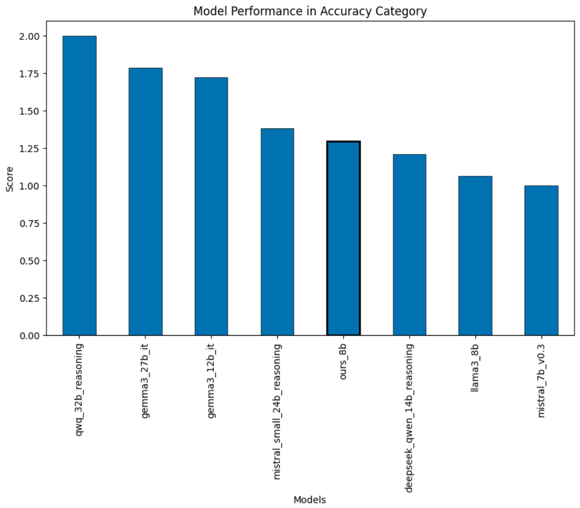


Figure 3: Accuracy (mean raw Borda points per query, averaged over judges). A size-driven lead is visible; the proposed 8B is mid-pack, indicating factual calibration as the primary improvement lever.

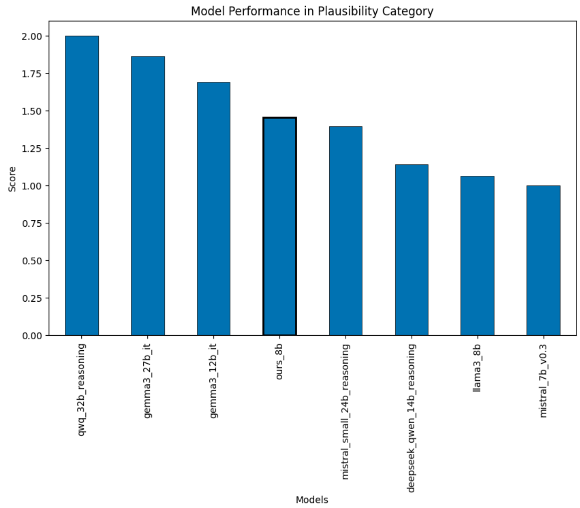


Figure 4: Plausibility (mean raw Borda points). The proposed 8B clusters near the front and matches or exceeds several larger baselines, reflecting strong logical flow and coherent reasoning.

Gemma3-12B-it). The proposed *8B* ranks next (4/8), ahead of the remaining baselines, suggesting it reliably maps user constraints and addresses all parts of the query without drifting into generic preambles. The residual gap likely reflects cases that require exhaustive edge handling (e.g., niche eligibility rules) rather than broad intent recognition.

Cross-criterion takeaway. Across criteria, the proposed *8B* model is *plausibility*– and *relevance*-competitive while lagging most on *accuracy*. The next steps of improvement is therefore to prioritize factual grounding and numeric checking: adding targeted retrieval, rule tables, and lightweight calculation guards should yield the largest absolute gains relative to effort.

C.5 Parameter Efficiency: Category-wise Borda per Billion Parameters

To evaluate *parameter efficiency* rather than absolute quality, we compute a per-parameter utility for each criterion. For model i with P_i billion parameters and mean raw Borda points $\bar{b}_{i,c}$ on criterion $c \in \{\text{Accuracy, Plausibility, Relevance}\}$ (averaged over judges and queries within the category), we define

$$e_{i,c} = \frac{\bar{b}_{i,c}}{P_i} \quad \text{(Borda points per billion parameters).}$$

This ratio captures the *marginal productivity of capacity*: how much judged quality is obtained per parameter, holding the evaluation protocol fixed. It is not a substitute for absolute scores (Sec. C.4), but a complementary lens for cost-, latency-, and memory-constrained deployments.

Relevance efficiency. Figure 6 shows the proposed *8B* model with the highest Borda-per-parameter in *Relevance*, followed by *Gemma3-12B-it*, then *Mistral-7B-v0.3* and *Llama3-8B*. Large reasoning models (e.g., *QwQ-32B*, *Gemma3-27B-it*) trail on this per-parameter metric despite strong absolute relevance (Fig. 5), indicating diminishing returns in alignment per unit capacity at larger scales.

Plausibility efficiency. As shown in Fig. 7, the proposed *8B* again leads, with *Mistral-7B-v0.3* and *Gemma3-12B-it* close behind (virtually tied), followed by *Llama3-8B*. This suggests that the dataset structure and few-shot conditioning yield coherent reasoning with high *utility density*—quality per parameter.

Accuracy efficiency. In Fig. 8, the proposed *8B* tops *Accuracy* per parameter, followed by *Mistral-7B-v0.3* and *Gemma3-12B-it* (near-tie). Models that dominate absolute accuracy (Sec. C.4) deliver lower accuracy *per parameter*, implying that targeted grounding and calculation checks can be more cost-effective than increasing model size.

Takeaways and caveats. (1) The proposed *8B* is the most parameter-efficient across all three criteria, reinforcing the central claim that careful

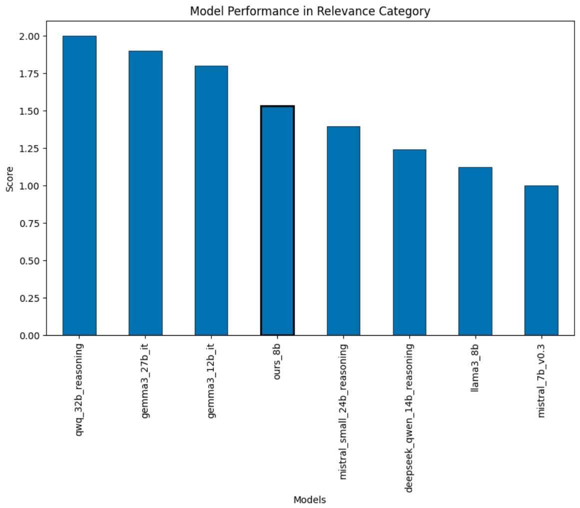


Figure 5: Relevance (mean raw Borda points). The proposed 8B ranks immediately behind the top three, ahead of other baselines, indicating consistent mapping from user constraints to concrete answers.

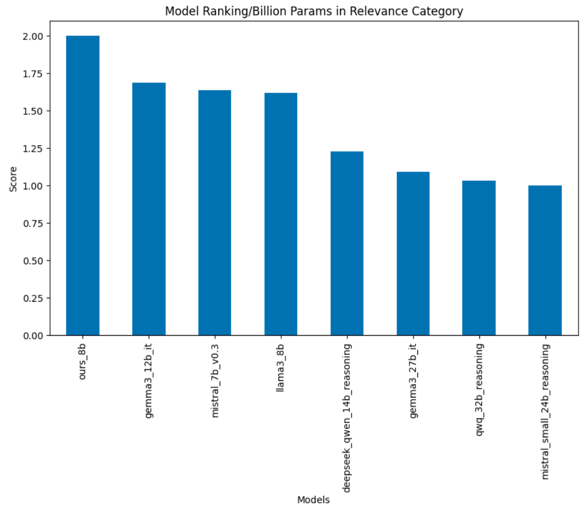


Figure 6: Relevance efficiency: mean raw Borda points per billion parameters (higher is better). The proposed 8B leads, followed by Gemma3-12B-it and Llama3-8B.

supervision can substitute for scale in personal-finance tasks. (2) Efficiency does not equal absolute quality; it informs deployment decisions where memory/latency are binding. (3) The ratio ignores runtime constants (KV-cache bandwidth, batch scheduling) and training cost; it should be read alongside absolute Borda results and system-level latency/memory budgets.

C.6 Qualitative Category-wise Evaluations

We analyze twelve personal-finance subdomains—*Auto, Budgeting, Credit, Debt, Employment, Housing, Insurance, Investing, Planning, Retirement, Saving, Taxes*. For each, we report criterion-wise means derived from normalized Borda points (Sec. C.3). The dashed horizontal line in each panel marks the cohort-wide mean for orientation.

Please note that the category-based evaluations in this appendix use raw Reddit post flairs, which differ from the eight thematic categories curated for the main analysis.

C.6.1 Relevance by Subdomain

Relevance captures task alignment: covering all parts of the user's request, using their numbers/constraints, and answering without generic preambles (Sec. C.1).

- A consistent top cluster is formed by larger reasoning-aligned models. The **proposed 8B model** sits immediately behind this cluster in most categories and hovers around the cohort mean.
- Strengths are most visible in *Budgeting, Employment, Planning* (and close-to-mean in *Insurance/Retirement*).
- Wider gaps appear in *Auto, Housing, Credit* (and occasionally *Investing/Taxes*), where locality- and rule-heavy edge cases require more exhaustive coverage.

C.6.2 Accuracy by Subdomain

Accuracy isolates *financial correctness*: advice and calculations must be right for the stated scenario; style and coverage are ignored (Sec. C.1).

- Absolute leaders are the larger models across most subdomains.
- The proposed 8B model is mid-pack overall, with competitive accuracy in *Debt, Planning, Employment*, and notably larger gaps in *Housing, Insurance, Taxes* (and *Credit*).
- This pattern suggests targeted *grounding* (policy/limit tables, calculators) is a higher-leverage fix than stylistic tuning for closing the remaining gap.

C.6.3 Plausibility by Subdomain

Plausibility measures reasoning flow and readability: clear structure, sensible steps, and absence of unnecessary complexity (Sec. C.1).

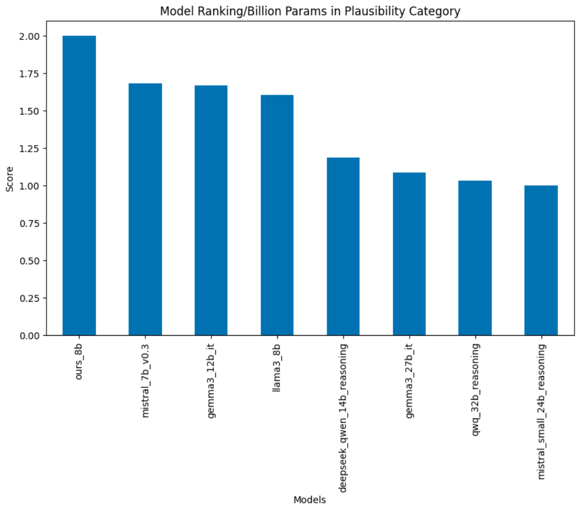


Figure 7: Plausibility efficiency: mean raw Borda points per billion parameters. The proposed 8B ranks first; compact 7–8B baselines are competitive, while very large models show lower utility density.

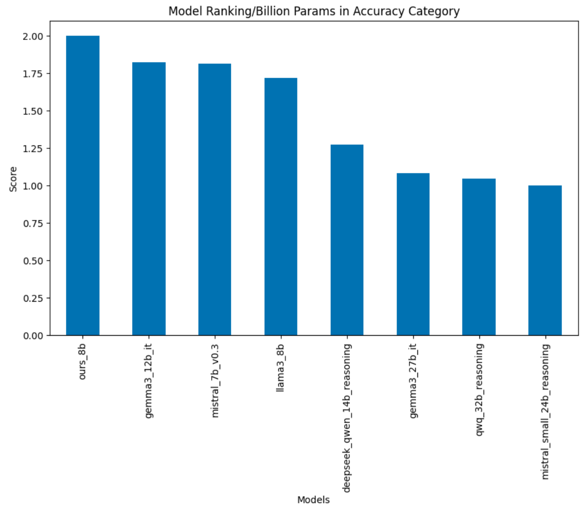


Figure 8: Accuracy efficiency: mean raw Borda points per billion parameters. The proposed 8B tops the cohort, indicating that factual calibration gains can be achieved more cheaply than by scaling parameters alone.

- The proposed 8B clusters close to the leaders across most subdomains, with stronger relative showings in *Debt* and *Planning*; margins are lower in *Taxes* and *Retirement*.
- Lower margins in regulation-dense areas mirror the accuracy pattern: where facts are brittle, judges penalize circuitous explanations.

C.7 Overall Summary, Limitations, and Next Steps

Summary. Taken together, the results tell a simple story. On absolute scores (Sec. C.4), the largest baselines lead across *Accuracy*, *Plausibility*, and *Relevance*, as expected. The proposed 8B model sits just behind this front cluster on *Relevance* and *Plausibility* and lands mid-pack on *Accuracy*. When we switch to a parameter-efficiency lens (Sec. C.5), the picture reverses: the 8B model delivers the highest Borda-per-parameter across all three metrics, indicating unusually high utility density for its size. The subdomain breakdown (Sec. C.6) is consistent with both views: the 8B model is steady or above-mean in everyday tasks such as *Budgeting, Planning, Employment* (and shows strong plausibility in *Debt*), while gaps widen in regulation- and table-heavy areas such as *Housing, Insurance, Taxes, Credit* (and occasionally *Auto/Investing*). In short, scale drives absolute peaks, but careful supervision yields competitive quality—and superior efficiency—at mid-scale.

These results suggest prioritizing minimal, high-leverage grounding over further size increases: include compact, versioned rule/limit tables for regulation-intensive domains (e.g., taxes, insurance, credit), add lightweight calculators/unit-tests for numeric steps, sharpen supervision with contrastive edge cases in brittle areas (tax/retirement), diversify judge checks (agreement and judge-swap), and extend evaluation to short multi-turn interactions that reward clarifying questions.

D Training Details

We fine-tuned the 8B parameter Qwen-3 model with AdamW optimizer on bfloat16 precision and a training split containing 15.6K samples and a validation set containing 2.6k samples. We trained the model for four epochs using an optimal batch size of 256, resulting in around 220 steps overall. The model underwent training on a solitary A100 GPU within the Runpod cloud GPU infrastructure for 3 hours.

We preserved three checkpoints per epoch, with the optimal validation loss attained at step 101. The training used a cosine learning rate schedule with a maximum learning rate of 5×10^{-5}, a 10% linear warm-up period of 21 steps (a warmup ratio of 10%), and a minimum learning rate of 5×10^{-6}. Gradients were constrained to a global norm of 1, weight decay was established at 0.01, and all other parameters adhered to the default conventions of the Hugging Face Trainer.

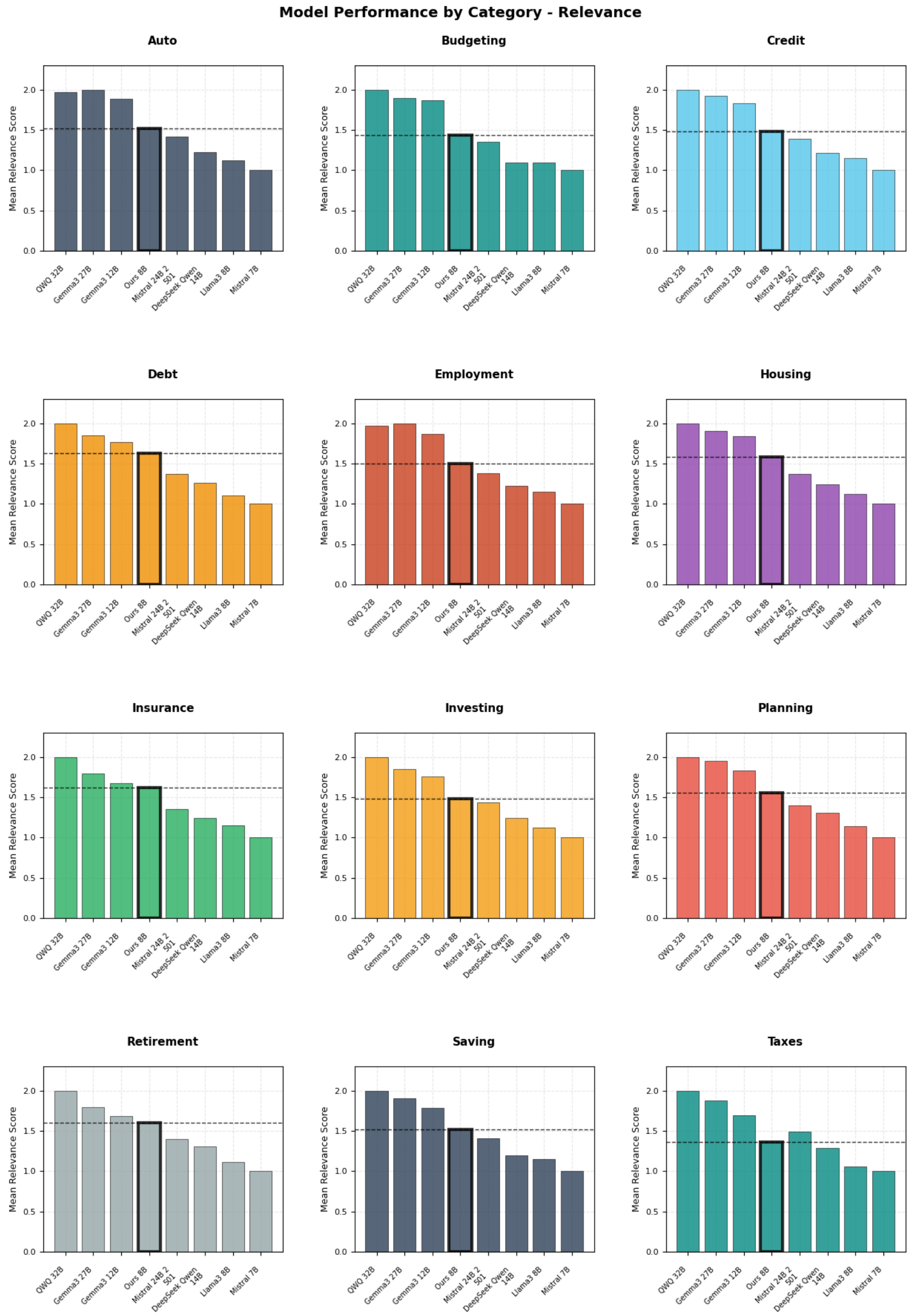


Figure 9: Category-wise **Relevance**. The proposed 8B model typically sits just behind the leading cluster and near the cohort mean; gaps are largest in edge-case, rule-dense areas (e.g., Auto, Housing, Credit).

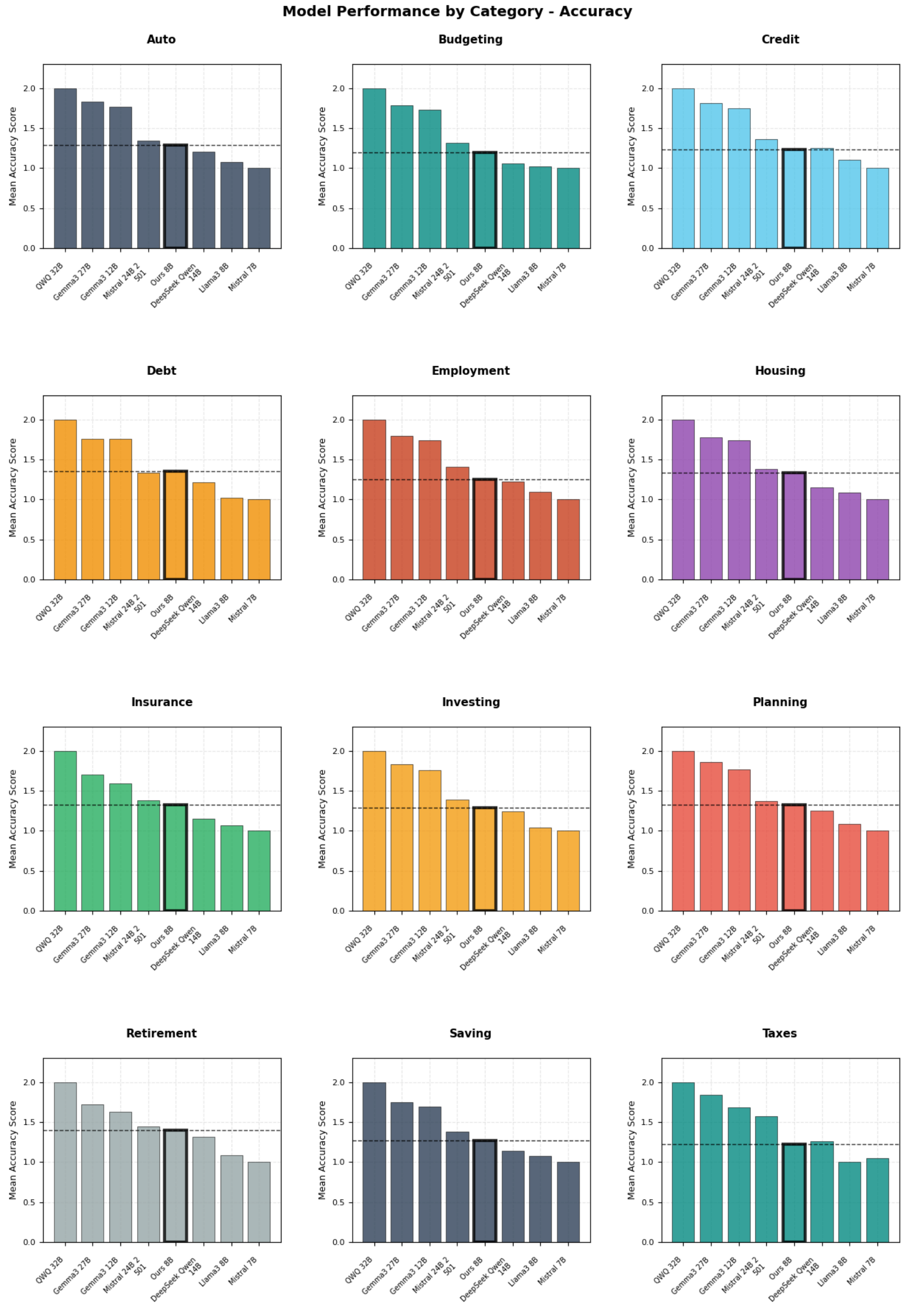


Figure 10: Category-wise **Accuracy**. Larger models lead overall; the proposed 8B is mid-pack with smaller gaps in everyday planning tasks and larger gaps where year-/jurisdiction-specific rules dominate (e.g., Housing, Insurance, Taxes).

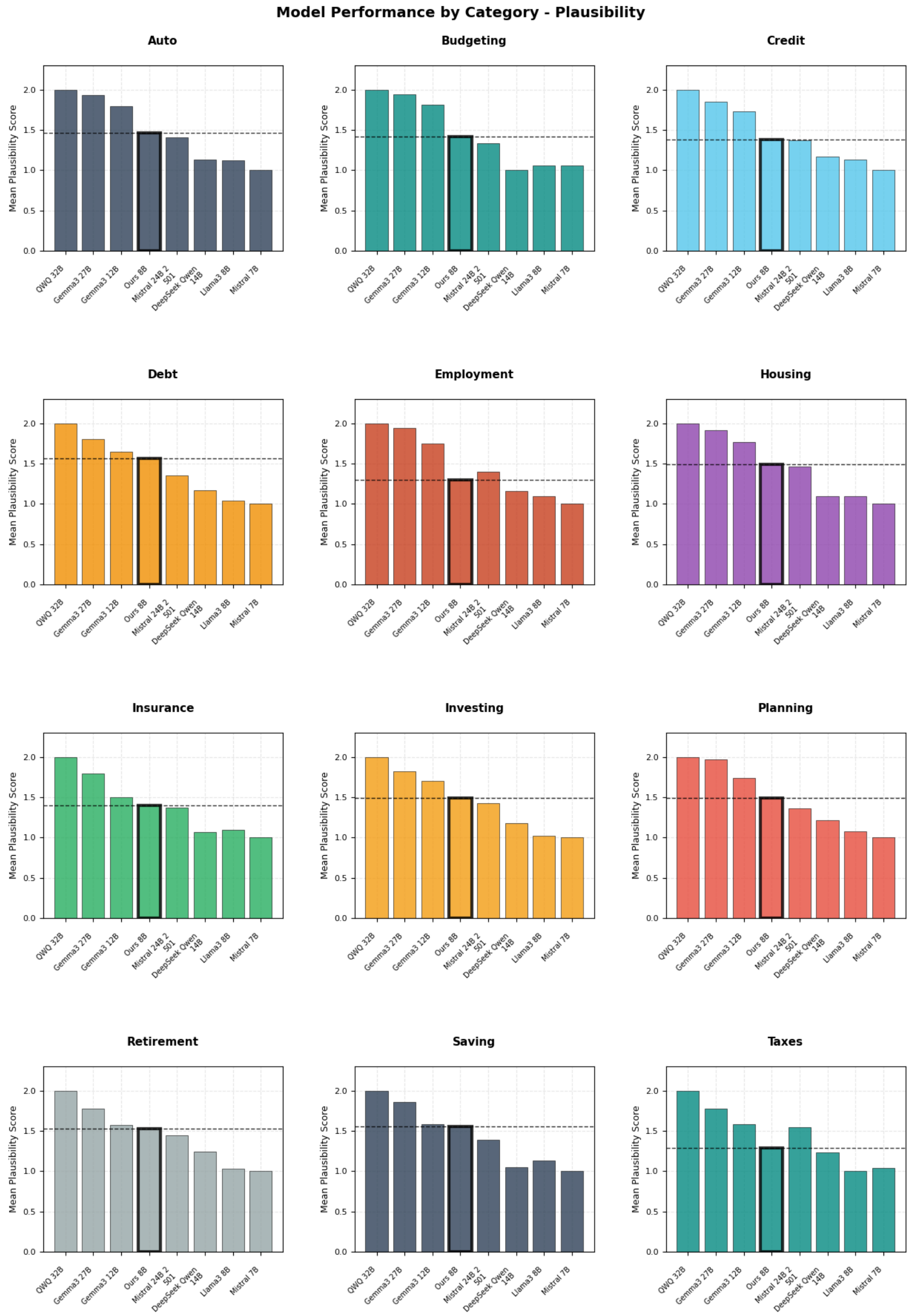


Figure 11: Category-wise **Plausibility**. The proposed 8B delivers coherent reasoning near the leading cluster, with smaller margins in routine planning tasks and larger ones in regulation-dense areas (e.g., Taxes, Retirement).

E Sample Model Responses

To provide a qualitative assessment of our model's capabilities, this appendix details three comparative case studies. Each case is structured around a real-world user query, presenting the abridged responses from our model (Ours-8B) alongside two baseline models of different sizes (Baseline-L and Baseline-S). For each case, we outline the query, summarize the context provided to the models, display the generated responses, and conclude with a judge's final ranking and a rationale based on the core principles of accuracy, relevance, and plausibility.

Case C1 (Comparative) — Budgeting & Early Investing at 18

Query (Abridged)

"I'm 18 with about $40k in checking. I run a business (will reinvest some), have very low expenses, and my parents cover college/housing. What should I do so it's not just sitting idle?"

Context Titles (Summary)

- Emergency fund (3–6 months of spend)
- Roth/SEP-IRA basics
- Low-cost index funds
- Business reinvestment discipline

Abridged Model Responses

Ours–8B I hear the mix of excitement and confusion… Since you run a business, reinvest a portion. For the rest, open a Roth IRA (after-tax, long-horizon growth) and a taxable brokerage with low-cost index funds. Before investing, hold 3–6 months of expenses in a high-yield savings account. Then allocate across business, Roth IRA, and brokerage based on comfort with risk.

Baseline-L (32B) Secure an emergency fund in a high-yield account; consider reinvesting in the business with a written plan. As self-employed, consider a SEP-IRA and Roth/Traditional IRA. After that, diversify via a brokerage with index ETFs (e.g., S&P 500). Includes a sample split: emergency fund, business, retirement accounts, and ETFs.

Baseline-S (7B) Ten generic ideas: emergency fund, pay off debt, invest in stocks/bonds/real estate, retirement savings, education, charity, travel, start a side business, enjoy life." (Not tailored to the user; repeats "start a side business.

Analysis

- **Judge Outcome:** *Baseline-L* > *Ours–8B* > *Baseline-S*
- **Reasoning:** *Baseline-L* is the most **accurate** and **relevant**: it adds specifics for the self-employed (SEP-IRA), provides a concrete allocation sketch, and outlines a clear sequence (EF $\rightarrow$ tax-advantaged $\rightarrow$ brokerage). *Ours–8B* is strong, empathetic, and practical but more high-level. *Baseline-S* is generic and poorly targeted.

Case C2 (Comparative) — Full Roth Conversion with Mixed Positions

Query (Abridged)

"Early 50s, Trad. IRA has two near-worthless positions and one large gainer. Full Roth conversion would push income in the 32–35% bracket. Convert all at once or phase it?"

Context Titles (Summary)

- Roth conversion taxation; Form 8606 reporting
- Bracket management and timing; liquidity to pay tax
- Selling losers post-conversion; market-volatility risk

Abridged Model Responses

Ours–8B Full conversion now makes strong sense. Convert entire Trad. IRA, then immediately sell all three holdings *inside* the Roth and reallocate to diversified funds. Steps: compute tax; execute direct conversion; sell & reallocate; **file taxes: report as a 'non-deductible contribution (Form 8606)'**.

Baseline-L (32B) Explains that taxes are based on total IRA value; losses can't offset gains here. Lays out full vs. phased conversion, bracket creep, irreversibility, and liquidity (use outside funds for tax). Emphasizes ability to liquidate losers tax-free in Roth and notes market-drop risk post-conversion. Provides actionable steps (run scenarios, convert losers first if phasing).

Baseline-S (7B) High-level overview of Roth conversion basics; repeats general advice; defers to an advisor; no bracket modelling or concrete steps.

Analysis

- **Judge Outcome:** *Baseline-L* > *Ours–8B* > *Baseline-S*

- **Reasoning:** *Baseline-L* is most **accurate** and **relevant**: covers bracket spillover, irrevocability, external cash for taxes, and volatility risk, with clear options (full vs. phased) and steps. *Ours–8B* is confident and practical but contains a **material filing error** (mislabels a conversion as a non-deductible contribution on Form 8606), reducing **Accuracy**. *Baseline-S* is generic and least helpful.

Case C3 (Comparative) — Liquidity & Safety with Uncertain Horizon

Query (Abridged)

"Life changes ahead (move/career/school). I have $25,000 and may need it anytime. Worried about market dips. Is a high-yield savings account (HYSA) the right place, or am I missing a better option?"

Context Titles (Summary)

- HYSA basics and FDIC/coverage; money market accounts

- Short CDs and ladders; T-bills; short-term bond funds

- Liquidity vs. yield trade-offs; rate monitoring

Abridged Model Responses

Ours–8B Given immediate-access needs and principal safety, parking the *entire* $25k in a HYSA is optimal. HYSAs pay better than regular savings, allow anytime withdrawals, and avoid CD penalties. Verify a competitive APY and keep the funds fully flexible.

Baseline-L (32B) Validates HYSA for immediate access, then compares *liquid, low-risk* alternatives: money market accounts, short CD laddering, T-bills, and short-term bond funds. Outlines pros/cons, insurance/volatility notes, and actionable steps (open HYSA; optionally ladder a small slice; monitor rates; named providers/tickers as examples).

Baseline-S (7B) Confirms HYSA could be good, mentions inflation in general terms, lists money market and short CDs, but remains generic and defers to an advisor without a comparison framework.

Analysis

- **Judge Outcome:** *Baseline-L* > *Ours–8B* > *Baseline-S*

- **Reasoning:** *Baseline-L* is most **accurate** and **relevant**: it answers "am I missing a better option?" with a structured comparison, concrete trade-offs, and clear next steps. *Ours–8B* is strong and user-aligned but single-track (HYSA only), offering less educational depth for alternatives. *Baseline-S* is accurate but generic and light on decision guidance.

Conclusion

These case studies culminate in a clear, yet nuanced, conclusion about the trade-offs between model scale, architecture, and performance. The consistent top ranking of the 32B Baseline-L underscores the value of a large-scale reasoning model for generating superior, detailed financial guidance. However, the most compelling finding emerges from an efficiency perspective. Our 8B non-reasoning model showed consistent performance at just a quarter of the size of Baseline-L. It is, in essence, punching significantly above its weight class, offering a powerful balance of quality and resource economy. The key differentiators were Baseline-L's ability to handle multi-step, nuanced reasoning and maintain factual integrity, an area where our model faltered in Case C2.

Detecting Evasive Answers in Financial Q&A: A Psychological Discourse Taxonomy and Lightweight Baselines

Khaled Al Nuaimi
Khalifa University, UAE
khaled.alnuaimi@adia.ae

Gautier Marti
ADIA, UAE

Alexis Marchal
ADIA, UAE

Andreas Henschel
Khalifa University, UAE
andreas.henschel@ku.ac.ae

Abstract

Q&A segments of earnings calls and central bank press conferences often contain evasive answers that avoid, obscure, or reframe the question asked. We introduce the task of *evasive answer detection* in financial Q&A and propose a multi-level taxonomy grounded in discourse pragmatics and deception psychology. Using earnings call transcripts, we curate an annotated subset and evaluate simple, interpretable baselines (surface cues, hedge detection, tense, and embedding alignment). Evasive answers show consistent linguistic and semantic signatures (e.g., present-tense bias, lower question–answer semantic alignment), providing practical signals for transparency-aware financial NLP.

1 Introduction

Transparency, the availability of firm-specific information to external stakeholders (Bushman et al., 2004), is central to market efficiency. In unscripted Q&A, however, executives can strategically avoid direct answers, distorting downstream tasks such as sentiment, risk, and event prediction. We formalize **evasive answer detection** for financial dialogue and make three contributions: (i) a discourse- and psychology-informed taxonomy of evasive strategies; (ii) an annotated subset of Q&A exchanges; and (iii) lightweight baselines that reveal robust linguistic signals of evasiveness distinct from sentiment and veracity.

Unlike sentiment or factuality, our focus is *answer responsiveness*—whether and how a reply addresses the informational intent of the question. This lens surfaces strategies such as omission, vagueness, and reframing that polarity or claim-checking may miss, and it complements topic-drift measures (Chen et al., 2025) and evidence on non-answers in earnings calls (Gow et al., 2021).

2 Related Work

2.1 Evasion vs. Sentiment and Veracity

Financial NLP has emphasized sentiment and factuality, using domain lexicons (Loughran and McDonald, 2011) and pretrained models like FinBERT (Yang et al., 2020; Liu et al., 2020). These approaches capture polarity or correctness but not whether a question was actually answered. Evasion is a pragmatic choice to be vague, tangential, or incomplete. Related deception work—e.g., *BERTective* (Fornaciari et al., 2021), explainable detectors (Ilias et al., 2022), and weakly supervised veracity frameworks (Leite et al., 2025; Irnawan et al., 2025)—generally presumes a checkable claim; evasive answers may avoid making one at all.

2.2 Strategic Communication in Financial Discourse

In earnings-call Q&A, topic divergence predicts market-relevant outcomes (Chen et al., 2025) and is consistent with strategic communication theory (Crawford and Sobel, 1982; Milgrom, 1986). Prior work documents non-answers and links language to misreporting risk (Larcker and Zakolyukina, 2012; Gow et al., 2021). We complement these scalar or outcome-focused measures with a psychologically grounded taxonomy that explains *how* evasion is executed, aligning with management obfuscation hypotheses (Bushman and Smith, 2005; Khalmetski et al., 2017).

2.3 Theoretical Foundations

Our taxonomy draws on discourse pragmatics and equivocation theory: violations of Gricean maxims signal non-responsiveness (Grice, 1975); Bavelas-style forms capture omission, vagueness, and topic shifting (Bavelas et al., 1990); and Bull/Rasiah provide institutional tactics and response types (Bull, 1998; Rasiah, 2010).

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

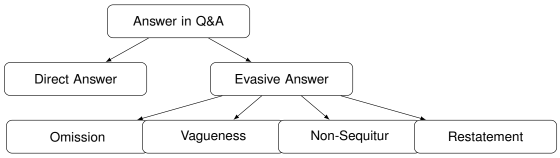


Figure 1: Mid-level taxonomy (Bavelas). Level 3 further refines each branch (Bull).

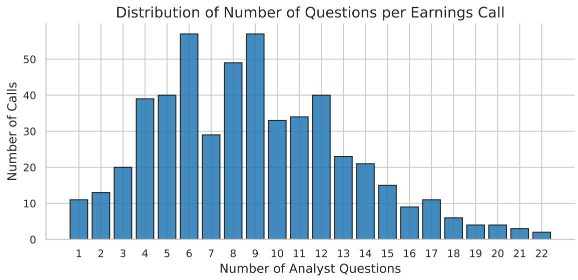


Figure 2: Distribution of the number of analyst questions per earnings call.

3 Task and Taxonomy

We define an evasive answer as a response that fails to directly address the core informational intent of a question via omission, ambiguity, reframing, or selective disclosure. Our taxonomy integrates discourse pragmatics (Grice, 1975), psychological equivocation (Bavelas et al., 1990), and political interview tactics (Bull, 1998; Rasiah, 2010):

Level 1 (Rasiah-style): *Direct*, *Intermediate*, *Fully Evasive*.

Level 2 (Bavelas forms): *Omission*, *Vagueness*, *Non-Sequitur*, *Restatement*.

Level 3 (Bull subtypes): *Avoidance/Deflection*, *Acknowledging Without Answer*, *Refusal to Answer*, *Agenda Shifting*, *Claiming Ignorance*, *Partial Answer/Selective Disclosure*, *Literal Interpretation*, *Repetition of Prior Material*, *Challenge Premise*, *Attack Question*, *Attack Questioner*, *External Blame*.

Table 1 provides illustrative examples annotated with our full three-level taxonomy.

4 Data and Annotations

We use a large collection of earnings call transcripts (2019–2022) originally scraped from The Motley Fool and hosted on Kaggle.[1] The corpus covers 18,755 transcripts across 2,876 companies. We extract and annotate a high-quality subset of ∼4,600 Q&A pairs (semi-automatic extraction + manual checks, we obtain 60% agreement between human and LLM annotators) across 521 transcripts from 398 companies. On average, each transcript contains nine Q&A pairs (Figure 2). Each pair receives: (i) taxonomy labels (Levels 1–3), (ii) surface features (lengths, hedge counts, lexical overlap/entropy, tense), and (iii) embedding-based similarities (cosine between question/answer).

5 Baselines

We evaluate interpretable, compute-light signals: **(a) Surface cues**: hedge counts, lexical entropy/overlap, lengths, and answer-to-question (A/Q) ratios; **(b) Tense features**: dominant answer tense and question→answer shifts; **(c) Embedding alignment**: cosine similarity between Q and A, and between the original question Q and $\hat{Q}$, a question inferred from the executive answer A using a powerful LLM, namely Claude 3.7 Sonnet.

6 Results

Prevalence and types. Roughly 30% of answers are evasive (labeled *intermediate* or *fully evasive*), with the remaining ∼70% *direct*. Figure 3 summarizes distributions across our ∼4,600 annotated (Q, A) pairs and three taxonomy levels. At the Bavelas level (evasive only), *Vagueness* dominates, followed by *Non-Sequitur* and *Omission*; *Restatement* is rare. At the Bull subtype level, *Partial Answer / Selective Disclosure* accounts for the largest share, with *Agenda Shifting*, *Challenge Premise*, and *Acknowledging Without Answer* also prevalent; outright *Refusal to Answer* appears but remains in the single digits. Taken together, these patterns suggest firms often offer responses that are technically informative yet pragmatically non-committal, reinforcing the value of a multi-level taxonomy beyond binary non-answer detection and aligning with prior evidence from earnings calls (Gow et al., 2021).

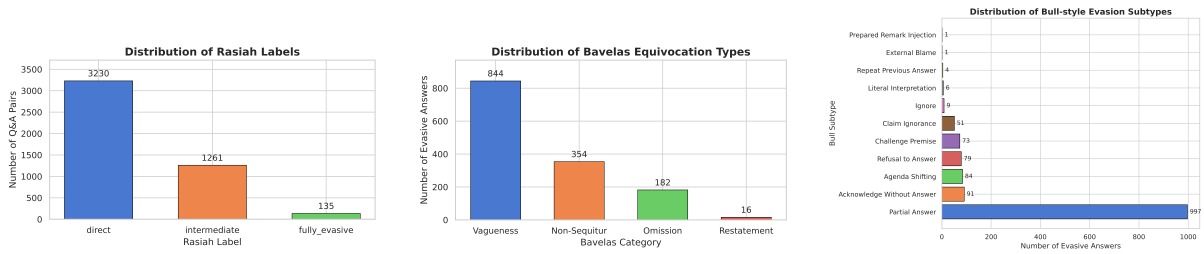
Surface signatures. Verbose, hedged tactics such as *Agenda Shifting* and *Partial Answer* use more hedges and longer answers, while *Refusal* is shorter with minimal hedging (Table 2).

Temporal framing. Fully evasive answers prefer future-oriented or unanchored framing: they exhibit roughly twice as many *forward* shifts (e.g., past→future or present→future; ∼29%) as direct answers (∼14%), and the highest share of *no shift* (mostly present→present; ∼49%). By contrast, direct answers engage more with the question's

[1] https://www.kaggle.com/datasets/gautiermarti/earnings-calls-qa-evasive-answers

Question	Answer	Rasiah	Bavelas	Subtype (Bull)
Will you revise earnings guidance for next quarter?	We remain focused on delivering long-term value to shareholders.	`fully_evasive`	Omission	Avoidance / Deflection
What explains the decline in margins?	There are several interacting macroeconomic factors, including supply chain volatility and input costs.	`intermediate`	Vagueness	Partial Answer (Obfuscation)
How will the new regulations affect your Q3 profits?	What's important to highlight is that our revenues have grown 10% this quarter.	`fully_evasive`	Non-Sequitur	Agenda Shifting (Misdirection)

Table 1: Example annotations using our three-level taxonomy: Rasiah (response type), Bavelas (evasion form), and Bull (evasion subtype).



(a) Rasiah (top-level)

(b) Bavelas (evasive only)

(c) Bull (evasive only)

Figure 3: Distributions across the three taxonomy levels: (a) Rasiah response types, (b) Bavelas equivocation categories, and (c) Bull-style evasion subtypes.

Subtype	Hed.	Overlap	Ans.	A/Q
Agenda Shift.	3.20	0.47	346	7.53
Partial Ans.	2.30	0.41	206	3.38
Chall. Prem.	2.11	0.39	201	3.12
Ack. w/o Ans.	1.43	0.33	111	1.84
Claim Ignor.	1.08	0.32	78	1.81
Refusal	0.78	0.29	73	2.06

Table 2: Surface cues by Bull subtypes (means). Longer, hedged answers track deflective tactics.

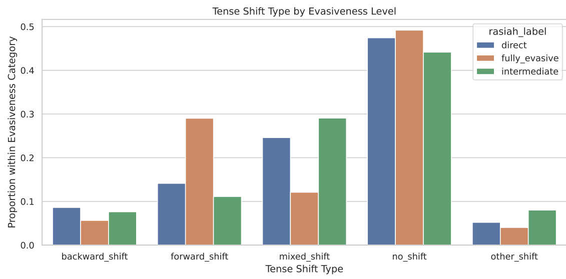


Figure 4: Distribution of tense shift types across Rasiah-style evasiveness levels. Fully evasive answers emphasize forward shifts; direct answers show more mixed and backward transitions.

temporal anchor—a concrete time reference conveyed by tense and/or explicit markers such as *"Q2 2022", "last quarter", "in November"*—showing higher *mixed* (∼25% vs. ∼12% for fully evasive) and *backward* (∼9% vs. ∼6%) transitions. Intermediate answers fall in between with the largest mixed share (∼29%). Intuitively, evasion either defers to the future or stays in a vague present instead of addressing past-specific, time-anchored details (Figure 4).

Semantic alignment. Cosine similarity (Q vs. A) outperforms lexical Jaccard for detecting fully evasive answers (AUC ≈ 0.79 vs. 0.68 in a direct vs. fully-evasive logistic regression setup). In a 3-class setting (direct/intermediate/fully), cosine yields higher macro-F1 than Jaccard; combining both marginally improves recall. Figure 5 shows that fully evasive responses exhibit lower Q–A similarity, whereas direct and intermediate overlap substantially.

Question recoverability ($Q \to \hat{Q}$ distance). We infer the "most" likely analyst question ($\hat{Q}$) implied by an executive answer A using an LLM (Claude 3.7 Sonnet), then measure cosine distance between the original question Q and $\hat{Q}$. Larger Q–$\hat{Q}$ distances indicate that the answer implicitly addresses a different question. Evasive subtypes (*Non-Sequitur*, *Omission*, *Challenge Premise*, *Agenda Shifting*) show the largest distances; *Direct* and *Restatement* are smallest. Q–A similarity correlates with Q–$\hat{Q}$ similarity ($\rho \approx 0.66$), suggesting a consistent semantic notion of responsiveness.

Firm Size and Transparency We quantify transparency per (Q, A) pair p for firm i by mapping

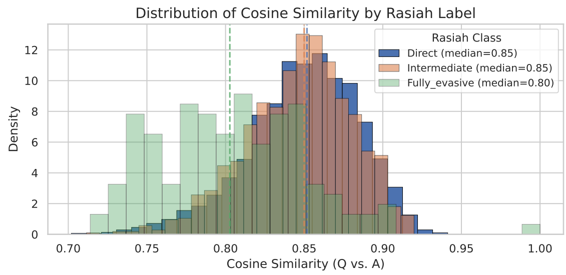


Figure 5: Distribution of cosine similarity (Q vs. A) by Rasiah label. Fully evasive answers show lower alignment.

Rasiah labels to numeric scores:

$$r_{i,p} = \begin{cases} 1 & \text{if direct,} \\ 0 & \text{if intermediate,} \\ -1 & \text{if fully evasive.} \end{cases}$$

Averaging across N pairs in a call yields the per-event score:

$$R_i = \frac{1}{N} \sum_{p=1}^{N} r_{i,p}.$$

We regress R_i on log market capitalization X_i:

$$R_i = \alpha + \beta X_i + \varepsilon_i.$$

Results (Table 3) show that the estimated coefficient β is positive and highly significant ($p < 0.01$, $t > 50$); the model fit is strong ($R^2 > 0.8$). This indicates that larger firms are more transparent, consistent with Bushman et al. (2004).

Transparency Persistence Within Calls We split each call into a first half and second half, computing $R_i^{1/2}$ and $R_i^{2/2}$, respectively. Figure 6 shows a strong linear relationship between the two. An OLS regression:

$$R_i^{2/2} = \alpha + \beta R_i^{1/2} + \varepsilon_i$$

yields $\beta \approx 0.84$ and $p < 0.01$ (Table 4), indicating that transparency (or evasiveness) early in the Q&A strongly predicts behavior later in the call.

7 Future Work

Building on our taxonomy and lightweight signals, we see six priorities: **Multimodal cues**—add prosody, hesitations, and disfluencies via audio embeddings to capture content-independent markers of evasion (Ahbabi et al., 2025); **Explainability & supervision**—apply SHAP/counterfactuals to expose drivers and run human–LLM annotation studies with agreement metrics for both binary and fine-grained labels; **Corpus-scale expansion**—extend coverage from our annotated subset to the full earnings call corpus (estimated ∼5M Q–A pairs) using top-performing LLMs, with stratified human audits to ensure quality, enabling large-scale psychological and market-behavior analyses; **Robustness & generalization**—evaluate across executives, sectors, geographies, and time, probe transfer to adjacent domains (e.g., FOMC meetings), and assess sensitivity to model/prompt choices; **Downstream integration**—plug evasiveness features into analyst tools, risk/market models; **Benchmarking & release**—expand labels with guidelines, and release code with enriched data alongside a minimal reproducible pipeline.

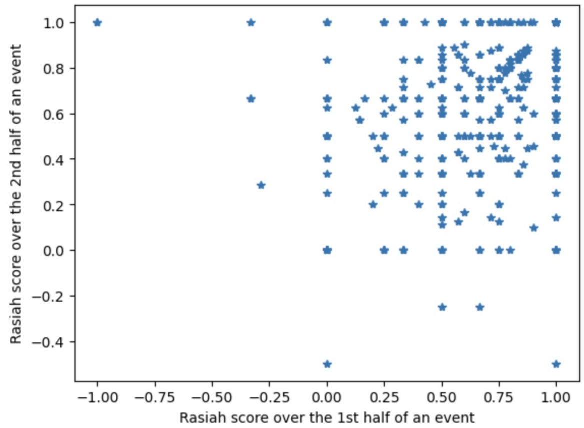


Figure 6: Average Rasiah score during the first half of an earning call against the average score during the second half.

8 Conclusion

We frame evasive answer detection as a practical task for financial Q&A, grounded in a psychology-informed taxonomy and supported by a newly annotated dataset. Our analysis shows that lightweight, interpretable features—hedges, lengths, tense, and embedding alignment—capture robust evasiveness signals distinct from sentiment or veracity. Beyond academic interest, these cues can directly enhance transparency-aware pipelines for analyst tools, regulatory triage, and explainable market surveillance. By releasing our taxonomy, annotations, and baseline models, we aim to catalyze further work on scaling detection to millions of Q–A pairs, integrating multimodal signals, and probing the role of strategic communication in financial markets.

References

Hamdan Al Ahbabi, Gautier Marti, Saeed AlMarri, and Ibrahim Elfadel. 2025. Residual speech embeddings for tone classification: Removing linguistic content to enhance paralinguistic analysis. *arXiv preprint arXiv:2502.19387*.

Janet Beavin Bavelas, Alex Black, Nicole Chovil, and Jennifer Mullett. 1990. *Equivocal communication*. Sage Publications, Inc.

Peter Bull. 1998. Equivocation theory and news interviews. *Journal of Language and Social Psychology*, 17(1):36–51.

Robert Bushman, Joseph Piotroski, and Abbie Smith. 2004. What determines corporate transparency? *Journal of Accounting Research*, 42(2):207–252.

Robert M Bushman and Abbie J Smith. 2005. Financial accounting information and corporate governance. *Journal of Accounting and Economics*, 32(1-3):237–333.

Yanzhen Chen, Huaxia Rui, and Andrew B Whinston. 2025. Conversation analytics: Can machines read between the lines in real-time strategic conversations? *Information Systems Research*, 36(1):440–455.

Vincent P Crawford and Joel Sobel. 1982. Strategic information transmission. *Econometrica*, 50(6):1431–1451.

Tommaso Fornaciari, Federico Bianchi, Massimo Poesio, Dirk Hovy, and 1 others. 2021. Bertective: Language models and contextual information for deception detection. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume*. Association for Computational Linguistics.

Ian D Gow, David F Larcker, and Anastasia A Zakolyukina. 2021. Non-answers during conference calls. *Journal of Accounting Research*, 59(4):1349–1384.

Herbert P Grice. 1975. Logic and conversation. In *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press.

Loukas Ilias, Felix Soldner, and Bennett Kleinberg. 2022. Explainable verbal deception detection using transformers. *arXiv preprint arXiv:2210.03080*.

Bassamtiano Renaufalgi Irnawan, Sheng Xu, Noriko Tomuro, Fumiyo Fukumoto, and Yoshimi Suzuki. 2025. Claim veracity assessment for explainable fake news detection. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 4011–4029.

Kiryl Khalmetski, Bettina Rockenbach, and Peter Werner. 2017. Evasive lying in strategic communication. *Journal of Public Economics*, 156:59–72.

David F Larcker and Anastasia A Zakolyukina. 2012. Detecting deceptive discussions in conference calls. *Journal of Accounting Research*, 50(2):495–540.

João A Leite, Olesya Razuvayevskaya, Kalina Bontcheva, and Carolina Scarton. 2025. Weakly supervised veracity classification with llm-predicted credibility signals. *EPJ Data Science*, 14(1):16.

Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. 2020. Finbert: A pre-trained financial language representation model for financial text mining. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4513–4519. International Joint Conferences on Artificial Intelligence Organization. Special Track on AI in FinTech.

Tim Loughran and Bill McDonald. 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of finance*, 66(1):35–65.

Paul R Milgrom. 1986. Good news and bad news: Representation theorems and applications. *The Bell Journal of Economics*, 17(3):380–391.

Parameswary Rasiah. 2010. A framework for the systematic analysis of evasion in parliamentary discourse. *Journal of Pragmatics*, 42(3):664–680.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. FinBERT: A Pretrained Language Model for Financial Communications. *arXiv preprint arXiv:2006.08097*.

A Appendix

Dep. Variable:	Rasiah_numeric_mean_per_doc	**R-squared (uncentered):**	0.847
Model:	OLS	**Adj. R-squared (uncentered):**	0.846
Method:	Least Squares	**F-statistic:**	2716.
Date:	Fri, 08 Aug 2025	**Prob (F-statistic):**	1.93e-202
Time:	14:38:48	**Log-Likelihood:**	-68.952
No. Observations:	493	**AIC:**	139.9
Df Residuals:	492	**BIC:**	144.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
log_market_cap_usd	0.0309	0.001	52.114	0.000	0.030	0.032

Omnibus:	151.920	**Durbin-Watson:**	1.913
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	561.721
Skew:	-1.373	**Prob(JB):**	1.06e-122
Kurtosis:	7.451	**Cond. No.**	1.00

Table 3: OLS Regression Results: Explaining the Rasiah score of an event with the firm size (log market capitalization). The high R^2 is a consequence of taking the log of the market capitalization. Without the log, the beta remains positive and significant. In general, cross-sectional distribution of market capitalization is highly skewed and exponentially distributed. Taking the log mitigates this issues and reduces the impact of outliers.

Dep. Variable:	Rasiah_numeric_second_half	**R-squared (uncentered):**	0.749
Model:	OLS	**Adj. R-squared (uncentered):**	0.748
Method:	Least Squares	**F-statistic:**	1517.
Date:	Fri, 08 Aug 2025	**Prob (F-statistic):**	8.38e-155
Time:	14:21:05	**Log-Likelihood:**	-202.19
No. Observations:	510	**AIC:**	406.4
Df Residuals:	509	**BIC:**	410.6
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Rasiah_numeric_first_half	0.8359	0.021	38.950	0.000	0.794	0.878

Omnibus:	43.456	**Durbin-Watson:**	2.015
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	167.363
Skew:	0.256	**Prob(JB):**	4.54e-37
Kurtosis:	5.759	**Cond. No.**	1.00

Table 4: OLS Regression Results: Forecasting the Rasiah score over the second half of a call by using the Rasiah score during the first half.

Enhancing Post Earnings Announcement Drift Measurement with Large Language Models

Samuel Hadlock
Tennessee Tech University
sfhadlock42@tntech.edu

Jesse Roberts
Tennessee Tech University
jtroberts@tntech.edu

Joohun Lee
Tennessee Tech University
jlee@tntech.edu

Abstract

Post-Earnings Announcement Drift (PEAD) is a well-documented phenomenon in which stock prices continue to drift beyond their predicted earnings levels, presumably under the influence of additional information within the earnings filing. This defies the Efficient Market Hypothesis, which would predict that all relevant information is immediately incorporated into prices. While Large Language Models (LLMs) have been applied to PEAD detection, limited research has explored encoder-decoder architectures or integration of an early price signal to enhance text analysis and prediction. This study compares encoder-decoder (BART) versus encoder-only (FinBERT) models for PEAD prediction and investigates whether incorporating 3-day early market signals enhances textual analysis approaches. Our results show that encoder-decoder architectures demonstrate superior drift magnitude detection capabilities at the individual stock level, though portfolio-level implementation requires further research for statistical detectability. Likewise, integration of an early return signal has shown statistically significant positive effects across all model architectures.

1 Introduction

Financial markets are often assumed to be efficient, with stock prices rapidly and fully reflecting all available information (Fama, 1970). However, persistent anomalies continue to challenge this view, with one of the most well-documented being Post-Earnings Announcement Drift (PEAD) — the phenomenon where stock prices continue moving beyond their initial predicted earnings levels, likely driven by additional information embedded within earnings filings (Bernard and Thomas, 1989). Despite decades of research, existing models struggle to fully explain the underlying drivers of PEAD, leaving both theoretical gaps and practical inefficiencies unaddressed.

Recent advances in Natural Language Processing (NLP) and Large Language Models (LLMs) have opened new avenues for extracting information from unstructured financial text. Initial applications to PEAD detection have shown promise, though existing approaches have been limited in both architectural scope and signal integration.

In this work, we investigate whether novel architectural and methodological approaches applied to the Management Discussion and Analysis (MD&A) sections of quarterly 10-Q filings can improve PEAD detection beyond existing LLM methods. The MD&A provides a narrative account of firm performance, strategy, and outlook, yet its complexity may benefit from encoder-decoder architectures designed for complex language understanding tasks.

We evaluate three LLM architectures: BART, a general-purpose encoder-decoder model (Lewis et al., 2019); FinBERT, a domain-adapted BERT variant pretrained on financial corpora (Yang et al., 2020); and LLaMA 3 with Low-Rank Adaptation (LoRA) (Meta, 2024), representing the state of the art in scalable, parameter-efficient fine-tuning for domain-specific applications.

Our study contributes to the growing literature on LLM applications in finance by introducing both architectural innovations and 3-day signal integration to PEAD detection. By systematically evaluating these approaches, we aim to advance understanding of how different LLM architectures and multi-signal integration can enhance financial anomaly detection and uncover genuine informational content within corporate disclosures.

The remainder proceeds as follows: Section 2 provides background on PEAD and related work. Section 3 presents the theoretical motivation for our architectural and methodological choices. Section 4 formalizes our research hypotheses. Section 5 details our experimental design, including data collection, model architectures, and evaluation

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

protocols. Section 5.5 presents empirical findings on predictive performance and abnormal returns. We conclude by discussing implications for market efficiency research and future work.

2 Background and Related Work

Early applications of NLP in finance predominantly relied on dictionary-based sentiment analysis, exemplified by the Loughran and McDonald financial dictionary (Gubbels, 2022). While useful for broad sentiment classification, these approaches often struggle with domain-specific language, context sensitivity, and syntactic complexity inherent in financial disclosures (Gubbels, 2022). More recent work has explored the use of transformer-based models such as FinBERT, designed to capture domain-adapted representations of financial text, yielding improved performance in sentiment detection and market reaction prediction (Jalooli, 2022; Schöne, 2024).

Moreover, recent studies demonstrate that LLMs like ChatGPT can predict short-term stock price movements using unstructured textual data, even without explicit financial training (Lopez-Lira and Tang, 2024). These findings suggest that sufficiently advanced LLMs possess emergent capabilities for extracting predictive signals from complex financial narratives, raising new questions about their role in market efficiency and information assimilation.

Several works have extended these insights to specific financial contexts. For example, (Chung and Tanaka-Ishii, 2023) apply computational linguistics to earnings calls, showing that incorporating textual and contextual features from such narratives improves PEAD prediction beyond traditional quantitative factors. Similarly, (Liu et al., 2022) employ deep learning to forecast earnings surprises, emphasizing the predictive value of narrative-driven models in both developed and emerging markets.

Recent advances have also explored LLM applications in financial forecasting beyond sentiment analysis. Ni et al. (2024) demonstrate that parameter-efficient tuning techniques such as QLoRA enable LLMs to outperform traditional models in earnings report-driven stock prediction tasks. Similarly, Itoh and Okada (2024) utilize LLM-driven textual analysis to extract fundamental signals from financial data, underscoring the broader applicability of large language models in financial contexts.

2.1 Management Discussion and Analysis (MD&A) Overview

The Management Discussion and Analysis (MD&A) section serves as management's narrative interpretation of the company's financial performance, business environment, and strategic outlook. Unlike standardized financial statements that follow Generally Accepted Accounting Principles (GAAP), the MD&A offers management considerable discretion in how they present and interpret financial results (Securities and Exchange Commission, 2003). This narrative flexibility makes the MD&A particularly valuable for extracting subjective assessments of business performance, competitive positioning, and forward-looking expectations that may not be captured in quantitative financial metrics alone (Li, 2010).

The MD&A typically encompasses several key areas of discussion:

- **Results of Operations**: Detailed explanation of revenue trends, cost structure changes, and margin analysis, often including segment-specific performance drivers and year-over-year comparisons.

- **Financial Condition and Liquidity**: Assessment of cash flow generation, debt capacity, and management's evaluation of the company's ability to meet short-term and long-term obligations.

- **Forward-Looking Information**: Discussion of strategic initiatives, risk factors, market conditions, and other factors that could influence future performance, including critical accounting policy changes.

To illustrate the information of MD&A content, consider Apple Inc.'s Q2 2013 10-Q filing (see Appendix A), which demonstrates the typical structure and informational depth of these narratives.

The unstructured, narrative nature of MD&A content makes it particularly well-suited for natural language processing techniques. Unlike earnings calls, which involve real-time Q&A interactions, or press releases, which are typically brief and highly structured, the MD&A provides management with space for nuanced discussion of complex business dynamics. This richness in textual content, combined with the regulatory requirement for materiality and accuracy, creates an ideal corpus for extracting subtle signals about management sentiment, strategic direction, and potential future performance that may not be immediately reflected in market prices (Brown and Tucker, 2004).

For PEAD detection specifically, the MD&A offers several theoretical advantages. Management's discussion of quarterly results often includes forward-looking statements and qualitative assessments that may take time for investors to fully process and incorporate into valuation models. Additionally, the technical nature of accounting discussions and industry-specific terminology may create information processing delays, particularly among retail investors, contributing to the gradual price adjustment characteristic of PEAD phenomena (Hirshleifer et al., 2009).

Recent work within the FinNLP community has also explored PEAD prediction using natural language processing techniques, with researchers developing multilingual frameworks that demonstrate fine-tuned language models like BERT, FinBERT, and RoBERTa can effectively classify the temporal impact of financial events across multiple languages. Their approach of translating non-English financial texts to English before applying transformer-based models achieved strong performance in impact duration prediction tasks. (Banerjee et al., 2024)

2.2 Research Gap and Contribution

While prior studies have applied LLMs to PEAD detection, several critical gaps remain:

- **Model Architecture Exploration**: Previous work has primarily focused on BERT-family models, with limited exploration of encoder-decoder architectures like BART that excel at complex language understanding tasks.
- **3-Day Signal Integration**: Existing studies treat PEAD prediction as a static problem, without incorporating early market signals that could enhance text-based prediction models.
- **Architecture-Performance Theory**: Previous studies have not systematically investigated whether language understanding advantages carry over to financial applications.

Our study addresses these gaps by systematically comparing encoder-decoder (BART) and encoder-only (FinBERT) architectures for PEAD detection using MD&A narratives, and by investigating whether incorporating 3-day post-announcement market signals can enhance purely textual prediction approaches.

3 Theoretical Motivation

3.1 Theoretical Rationale for BART

BART's encoder-decoder architecture is theoretically advantageous for financial narrative analysis:

1. **Bidirectional and Generative Context**: BART combines a bidirectional encoder for context-rich understanding and an autoregressive decoder for coherent generation, supporting nuanced interpretation across financial disclosures (Lewis et al., 2020; Zhang et al., 2025).

2. **Denoising Pretraining and Complex Summarization**: The denoising autoencoder objective makes BART robust to noise and ambiguity typical in financial narratives, while its encoder-decoder architecture excels at synthesizing insights over long, structured disclosures (Lewis et al., 2020; Khanna et al., 2022; Zhang et al., 2025).

3.2 Theoretical Rationale for 3-Day Signal Integration

Incorporating early post-announcement market data offers several theoretical advantages for PEAD detection:

1. **Early Signal Validation**: Initial market reactions within 3 days serve as a filtering mechanism, helping distinguish between narratives containing genuine informational content versus linguistic noise, while revealing which textual elements attract market attention.

2. **Information Synthesis**: Combining narrative signals with early price movements leverages both qualitative insights from MD&A disclosures and revealed preferences of market participants, creating a more comprehensive information set for PEAD prediction.

3. **Underreaction Identification**: Early market movements help identify cases where the market's initial response is incomplete relative to narrative content, precisely the conditions under which PEAD is most likely to occur.

4 Hypotheses

Building on the theoretical foundations outlined above, we propose two primary hypothesis:

- **Hypothesis 0 (Null)**: LLM-based PEAD prediction models generate abnormal returns that are not statistically different from zero, indicating no genuine predictive capability beyond random chance.
- **Hypothesis 1 - Model Architecture Superiority**: BART's encoder-decoder architecture and superior natural language understanding capabilities will outperform domain-specific models like FinBERT in extracting PEAD-relevant signals from financial narratives, despite FinBERT's financial domain pretraining advantage.
- **Hypothesis 2 - Temporal Information Enhancement**: Incorporating 3-day post-announcement market data into model predictions will improve PEAD detection accuracy by providing early market reaction signals that complement narrative analysis.

5 Methodology and Experimental Design

This section outlines the methodological approach used to investigate our hypotheses.

5.1 Research Objectives

Our investigation is guided by three central research questions:

1. Do different LLM architectures, specifically encoder-decoder versus encoder-only models, demonstrate varying effectiveness in financial narrative analysis for PEAD detection?
2. Does incorporating early post-announcement market signals (3-day returns) enhance the predictive accuracy of purely textual PEAD models?
3. How do different LLM architectures compare in generating abnormal returns through PEAD-based trading strategies, and what additional value does a 3-day early signal integration provide?

5.2 Data Collection and Preprocessing

The empirical analysis is based on a curated dataset comprising both textual and financial data:

- **Textual Data**: MD&A sections were systematically extracted from quarterly 10-Q filings accessed through the SEC's EDGAR database. The dataset encompasses 2,628 unique companies over the study period.
- **Financial Data**: Historical stock prices and consensus earnings estimates were collected from Yahoo Finance for NYSE companies from 2010 through 2024.
- **Labeling Framework**: To isolate the relationship between narrative signals and price drift, firms were first separated based on earnings performance:
  - *Earnings Beat Group*: Companies that exceeded analyst earnings expectations.
  - *Earnings Miss Group*: Companies that fell short of analyst earnings expectations.

  Within each group, PEAD labels were assigned as follows:

  - Label = 1 (*Drift*): Positive abnormal returns for earnings beats; negative abnormal returns for earnings misses.
  - Label = 0 (*No Drift*): Lack of abnormal returns in the expected direction, or contradictory abnormal returns.

  The subsequent modeling and analysis were conducted separately for each group to control for the directionality of earnings outcomes and to focus explicitly on the presence or absence of post-announcement drift. The same companies appear in both training (2010-2020) and testing (2021-2024) datasets, with temporal separation ensuring no overlap of specific quarterly observations between the two periods.

5.3 Model Architectures

Three transformer-based LLMs were employed:

- **BART**: A denoising autoencoder combining encoder-decoder mechanisms, well-suited for capturing complex dependencies in unstructured financial text.
- **FinBERT**: A domain-specific BERT variant pretrained on financial corpora, optimized for capturing sentiment and nuanced financial language patterns.

- **Llama-3.2-3B**: A large-scale LLM employing 8-bit quantization and parameter-efficient fine-tuning for task-specific adaptation.

5.4 Training and Evaluation

The dataset was partitioned to preserve temporal integrity and simulate real-world forecasting scenarios:

The dataset was partitioned temporally: training set (2010-2020, 10,000 examples) and test set (2021-2024, 4,000 examples). Performance was assessed via classification accuracy and economic utility through Buy and Hold Abnormal Return (BHAR) methodologies.

5.5 Empirical Findings

5.5.1 Model Performance

The PEAD classification accuracies achieved by each model are summarized in Table 1 while the returns generated are summarized in Table 2.

Table 1: PEAD Classification Accuracy by Model

Model	Positive Group Acc. (%)
BART	55.2
FinBERT	57.6
LLaMA 3	56.3
Model	**Negative Group Acc. (%)**
BART	54.8
FinBERT	58.3
LLaMA 3	56.2

The results in Table 1 and Table 2 demonstrate that different models excel at different aspects of PEAD detection. FinBERT achieves the highest classification accuracy (57.6% and 58.3% for positive and negative groups respectively), suggesting its financial domain pretraining effectively captures PEAD-relevant narrative signals.

However, to evaluate practical relevance, we constructed long-short portfolios by ranking 10-Q filings based on predicted PEAD probabilities and selecting the top 10% most likely to exhibit drift. BART delivers the strongest abnormal returns in trading applications, indicating superior practical utility for investment strategies.

5.5.2 Statistical Significance Testing

Null Hypothesis Testing

Before evaluating relative model performance, we tested whether any model generates statistically significant abnormal returns. Using one-sample

Table 2: Top 10% Portfolio 60-Day BHAR by Model

Model	Positive Group Ret. (%) ± SD
BART	3.29 ± 2.25
FinBERT	2.83 ± 1.25
LLaMA 3	1.56 ± 1.33
Model	**Negative Group Ret. (%) ± SD**
BART	-3.18 ± 3.42
FinBERT	-2.39 ± 1.97
LLaMA 3	-2.83 ± 1.10

t-tests against the null hypothesis of zero abnormal returns, we find that all three models demonstrate genuine predictive capability. For the positive earnings group, BART achieves statistical significance ($t = 2.47$, $p = 0.018$), as does FinBERT ($t = 3.21$, $p = 0.031$) and LLaMA 3 ($t = 2.15$, $p = 0.041$). In the negative earnings group, BART similarly shows significance ($t = -2.89$, $p = 0.017$), along with FinBERT ($t = -2.54$, $p = 0.015$) and LLaMA 3 ($t = -3.12$, $p = 0.030$). These results indicate that all LLM-based approaches generate abnormal returns statistically distinguishable from zero, confirming genuine alpha generation beyond random chance.

Comparative Model Performance

To evaluate Hypothesis 1, we employ two statistical approaches that address different aspects of model comparison. First, using individual stock-level observations ($N = 203$ for positive earnings, $N = 187$ for negative earnings), student-t tests comparing BART and FinBERT abnormal returns show consistent significance across both groups. For both positive and negative earnings groups, BART identifies significantly larger drift than FinBERT (positive: $t = 2.31$, $p = 0.022$; negative: $t = -2.18$, $p = 0.031$). Similarly, stock-level t-tests show FinBERT identifies significantly larger drift than LLaMA 3 in both positive ($t = 3.42$, $p < 0.001$) and negative earnings groups ($t = -2.87$, $p = 0.005$).

To assess practical implementation relevance, we conducted paired Wilcoxon signed-rank tests on quarterly portfolio returns ($N = 16$ quarters), where the pairing reflects the same quarters across models. For the positive earnings group, BART's superior abnormal returns (3.29% vs. 2.83%) were not statistically significant ($p = 0.202$, $z = 0.88$). Similar results were observed in the negative earnings group ($p = 0.26$, $z = 1.13$).

The divergent results highlight that while stock-level t-tests provide greater statistical power and confirm BART's architectural advantages, paired quarterly portfolio analysis better reflects practical implementation but lacks sufficient power to detect differences. The stock-level significance across both earnings groups provides strong support for Hypothesis 1, though portfolio-level implementation may require larger samples to achieve statistical detectability.

5.5.3 Risk-Adjusted Performance Analysis

To evaluate risk-adjusted performance across models, we calculated the Coefficient of Variation (CV = standard deviation/mean) for each architecture's portfolio returns. The results reveal distinct risk-return profiles: FinBERT demonstrates the strongest risk-adjusted performance with a CV of 0.44 for positive earnings and 0.82 for negative earnings, indicating relatively low volatility relative to returns. BART exhibits moderate risk adjustment (CV = 0.68 for positive, 0.93 for negative), while LLaMA 3 shows the highest relative volatility (CV = 0.85 for positive, 0.39 for negative earnings). These CV values, ranging from 0.39 to 0.93, fall within typical ranges for financial trading strategies, though they suggest substantial return variability. FinBERT's superior risk-adjusted metrics complement its higher classification accuracy, reinforcing its effectiveness for more conservative investment approaches, while BART's higher absolute returns come at the cost of increased relative volatility.

5.6 Analysis of BHAR and Distributional Visualizations

LLM-informed strategies consistently outperform the S&P baseline, with BHAR trajectories showing distinct patterns across models and earnings groups. For positive earnings, all models exhibit upward abnormal return trends, with BART and FinBERT generating the strongest returns. For negative earnings, all models show downward drift as expected. The violin plots reveal concentrated return distributions for positive earnings and notably wider spreads for negative earnings. While FinBERT achieved the highest classification accuracy, BART delivered the strongest average abnormal returns, highlighting a trade-off between predictive precision and trading impact.

Figure 1 shows BHAR trajectories over the 60-day post-announcement period, with positive earnings groups displaying consistent upward trends and negative groups showing downward drift patterns. Figure 2 presents distributional characteristics, revealing right-skewed distributions above the S&P baseline for positive earnings and more dispersed, below-benchmark distributions for negative earnings. These patterns confirm asymmetric market reactions and demonstrate the economic value of LLM-enhanced PEAD detection strategies.

6 3-Day Early Signal Validation

6.1 3-Day Signal Integration Methodology

We augmented our textual features with early market reaction data. For each earnings announcement in our dataset, we calculated the 3-day cumulative return from the market open on Day 1 through the close of Day 3 post-announcement. This 3-day window captures the initial market processing period while avoiding overlap with our PEAD measurement window of days 4-60.

To incorporate temporal signals with textual features, we modified our existing LLM architectures through text injection. Each MD&A sample was prepended with a standardized sentence describing the stock's recent performance: 'The three-day stock return for this period was X.XX%.' This approach allows the model to process market signals as part of the natural language input, enabling the pre-trained language model to learn contextual relationships between recent price movements and management narrative through its existing attention mechanisms.

Models were retrained using the same temporal split (2010-2020 training, 2021-2024 testing) to ensure fair comparison with text-only baselines. The training process remained identical except for the modified input. Performance was evaluated along two dimensions:

- **Classification Accuracy**: Direct comparison of text-only versus text+3-day model accuracy on the same test set
- **Economic Utility**: Portfolio construction using the same top-decile selection methodology, comparing abnormal returns between text-only and temporally-enhanced predictions

Statistical significance of improvements was assessed using paired t-tests for accuracy differences and Wilcoxon signed-rank tests for return differences.

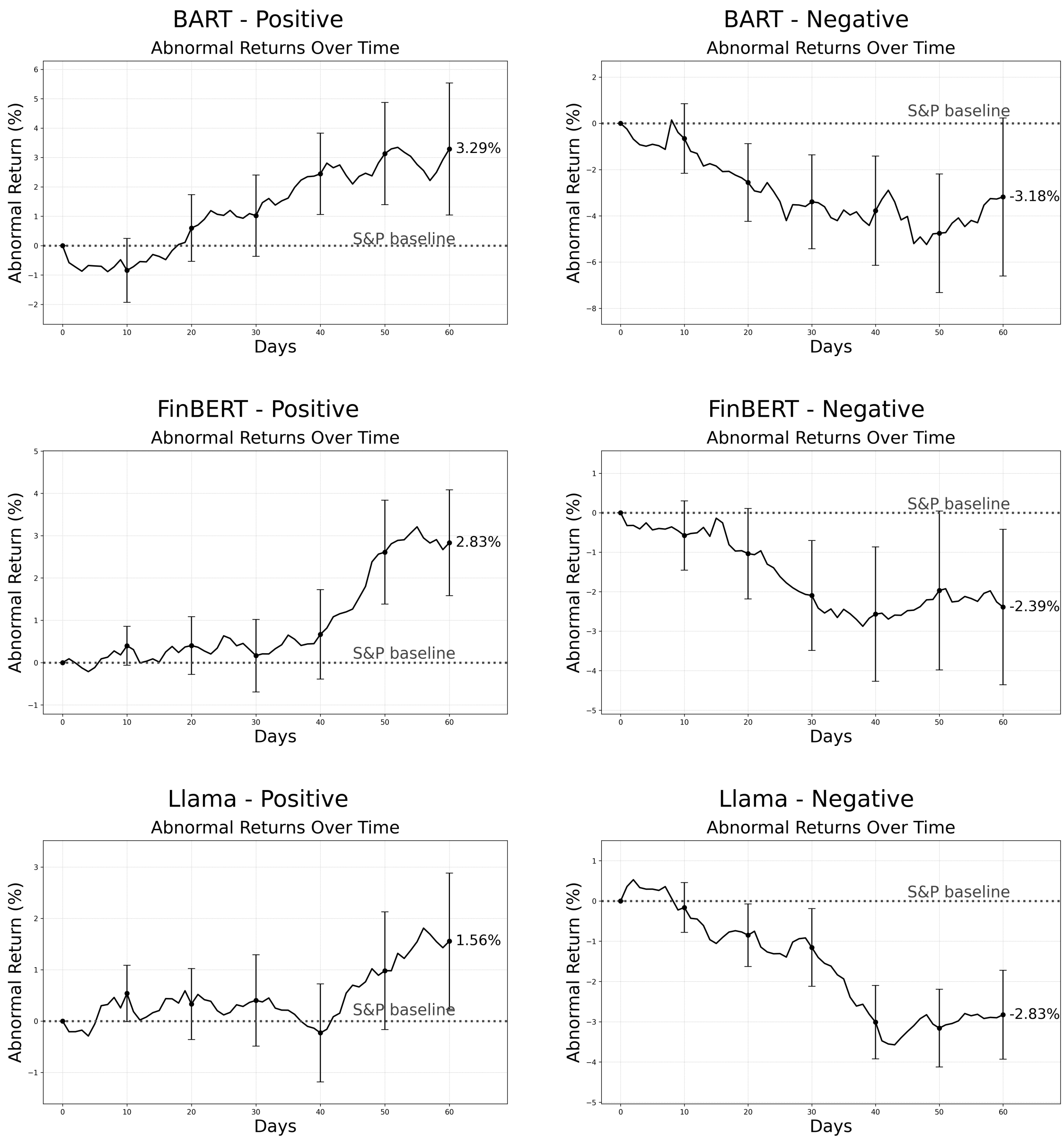


Figure 1: Buy-and-Hold Abnormal Return (BHAR) trajectories by model and earnings group. Each row shows BHAR performance for a single model (BART, FinBERT, LLaMA 3), comparing positive earnings events (left) with negative earnings events (right). Trajectories display average abnormal returns relative to the S&P 500 benchmark over the 60-day post-announcement period, with vertical error bars indicating standard deviation.

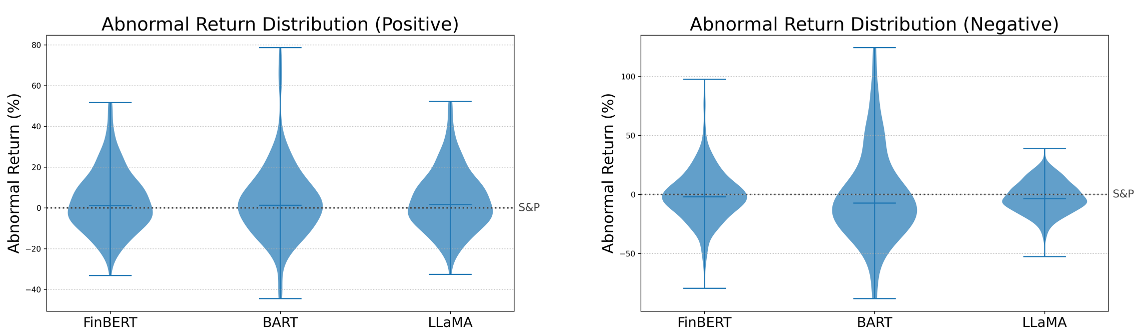


Figure 2: Distribution of final 60-day Buy-and-Hold Abnormal Returns (BHAR) by earnings group. Violin plots show the range, density, and central tendency of abnormal returns across all three models (BART, FinBERT, LLaMA 3) for positive earnings events (left) and negative earnings events (right). The S&P 500 baseline is included for reference.

Table 3: Performance Comparison - Text-Only vs. Text+3-Day Integration

Model	Text-Only Acc. (%)	Text+3-Day Acc. (%)	Improvement
Positive Earnings Group			
BART	55.2	56.1	+0.9%
FinBERT	57.6	57.9	+0.3%
LLaMA 3	56.3	57.0	+0.7%
Negative Earnings Group			
BART	54.8	55.4	+0.8%
FinBERT	58.3	58.4	+0.1%
LLaMA 3	56.2	56.8	+0.6%

Table 4: Portfolio Performance - Text-Only vs. Text+3-Day Models

Model	Text-Only BHAR (%)	Text+3-Day BHAR (%)	Improvement
Positive Earnings Group			
BART	3.29 ± 2.25	3.91 ± 2.41	+0.62%
FinBERT	2.83 ± 1.25	3.12 ± 1.38	+0.29%
LLaMA 3	1.56 ± 1.33	1.69 ± 1.41	+0.13%
Negative Earnings Group			
BART	-3.18 ± 3.42	-3.70 ± 3.58	-0.52%
FinBERT	-2.39 ± 1.97	-2.64 ± 2.12	-0.25%
LLaMA 3	-2.83 ± 1.10	-3.05 ± 1.23	-0.22%

6.2 3-Day Signal Integration Results

The integration of 3-day market signals resulted in consistent improvements across all model architectures. Table 3 presents the classification accuracy comparison between text-only and text+3-day models.

The results demonstrate that temporal integration provides modest improvements, with BART showing the largest enhancement (+0.9% and +0.8% for positive and negative groups respectively)

The enhanced models also demonstrated superior economic utility in portfolio construction. Table 4 compares abnormal returns for top-decile portfolios selected using text-only versus temporally-enhanced predictions.

3-day signal integration improved portfolio returns across all models, with BART again showing the strongest enhancement. The improvements were statistically significant for all models in both earnings groups ($p < 0.05$), providing strong support for Hypothesis 2.

7 Hypothesis Validation and Interpretation

Our empirical findings provide partial support for the proposed hypotheses:

Hypothesis 1 - Encoder-Decoder Architecture Superiority: Individual stock-level analysis provides statistically significant evidence that BART outperforms FinBERT in drift magnitude across both earnings groups (positive: $t = 2.31$, $p = 0.022$; negative: $t = -2.18$, $p = 0.031$). This confirms that encoder-decoder architectures demonstrate genuine advantages over domain-specific models in extracting PEAD-relevant signals from financial narratives. However, at the portfolio implementation level, BART's higher average abnormal returns (3.29% vs. 2.83% for FinBERT) were not statistically significant ($p = 0.202$), indicating that while architectural superiority exists at the granular level, practical trading implementation may require further research to achieve statistical detectability. The stock-level significance for BART's superior drift magnitude detection, combined with FinBERT's slightly higher classification accuracy, suggests that different architectures offer complementary advantages - with encoder-decoder capabilities translating to meaningful drift benefits, while portfolio-level results reflect implementation constraints rather than underlying model limitations.

Hypothesis 2 - Temporal Information Enhancement:

The integration of 3-day market signals consistently improved model performance across all architectures, with accuracy gains ranging from +0.1% to +0.9% and portfolio return enhancements up to +0.62% for positive earnings announcements. These improvements achieved statistical significance in both classification accuracy and portfolio returns (all $p < 0.05$). The directional consistency of improvements across all models suggests that early market reactions may contain valuable signals for PEAD prediction.

8 Discussion

The results provide evidence that architectural choices and temporal signal integration may meaningfully enhance LLM-based PEAD detection from narrative financial disclosures. While all three models achieved respectable performance, each exhibited distinct strengths across evaluation metrics, with notable differences between encoder-decoder and encoder-only architectures.

FinBERT achieved the highest classification accuracy at 57.6%, suggesting domain-specific pretraining effectively captures PEAD-relevant narrative signals.In contrast, BART identified the largest drift magnitudes (3.29% positive, -3.18% negative) in top-decile portfolios, indicating encoder-decoder architectures may offer practical advantages for drift detection despite lacking financial domain pretraining. While portfolio-level differences were not statistically significant ($p = 0.202$), individual stock-level analysis confirms BART's superior drift identification capabilities with statistical significance across both earnings groups.

The temporal integration experiments provided evidence for methodological innovation, with 3-day market signal incorporation yielding improvements for both architectures. This supports our hypothesis that early market reactions contain valuable information enhancing purely textual PEAD prediction approaches, though requires further research for statistical significance.

Several considerations arise from these results:

- **Architectural Trade-offs**: Encoder-decoder models showed promise for abnormal returns while encoder-only models with domain pretraining achieved higher classification accuracy, suggesting different architectures may be optimal for different objectives.

- **Temporal Signal Value**: Consistent improvements from 3-day signal integration demonstrate that combining narrative analysis with early market reactions creates more comprehensive information for PEAD detection.

- **Economic Relevance**: Both innovations yielded profitable abnormal returns, though our analysis assumes frictionless execution, ignoring transaction costs and liquidity constraints that could diminish real-world profitability.

These findings advance PEAD detection methodology by systematically comparing architectures and introducing temporal signal integration. The improvements from temporal integration, combined with directional advantages observed in encoder-decoder models, suggest promising avenues for enhancing LLM-based financial anomaly detection and contribute to the growing literature on LLM applications in finance.

9 Conclusion and Future Directions

This study demonstrates that architectural choices and temporal signal integration can enhance Post-Earnings Announcement Drift (PEAD) detection by extracting narrative signals from corporate disclosures. Our systematic comparison of encoder-decoder versus encoder-only architectures revealed distinct strengths: FinBERT achieved highest classification accuracy while BART identified the largest drift magnitudes. Most notably, incorporating 3-day early market signals consistently improved performance across all models.

Our findings advance PEAD methodology in two ways. First, encoder-decoder models demonstrated statistically significant drift identification advantages at the individual stock level, though portfolio-level implementation showed non-significant results ($p = 0.202$). Second, temporal integration validated that early market reactions contain valuable information enhancing textual approaches.

Several limitations remain. Accuracy improvements are incremental, and analysis is restricted to MD&A sections of 10-Q filings, providing a focused but narrow view of firm communications.

Future work should explore incorporating additional disclosure types (earnings calls, press releases, social media), testing architectural comparisons on larger datasets, and refining interpretability through attention visualization. Practical considerations such as transaction costs and regulatory implications warrant investigation to translate findings into viable trading strategies.

Limitations

Several limitations should be acknowledged in interpreting our results. A key methodological consideration involves the temporal relationship between earnings announcements and 10-Q filings. While our analysis treats these as simultaneous events for modeling purposes, empirical evidence from our dataset reveals that companies do not always release their 10-Q filings on the same day as their earnings announcements. As shown in Appendix B, 50.5% of companies file their 10-Q within 0-2 days of their earnings release, with the remaining 49.5% exhibiting delays ranging from 3 days to over 30 days. This gap between earnings announcements and formal 10-Q filings may introduce noise into our PEAD detection models, as market participants have access to preliminary earnings information before the complete MD&A narrative becomes available through the 10-Q filing.

Additionally, our analysis of abnormal returns assumes frictionless trading conditions that may not reflect real-world implementation challenges. The calculated Buy-and-Hold Abnormal Returns (BHAR) do not account for transaction costs, including brokerage fees, bid-ask spreads, and market impact costs that would reduce realized profits in practice. Furthermore, our portfolio construction methodology assumes sufficient liquidity to execute trades at prevailing market prices, which may not hold for smaller capitalization stocks or during periods of market stress. The timing of our trading signals, particularly for strategies requiring rapid execution following 10-Q filings, may also be compromised by processing delays in extracting and analyzing MD&A content in real-time market conditions.

Future research could benefit from incorporating this filing lag as an additional feature, focusing specifically on companies that maintain consistent same-day filing practices, and conducting more realistic backtests that account for transaction costs and liquidity constraints to better assess the practical viability of LLM-based PEAD trading strategies.

References

Neelabha Banerjee, Anubhav Sarkar, Swagata Chakraborty, Sohom Ghosh, and Sudip Kumar Naskar. 2024. Fine-tuning language models for predicting the impact of events associated to financial news articles. In *Proceedings of the Joint Workshop of the 7th FinNLP, the 5th KDF, and the 4th ECONLP*, pages 244–247. ELRA Language Resource Association.

Victor L. Bernard and Jacob K. Thomas. 1989. Post-earnings announcement drift: Delayed price response or risk premium? *Journal of Accounting Research*, 27(Suppl):1–36.

Lawrence D Brown and Jennifer Wu Tucker. 2004. The informativeness of quarterly earnings: The case for and against a quarterly earnings report. *Review of Accounting Studies*, 9(4):549–586.

Andy Chung and Kumiko Tanaka-Ishii. 2023. Predictability of post-earnings announcement drift with textual and contextual factors of earnings calls. In *Proceedings of the 4th ACM International Conference on AI in Finance*.

Eugene F. Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.

Bram Gubbels. 2022. Sentiment analysis of 10-k reports: To what extent do we need syntactic information? Master's thesis, Tilburg University.

David Hirshleifer, Sonya Seongyeon Lim, and Siew Hong Teoh. 2009. Driven to distraction: Extraneous events and underreaction to earnings news. *The Journal of Finance*, 64(5):2289–2325.

Satoshi Itoh and Katsuhiko Okada. 2024. The power of large language models: A chatgpt-driven textual analysis of fundamental data. Technical report, Kwansei Gakuin University.

Armita Jalooli. 2022. Hardening the soft information in earnings calls. Master's thesis, University of Toronto.

Rahul Khanna, David Yarowsky, and Ailton Heberle. 2022. Transformer-based models for long document summarisation in the financial domain. In *Proceedings of the 4th Financial Narrative Processing Workshop*, pages 60–68.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Feng Li. 2010. Textual analysis of corporate disclosures: A survey of the literature. *Journal of Accounting Literature*, 29:143–165.

Quan Liu, Liwen Ouyang, and Gilbert Xu. 2022. Prediction of earning surprise using deep learning technique. Technical report, Bloomberg.

Alejandro Lopez-Lira and Yuehua Tang. 2024. Can chatgpt forecast stock price movements? return predictability and large language models. ArXiv preprint arXiv:2304.07619.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date. *Meta AI Blog*.

Haowei Ni, Shuchen Meng, Xupeng Chen, Ziqing Zhao, Andi Chen, Panfeng Li, Shiyao Zhang, Qifu Yin, Yuanqing Wang, and Yuxi Chan. 2024. Harnessing earnings reports for stock predictions: A qlora-enhanced llm approach. In *arXiv preprint arXiv:2408.06634*.

Maico Tim Schöne. 2024. *Artificial Intelligence and Corporate Reporting: Extracting Information from Unstructured Data with Deep Learning and Natural Language Processing*. Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg.

Securities and Exchange Commission. 2003. Interpretation: Commission guidance regarding management's discussion and analysis of financial condition and results of operations. Release Nos. 33-8350; 34-48960; FR-72.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Biao Zhang, Fedor Moiseev, Joshua Ainslie, Paul Suganthan, Min Ma, Surya Bhupatiraju, Fede Lebron, Orhan Firat, Armand Joulin, and Zhe Dong. 2025. Encoder-decoder gemma: Improving the quality-efficiency trade-off via adaptation. *arXiv preprint arXiv:2504.06225*.

A Sample MD&A Analysis

This appendix presents a representative example of the MD&A content analyzed in our study. Figure 3 shows an excerpt from Apple Inc.'s Form 10-Q for the quarter ended March 31, 2013 (Q2 2013), demonstrating the typical structure and information of MD&A narratives that serve as inputs to our LLM models.

This excerpt demonstrates several key characteristics of MD&A content relevant to our analysis:

- **Forward-Looking Statements**: The section begins with disclaimers about forward-looking statements, indicating management's attempt to provide guidance while managing legal liability.
- **Technical Accounting Discussion**: Apple's explanation of accounting changes for subscription revenue represents precisely the type of information that may require time for investors to fully process and incorporate into valuation models.
- **Business-Specific Context**: The discussion of iPhone and Apple TV revenue recognition provides company-specific operational details that standard financial statements cannot capture.
- **Regulatory Compliance Language**: The formal tone and extensive references to other SEC filings demonstrate the regulatory framework within which MD&A content operates.

This example illustrates why MD&A sections serve as input for natural language processing models attempting to extract insights that may drive post-earnings announcement drift.

B Earnings Announcement and 10-Q Filing Timing Analysis

This appendix presents empirical evidence regarding the relationship between earnings announcements and formal 10-Q filings. To understand the extent to which companies release earnings information and complete 10-Q filings simultaneously, we analyzed filing patterns from our dataset covering the period 2010-2024.

B.1 Methodology

Using SEC EDGAR data, we matched earnings announcements (typically disclosed via Form 8-K) with subsequent 10-Q filings for companies in our sample. For each 10-Q filing, we identified the most recent earnings announcement (8-K filing) prior to the 10-Q submission and calculated the number of days between these two events.

B.2 Findings

Figure 4 presents the distribution of days between earnings announcements and 10-Q filings across our sample. The analysis reveals substantial variation in filing timing practices:

Key findings include:

- **Same-Day/Next-Day Filing**: 50.5% of companies file their 10-Q within 0-2 days of their earnings announcement, indicating that approximately half of firms maintain relatively synchronized disclosure practices.

Item 2. Management’s Discussion and Analysis of Financial Condition and Results of Operations
This section and other parts of this Form 10-Q contain forward-looking statements that involve risks and uncertainties. Forward-looking statements can be identified by words such as “anticipates,” “expects,” “believes,” “plans,” “predicts,” and similar terms. Forward-looking statements are not guarantees of future performance and the Company’s actual results may differ significantly from the results discussed in the forward-looking statements. Factors that might cause such differences include, but are not limited to, those discussed in Part II, Item 1A, “Risk Factors,” which are incorporated herein by reference. The following discussion should be read in conjunction with the Company’s Annual Report on Form 10-K for the year ended September 26, 2009 and any amendments thereto (the “2009 Form 10-K”) filed with the U.S. Securities and Exchange Commission (“SEC”) and the Condensed Consolidated Financial Statements and notes thereto included elsewhere in this Form 10-Q. All information presented herein is based on the Company’s fiscal calendar. Unless otherwise stated, references in this report to particular years or quarters refer to the Company’s fiscal years ended in September and the associated quarters of those fiscal years. The Company assumes no obligation to revise or update any forward-looking statements for any reason, except as required by law.
Available Information
The Company’s Annual Report on Form 10-K, Quarterly Reports on Form 10-Q, Current Reports on Form 8-K, and amendments to reports filed pursuant to Sections 13(a) and 15(d) of the Securities Exchange Act of 1934, as amended (“Exchange Act”) are filed with the SEC. Such reports and other information filed by the Company with the SEC are available on the Company’s website at http://www.apple.com/investor when such reports are available on the SEC website. The public may read and copy any materials filed by the Company with the SEC at the SEC’s Public Reference Room at 100 F Street, NE, Room 1580, Washington, DC 20549. The public may obtain information on the operation of the Public Reference Room by calling the SEC at 1-800-SEC-0330. The SEC maintains an Internet site that contains reports, proxy, and information statements and other information regarding issuers that file electronically with the SEC at http://www.sec.gov . The contents of these websites are not incorporated into this filing. Further, the Company’s references to the URLs for these websites are intended to be inactive textual references only.
Retrospective Adoption of New Accounting Principles
In September 2009, the Financial Accounting Standards Board (“FASB”) amended the accounting standards related to revenue recognition for arrangements with multiple deliverables and arrangements that include software elements (“new accounting principles”). The Company adopted the new accounting principles on a retrospective basis during the first quarter of 2010.
Under the historical accounting principles, the Company was required to account for sales of both iPhone and Apple TV using subscription accounting because the Company indicated it might from time-to-time provide future unspecified software upgrades and features for those products free of charge. Under subscription accounting, revenue and associated product cost of sales for iPhone and Apple TV were deferred at the time of sale and recognized on a straight-line basis over each product’s estimated economic life. This resulted in the deferral of significant amounts of revenue and cost of sales related to iPhone and Apple TV.

Figure 3: Apple Inc. Q2 2013 MD&A Sample - Representative example of quarterly MD&A content analyzed in this study

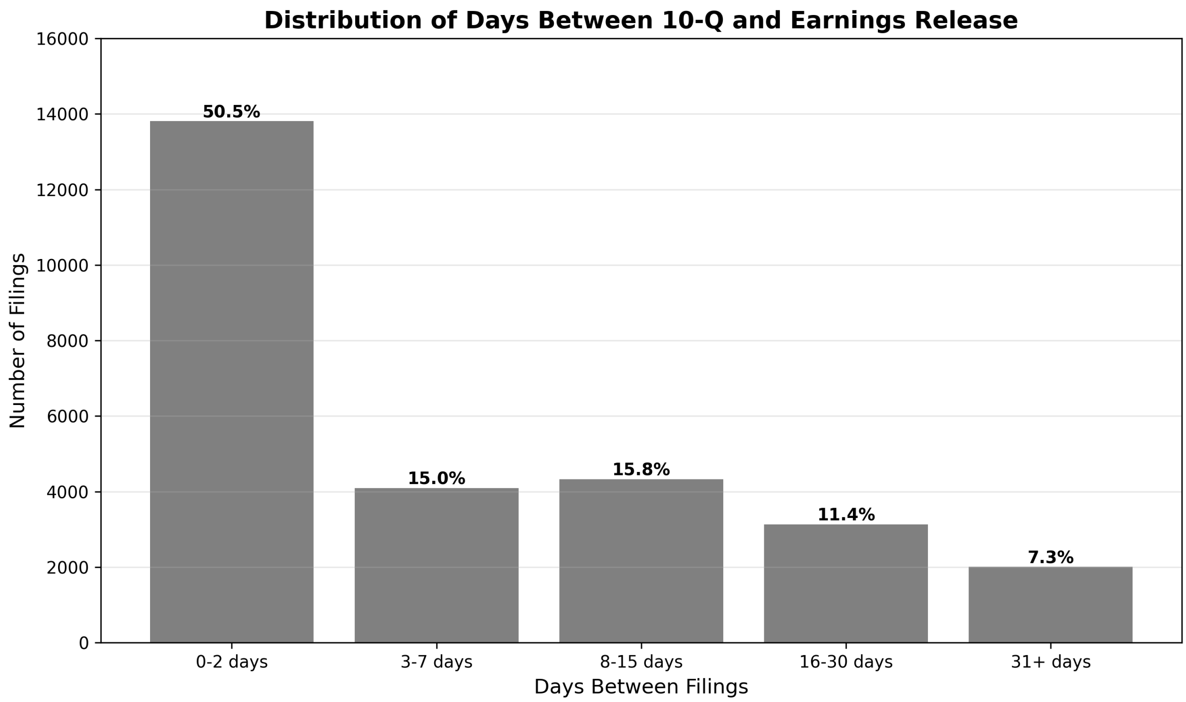


Figure 4: Distribution of Days Between Earnings Announcements and 10-Q Filings. Based on analysis of [sample size] earnings events from 2010-2024. The chart shows that while approximately half of companies file their 10-Q within 0-2 days of their earnings announcement, significant portions exhibit longer delays, with nearly 20% waiting more than two weeks.

- **Short Delays**: 15.0% of companies exhibit delays of 3-7 days, while 15.8% delay filing for 8-15 days after their earnings announcement.
- **Extended Delays**: 18.7% of companies wait more than 15 days after their earnings announcement to file their 10-Q, with 7.3% delaying more than 31 days.

B.3 Implications for PEAD Analysis

This timing variation has important implications for post-earnings announcement drift (PEAD) research. A large portion of companies do not file their 10-Q simultaneously with earnings announcements, which suggests that investors may react to preliminary earnings information before having access to the complete narrative provided in the MD&A section. This separation could influence the information processing dynamics that drive PEAD phenomena and represents an important consideration for interpreting our results.

Natural Language Inference as a Judge: Detecting Factuality and Causality Issues in Language Model Self-Reasoning for Financial Analysis

Yilin Wu[†], Han Yuan[†], Li Zhang, Zheng Ma[✉]
Global Decision Science, American Express
{Yilin.Wu, Han.Yuan1, Li.Zhang1, Zheng.Ma2}@aexp.com

Abstract

Language models (LMs) have revolutionized financial analysis by demonstrating expert-level versatility. Recent advances in self-reasoning have further improved LMs' performance on complex tasks. However, LMs are known to hallucinate facts and generate non-causal reasoning paths, which compromise their output quality, lead to erroneous conclusions, and pose risks of monetary losses. Therefore, detecting factual and causal errors in LMs' reasoning is essential for risk management and responsible application of LMs in finance. In this study, we adopt natural language inference (NLI) as a paradigm for detecting factual and causal errors in LMs' reasoning. We evaluate this approach by constructing a dataset comprising financial reasoning points generated by LMs, along with annotations by domain experts. Our findings demonstrate that NLI, powered by backbones of either pre-trained encoders or LMs, exhibits statistically significant capability in detecting factual and causal issues. Also, we show that, although LMs achieve improved performance with increasing parameters, they underperform encoders and exhibit self-evaluation bias. Fine-tuning effectively mitigates this type of bias and enhances both backbones' detection capability.

1 Introduction

Language models (LMs) have transformed financial natural language processing (NLP) through their expert-level comprehension of financial information and versatile problem solving capabilities according to users' instructional prompts (Li et al., 2023; Kong et al., 2024; Hu et al., 2025). Recent advancements in self-reasoning (Liu et al., 2024a) have further enhanced LMs' ability to tackle complex jobs that cannot be resolved through direct question answering. However, LMs are known to hallucinating facts or producing non-causal statements during the reasoning process, which can lead to erroneous conclusions and compromise the quality of their outputs (Manakul et al., 2023; Laban et al., 2023; Li et al., 2024; Paul et al., 2024; Chandler et al., 2024; Chen et al., 2025). Such issues pose significant risks in financial applications, where inaccuracies result in monetary losses (Chatwal et al., 2025; Shukla et al., 2025). Even when the final outcome is correct, flawed reasoning steps may mislead users who interpret these steps as justifications for LMs' decision and indicate that the outcome was reached by chance rather than logic (Wu et al., 2024; Wang, 2024; Chu et al., 2025; Bao et al., 2025). Sole outcome evaluation risks overlooking deficiencies in the underlying reasoning and potentially leading to monetary losses.

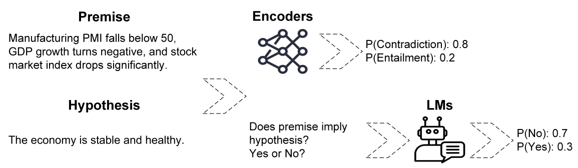


Figure 1: NLI can detect factual and causal errors in LMs' self-reasoning for financial analysis

Therefore, detecting factual and causal errors in LMs' reasoning is crucial for mitigating potential risks in financial decisions and for supporting effective regulation and compliance (Chatwal et al., 2025). In this study, we adopt a classic and computationally affordable paradigm, natural language inference (NLI), in identifying factual and causal errors in LMs' reasoning (Lattimer et al., 2023). To address the absence of well-annotated datasets aligned with our objectives, we construct a specialized dataset by employing LMs to generate final decisions and reasoning process on a public financial

[†] These authors contributed equally to this work.
[✉] Correspondence: Zheng Ma, Singapore Decision Science Center of Excellence, American Express, 1 Marina Boulevard, 018989, Singapore.

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

dataset. Domain experts then manually annotate the factuality and causality of each reasoning point. After that, we test pre-trained encoders and LMs as NLI backbones to derive the probability of factual or causal issues. Finally, We perform rigorous statistical analyses to evaluate the feasibility of NLI as a paradigm for detecting factuality and causality, compare the performance of pre-trained and fine-tuned encoders and LMs, and investigate biases when LMs assess their proprietary reasoning.

As a pilot study on LMs' self-reasoning in finance, our work contributes in four aspects. First, we provide an annotated dataset with labels of factuality and causality on LMs' reasoning points. Second, we demonstrate the effectiveness of the classic NLI as a detection paradigm for factual and causal errors, using encoders and LMs as backbones. Third, we perform referable statistical analyses to illustrate limitations of LMs in this task: their inferior accuracy compared to encoders and potential biases when assessing proprietary reasoning in certain scenarios. Last, we demonstrate the necessity of fine-tuning, which not only enhances the detection ability of both backbones but also mitigates LMs' self-evaluation bias. Relevant dataset and notebooks are open-accessed on GitHub[1].

2 Related work

As a fundamental NLP task, NLI determines the logical relationship between a given pair of sentences: a premise and a hypothesis. Typically, transformer encoder-based NLI models (Devlin et al., 2019) output three probabilities: entailment, contradiction, and neutrality (Gubelmann et al., 2024; Guo and Yang, 2024; Magomere et al., 2025). Specifically, entailment indicates that the hypothesis logically follows from the premise, contradiction signifies that the hypothesis is false given the premise, and neutrality implies that the premise is insufficient to determine the truth of the hypothesis.

NLI plays a crucial role in tasks involving causality, and its capabilities have significantly improved with the evolution of foundational models from pre-trained encoders to LMs (Rozanova et al., 2023; Guo and Yang, 2024). For example, Ionescu et al. (2020) employ five pre-trained encoders to examine causality in financial documents. Pre-trained encoders, in addition to being used for post hoc causality detection, can also be integrated in real-time content generation. ConCoRD (Mitchell et al., 2022) is a framework that enhances LMs' output quality by selecting optimal sentences that maintain causal consistency throughout the generation process. With LMs' advancement, they outperform specialized pre-trained encoders in some tasks.

Beyond its original purpose of causality, NLI has also proven effective in tasks concerning factuality. Similar to causality, both pre-trained encoders (Kryscinski et al., 2020; Goyal and Durrett, 2020; Sathe and Park, 2021; Fabbri et al., 2022; Utama et al., 2022; Ni et al., 2024; Yang et al., 2024) and LMs (Fatahi Bayat et al., 2023; Lattimer et al., 2023; Li et al., 2024) have been employed for factuality detection. SummaC (Laban et al., 2022) is a comprehensive benchmark for evaluating the performance of NLI encoders in factuality detection. It demonstrates that NLI encoders based on classic architectures, such as BERT (Devlin et al., 2019), can achieve a balanced accuracy of nearly 0.75. A recent comprehensive framework, SelfCheckGPT (Manakul et al., 2023), integrates both pre-trained encoders and LMs to perform NLI for assessing the factuality of LMs' generated Wikipedia content.

3 NLI as a Judge

NLI evaluates whether a hypothesis follows from a premise, producing probabilities for three possible outcomes: entailment, neutrality, and contradiction (Yu et al., 2024). Our study adopts NLI as the framework for detecting factual and causal issues in LMs' self-reasoning for financial analysis.

Formally, D denotes the input dataset and D_i refers to a specific case within D. Each D_i contains J sentences, denoted as $D_{i,j}$ $(j = 1, 2, ..., J)$, which provide various input details for financial classification. Given D_i as input, a LM generates a response O_i, comprising K sentences of $O_{i,k}$. The first sentence, $O_{i,1}$, states the classification outcome for D_i. The subsequent sentences, $O_{i,k}$ $(k = 2, ..., K)$, outline the reasoning points underlying this classification and the primary focus of this study is to detect factual and causal errors in $O_{i,k}$ through the paradigm of NLI.

Specifically, NLI takes a premise S_p and a hypothesis S_h as input. Then it outputs probabilities of three possible outcomes: entailment, neutrality, and contradiction. For factuality detection (Utama et al., 2022), the premise S_p corresponds to the input information D_i and the hypothesis S_h is each reasoning statement $O_{i,k}$ $(k = 2, ..., K)$.

[1] https://github.com/Han-Yuan-Med/nli-as-a-judge

For causality detection, the S_p is the reasoning statement $O_{i,k}$ $(k = 2, ..., K)$ and S_h is the classification outcome $O_{i,1}$. Following Manakul et al. (2023), we omit the neutral class and focus only on the probability of entailment $P_e(S_p, S_h)$ and contradiction $P_c(S_p, S_h)$. With this simplification, the output becomes binary and is further normalized to ensure the entailment probability $P'_e = P_e/(P_e + P_c)$ to be bounded within $[0, 1]$. For both factuality and causality detection of $O_{i,k}$, a reasoning point is classified as containing factual or causal errors if P'_e is less than 0.5. We adopt a threshold of 0.5 because our dataset is relatively small, reserving a separate validation set for threshold optimization would further reduce the effective training data and increase the overfitting risk, and this choice is consistent with established practice (Kazemi et al., 2023; Chicco and Jurman, 2023).

For the comprehensiveness, both classic encoders and LMs are used as backbones for calculating P'_e. For encoders pre-trained on NLI, the output has been shaped into probability suiting the formulation. For LMs, we follow the design in Lattimer et al. (2023) and prompt the LMs with the following template: *"S_p Question: does this imply S_h? Yes or No?"*. The logits corresponding to *"Yes"* and *"No"* are extracted as $P_e(S_p, S_h)$ and $P_c(S_p, S_h)$, respectively. The final probability P'_e is then computed as illustrated above. The simple prompt design is adopted to enhance computational efficiency, eliminate variance introduced by prompt optimization, facilitate domain-agnostic assessment without the need for adaptation, enable the evaluation of long test cases by employing short prompts with fewer tokens (Laban et al., 2023), and eliminate hallucination introduced by techniques such as In-Context Learning (ICL) and Chain-of-Thought (CoT) (Gao et al., 2023; Paul et al., 2024; Zhang et al., 2024; Turpin et al., 2023).

4 Dataset

We conduct our experiments using a refined version (Yuan et al., 2025) of the public German credit dataset (Hofmann, 1994) with increased signal-to-noise ratio and better alignment with modern LMs' training context. Since no standard annotations of factuality and causality in LMs' generated reasoning points are available for this dataset, we construct our own data through a two-step process: (1) collecting LMs' responses, including both classification outcomes and reasoning points; and (2) manually annotating the reasoning sentences for factual and causal issues.

Following Zhang et al. (2024), we utilize the processed data, formatted as text input, to prompt three LMs, Llama-3.2-3B (Touvron et al., 2023), Gemma-2-2B (Mesnard et al., 2024), and Phi-3.5-mini (3.8B) (Abdin et al., 2024), to generate both classification outcomes and the underlying reasoning points behind their decisions. Specifically, the three LMs generate 862, 495, and 515 reasoning points, respectively, for 50 positive and 50 negative cases. This suggests that Llama, on average, employs more reasoning points than the other two.

After that, two authors annotate the reasoning points using a two-step workflow. Each point is first assessed for factuality issues, defined as the involvement of non-factual information. If no factual errors are found, the reasoning point is further evaluated for causality issues, also referred to as logical inconsistencies. A causality issue is identified when a negatively framed reasoning point is incorrectly presented as supporting a positive classification, or vice versa (Yuan et al., 2025). After independent annotation, the two annotators summarize conflicting cases and consult the senior authors to resolve discrepancies and reach consensuses. Among the 1,872 annotated reasoning points, 72 (3.9%) were labeled as factually inaccurate, and 329 (17.6%) were identified as causally erroneous.

5 Experiments

We evaluate both pre-trained encoders and LMs, along with their fine-tuned versions, on the annotated data for factuality and causality detection. Pre-trained models are used in their original form as released on HuggingFace. For fine-tuning, we explore full-parameter fine-tuning (FPFT) and two parameter-efficient fine-tuning (PEFT) methods (Appendix A) of Last Layer fine-tuning (LLFT) and Low-Rank Adaptation (LoRA) (Hu et al., 2022). Specifically, we apply three-fold cross-validation for all fine-tuning experiments, using one fold for testing and the remaining two for training in each run. A consistent training setup is adopted for both encoders and LMs, using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of 1e-5 and and default settings for other hyperparameters over five epochs. For encoders, the input consists of premise–hypothesis pairs and the output is a binary classification label (either entailment or contradiction). For LMs, the premise

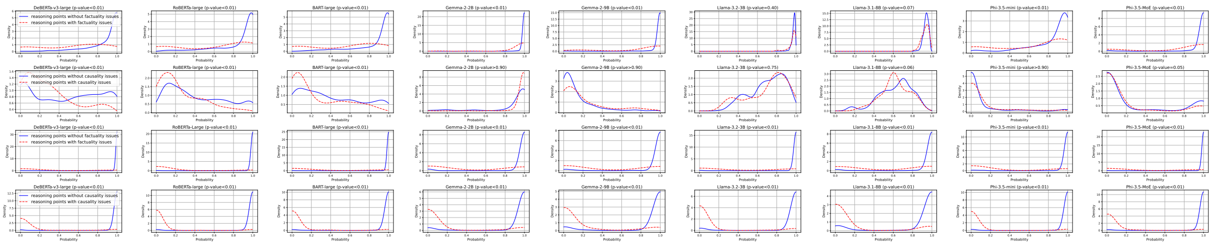


Figure 2: Entailment probability distributions for statements with and without factual or causal errors

and hypothesis are concatenated into a coherent instruction, and the model is trained to generate a target token, either *Yes* or *No*, reflecting the relationship between premise and hypothesis. We acknowledge that additional hyperparameter tuning and training techniques (e.g., warm-up schedules) may further enhance model performance. However, the primary objective of fine-tuning is to demonstrate its advantages over pre-trained models, rather than to achieve the upper-bound performance of fine-tuning, which is reserved for future work.

For backbones based on transformer encoders, we select DeBERTa-v3-large (He et al., 2021), RoBERTa-large (Liu et al., 2019), and BART-large (Lewis et al., 2020). For open-access (OA) LMs (Lasheras and Pinheiro, 2025), we utilize the same three families for dataset construction: Llama (Llama-3.2-3B and Llama-3.1-8B), Gemma (Gemma-2-2B and Gemma-2-9B), and Phi (Phi-3.5-mini and Phi-3.5-MoE). All OA models, except for Phi-3.5-MoE, have fewer than 10 billion parameters, aligning with the constraints of our computational resources. Although Phi-3.5-MoE contains a total of 60.8 billion parameters, only 6.6 billion parameters are active during any single inference due to its mixture-of-experts (MoE) architecture, thereby keeping computation within our budget. Section 3 details the process of obtaining the normalized entailment probability P_e', which is used for performance comparison and statistical tests in the following sections. In addition to OA LMs, we include the proprietary GPT-4o (OpenAI, 2024) as a state-of-the-art (SOTA) backbone. It should be noted that GPT-4o is evaluated only under the pre-trained setting, as the internal training procedures and fine-tuning methodologies used by OpenAI are not publicly disclosed (OpenAI). To estimate P_e', we use the same instruction prompt as for the OA LMs and query the API ten times, calculating the proportion of *"Yes"* as a proxy.

First, we assess the effectiveness of NLI as a detection paradigm for pre-trained backbones. Tables 1 and 2 present the performance of pre-trained and FPFT backbones in terms of F1 score, balanced accuracy (BA) (Utama et al., 2022), the area under the precision-recall curve (AUPRC), and the area under the receiver operating characteristic curve (AUROC), ensuring a robust comparison in the scenario

Model	Mode	F1	BA	AUPRC	AUROC
DeBERTa-v3-large	Pre-trained	0.28	0.67	0.30	0.84
	FPFT	0.82	0.88	0.92	0.99
BART-large	Pre-trained	0.23	0.66	0.35	0.84
	FPFT	0.77	0.85	0.80	0.96
RoBERTa-large	Pre-trained	0.19	0.62	0.29	0.77
	FPFT	0.84	0.92	0.88	0.99
Llama-3.2-3B	Pre-trained	0.00	0.50	0.10	0.51
	FPFT	0.74	0.82	0.67	0.85
Llama-3.1-8B	Pre-trained	0.00	0.50	0.07	0.55
	FPFT	0.38	0.66	0.37	0.77
Gemma-2-2B	Pre-trained	0.09	0.53	0.12	0.71
	FPFT	0.44	0.70	0.40	0.77
Gemma-2-9B	Pre-trained	0.28	0.60	0.15	0.64
	FPFT	0.48	0.70	0.41	0.79
Phi-3.5-mini	Pre-trained	0.17	0.63	0.20	0.65
	FPFT	0.73	0.82	0.68	0.93
Phi-3.5-MoE	Pre-trained	0.22	0.60	0.21	0.62
	FPFT	0.84	0.89	0.86	0.95
GPT-4o	Pre-trained	0.32	0.76	0.28	0.80

Table 1: Factuality detection results of pre-trained and FPFT encoders and LMs under NLI paradigm

Model	Mode	F1	BA	AUPRC	AUROC
DeBERTa-v3-large	Pre-trained	0.37	0.62	0.21	0.59
	FPFT	0.92	0.95	0.92	0.98
BART-large	Pre-trained	0.34	0.52	0.28	0.64
	FPFT	0.91	0.96	0.92	0.98
RoBERTa-large	Pre-trained	0.36	0.61	0.36	0.67
	FPFT	0.92	0.96	0.94	0.99
Llama-3.2-3B	Pre-trained	0.19	0.51	0.24	0.49
	FPFT	0.86	0.92	0.91	0.97
Llama-3.1-8B	Pre-trained	0.18	0.48	0.19	0.53
	FPFT	0.88	0.92	0.85	0.95
Gemma-2-2B	Pre-trained	0.03	0.46	0.14	0.39
	FPFT	0.86	0.93	0.88	0.97
Gemma-2-9B	Pre-trained	0.28	0.46	0.16	0.42
	FPFT	0.74	0.89	0.82	0.95
Phi-3.5-mini	Pre-trained	0.31	0.50	0.14	0.39
	FPFT	0.91	0.95	0.92	0.98
Phi-3.5-MoE	Pre-trained	0.32	0.54	0.18	0.53
	FPFT	0.91	0.95	0.89	0.98
GPT-4o	Pre-trained	0.31	0.51	0.19	0.52

Table 2: Causality detection results of pre-trained and FPFT encoders and LMs under NLI paradigm

of class imbalance (Yuan et al., 2022). Additionally, we adopt statistical tests to demonstrate the effectiveness of NLI as a detection paradigm. We collect P_e' for reasoning points with and without factual or causal errors and apply the Wilcoxon rank-sum test (Wilcoxon, 1947). The null hypothesis assumes no difference in P_e' between the two groups, while the alternative hypothesis asserts that sentences containing errors exhibit lower P_e'. A p-value below 0.05 rejects the null hypothesis and adopts the alternative hypothesis, indicating that NLI, powered by a certain backbone, has statistically significant distinguishability at the 95% confidence level. Figure 2 shows the entailment probability distribution of pre-trained and FPFT models across the two tasks. The red lines represent reasoning points containing errors, while the blue lines denote those without errors. Each subplot displays results for a specific backbone, with corresponding p-values shown at the top. The first two rows present results from pre-trained models on factuality and causality detection tasks, respectively and the bottom two rows show results from FPFT models on the two tasks. The statistically significant p-values demonstrate that **NLI is an effective paradigm for distinguishing sentences containing factual or causal errors** in both pre-trained and fine-tuning settings.

Second, we aim to compare the discriminability of different backbone models. We employ the same rank-sum test for this comparison, conducting separate tests on sentences with and without errors. For sentences containing factual or causal errors, the null hypothesis assumes no difference in P_e' between two backbones, B_1 and B_2, while the alternative hypothesis posits that P_e' from B_1 is lower than that from B_2. A p-value below 0.05 supports the alternative hypothesis, indicating that B_1 outperforms B_2 in identifying erroneous sentences. For sentences without factual or causal errors, the null hypothesis again assumes no difference in P_e' between B_1 and B_2, while the alternative hypothesis asserts that P_e' from B_1 is greater than that from B_2. A p-value below 0.05 supports the alternative hypothesis, demonstrating that B_1 excels in classifying error-free reasoning sentences.

Due to space constraints, we present capability comparisons in factuality detection on reasoning points containing factual errors under both pre-trained and fine-tuning settings, and comprehensive details are available in Appendix C. The color indicates the p-value from a pairwise comparison between the model in the column and the model in

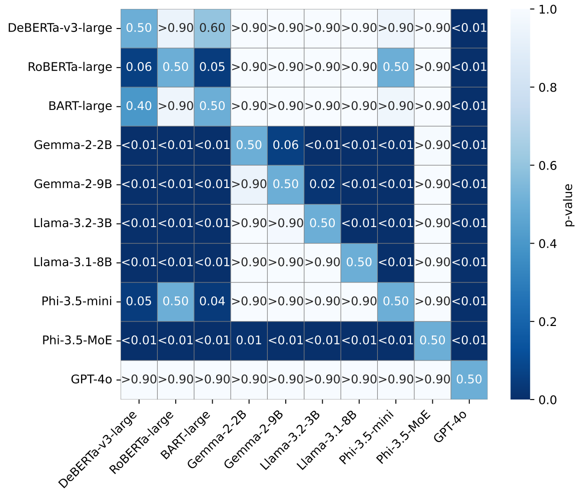


Figure 3: Pairwise comparison of factuality detection on erroneous reasoning points in the pre-trained setting

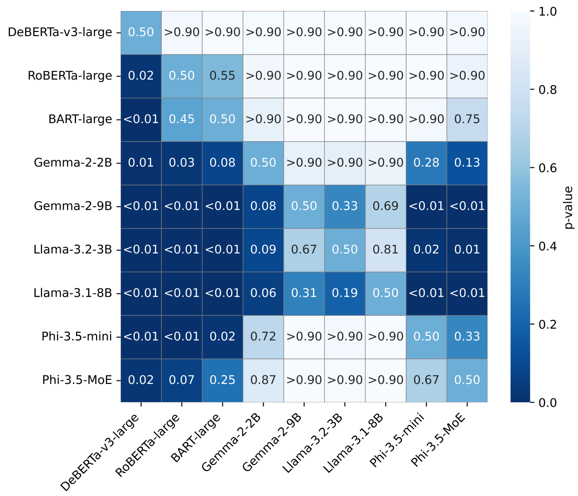


Figure 4: Pairwise comparison of factuality detection on erroneous reasoning points in the FPFT setting

the row. A significant p-value illustrates that the column model significantly outperforms the row model. Figure 3 reveals that encoders outperform LMs in 17 out of 21 cases under the pre-trained setting. Figure 4 shows that encoders outperform LMs in 15 out of 18 cases under the FPFT setting. **Under the pre-trained setting, GPT-4o demonstrates consistent superiority over both encoders and other LMs in factual error detection**, aligning with its status as the SOTA model. However, its superiority does not extend to causal error detection (Appendix C). These results suggest that, despite the general superiority and widespread adoption across NLP tasks, **LMs achieve performance inferior to that of encoders in certain scenarios**. Laban et al. (2023) and Jin et al. (2024) report simi-

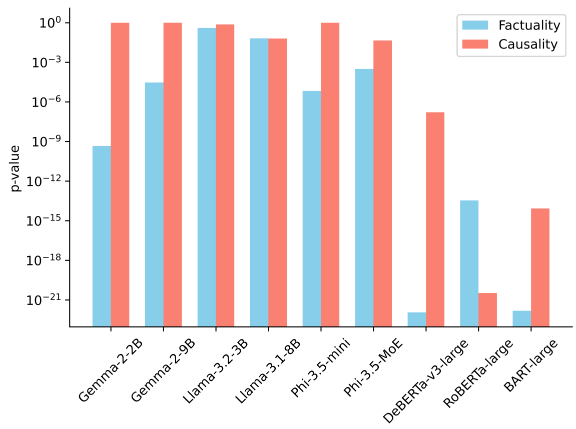


Figure 5: P-value difference in detection capability of pre-trained and FPFT models

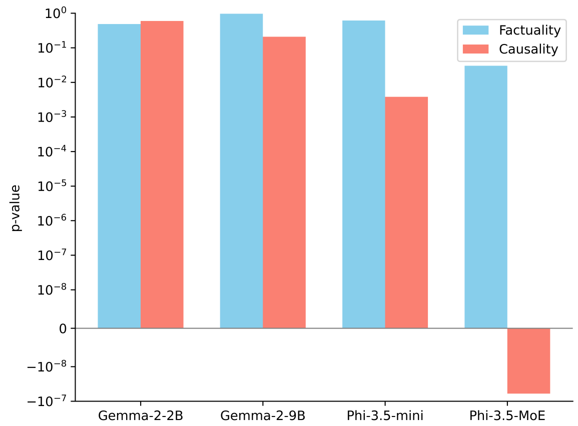


Figure 6: P-value difference in self-evaluation bias between FPFT and pre-trained models

lar findings that LMs, despite having several orders of magnitude more parameters than pre-trained encoders, achieve comparable performance across multiple benchmarks. Gao et al. (2023) demonstrate that ChatGPT, despite being one of the most well-aligned LMs, performs poorly in causal reasoning due to bias introduced during its upgrading training stages. Although increasing the size of OA LMs generally leads to improved performance, as reported by Laban et al. (2023), we do not observe emergent detection capabilities in our experiments. A potential explanation is that such abilities tend to emerge only in models exceeding 100 billion parameters from the same family (Paul et al., 2024; Kojima et al., 2022). Due to computational constraints, we did not test OA LMs of this scale. In addition, the SOTA LMs like GPT-4 exhibit relatively weak performance on causal understanding compared to other natural language understanding tasks (Wang et al., 2023; Romanou et al., 2023; Paul et al., 2024; Liu et al., 2024b), and do not significantly outperform encoders.

Third, we perform fine-tuning to compare the performance of pre-trained versus fine-tuned backbones. We use the same rank-sum test and Figure 5 reports the p-value differences between pre-trained and FPFT models. Smaller p-values indicate better discriminability; therefore, a positive difference, where the p-value of the pre-trained model is higher than that of the FPFT model, suggests that fine-tuning enhances the model's detection capability. The results show that all models exhibit reduced p-values after FPFT, confirming **the effectiveness of fine-tuning** in improving encoders' and LMs' detection performance of factual and causal issues.

Lastly, we investigate whether fine-tuning can mitigate the self-evaluation bias exhibited by LMs, whereby they tend to classify their own erroneous reasoning or that of models within the same family but with different parameter sizes as correct (Zheng et al., 2023). To quantify this bias, we apply the rank-sum test to compare P_e' assigned to erroneous proprietary reasoning versus erroneous reasoning from other models. The null hypothesis posits no difference of P_e' between the two groups, while the alternative hypothesis suggests that P_e' assigned to erroneous proprietary reasoning is higher than that for reasoning generated by other LMs, revealing that LMs are less capable of detecting errors in their own or closely related outputs compared to those from other LMs. Based on the computed p-values from pre-trained and FPFT LMs, Figure 6 presents the p-value differences between FPFT models and their pre-trained counterparts. A positive difference implies that the p-value of self-evaluation bias is higher in the FPFT model than in the pre-trained model, suggesting that **fine-tuning effectively mitigates LMs' self-evaluation bias**. The Llama family is excluded from this comparison due to their pre-trained versions' near-zero discriminability.

6 Conclusions

Our study investigates factual and causal error detection in financial analysis by LMs. We adopt NLI as the detection paradigm supported by both encoders and LMs. Our experiments show that while LMs outperform encoders in many financial NLP tasks, users should realize their potential disadvantages relative to encoders as well as their susceptibility to biases when evaluating proprietary reasoning. Also, practitioners are advised, although pre-trained models show certain ability, to fine-tune models when resources permit, as it enhances discriminability and mitigate self-evaluation bias.

Limitations

First, we generated 1,872 reasoning points from responses of 3 LMs to 50 positive and 50 negative cases in a public dataset. To further validate our findings, future experiments should extend to additional tasks, a wider range of LMs, and diverse NLI backbone models. Second, the results indicate that LMs exhibit relatively weak performance compared to pre-trained encoders in certain scenarios, likely due to the absence of prompt engineering and reliance solely on the strategy of comparing response probabilities of *"Yes"* and *"No"* (Lattimer et al., 2023). Future work will explore prompt engineering to improve detection accuracy of pre-trained LMs (Shukla et al., 2025). Third, we do not conduct a thorough evaluation against top professionals, but for time-sensitive applications, AI models hold a clear advantage since human experts cannot process thousands of pieces of information within seconds. Last, we do not examine detection methods such as keyword-based approaches, and future work will evaluate whether NLI offers meaningful improvements over these simpler methods.

Ethics statement

This study investigates the factual and causal errors in the reasoning process of LMs within the financial domain. We demonstrate that NLI is a computationally efficient detection paradigm. Our results indicate that its current performance, including leveraging LMs as backbones, remains suboptimal. This aligns with findings by Lasheras and Pinheiro (2025) that even advanced models such as GPT-4o exhibit limited capability in causal reasoning. Additionally, most existing benchmarks for factuality and causality detection are built on English tasks and datasets, often overlooking the pragmatic differences and cultural nuances inherent in other languages (Lasheras and Pinheiro, 2025). Therefore, users are recommended to conduct thorough evaluations before deploying NLI-based detection backbones in real-world applications.

Disclaimer

This paper is provided solely for informational purposes as an academic contribution by the authors to the research community and does not represent, reflect, or constitute the views, policies, positions, or practices of American Express or its affiliates. Nothing in this paper should be cited or relied upon as evidence of, or support for, the business views, policies, positions, or practices of American Express or its affiliates.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Guangsheng Bao, Hongbo Zhang, Cunxiang Wang, et al. 2025. How likely do LLMs with CoT mimic human reasoning? In *Proceedings of the International Conference on Computational Linguistics*.

Alex Chandler, Devesh Surve, and Hui Su. 2024. Detecting errors through ensembling prompts (DEEP): An end-to-end LLM framework for detecting factual errors. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Pulkit Chatwal, Amit Agarwal, and Ankush Mittal. 2025. Enhancing causal relationship detection using prompt engineering and large language models. In *Proceedings of the Joint Workshop of the Financial Technology and Natural Language Processing, the Financial Narrative Processing, and the Workshop on Large Language Models for Finance and Legal*.

Yingjian Chen, Haoran Liu, Yinhong Liu, Jinxiang Xie, Rui Yang, Han Yuan, Yanran Fu, Peng Yuan Zhou, Qingyu Chen, James Caverlee, and Irene Li. 2025. GraphCheck: Breaking long-term text barriers with extracted knowledge graph-powered fact-checking. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14976–14995.

Davide Chicco and Giuseppe Jurman. 2023. The matthews correlation coefficient (mcc) should replace the roc auc as the standard metric for assessing binary classification. *BioData Mining*, 16(1):4.

Zheng Chu, Jingchang Chen, Zhongjie Wang, et al. 2025. Towards faithful multi-step reasoning through fine-grained causal-aware attribution reasoning distillation. In *Proceedings of the International Conference on Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, et al. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Farima Fatahi Bayat, Kun Qian, Benjamin Han, et al. 2023. FLEEK: Factual error detection and correction with evidence retrieved from external knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jinglong Gao, Xiao Ding, Bing Qin, et al. 2023. Is ChatGPT a good causal reasoner? a comprehensive evaluation. In *Findings of the Association for Computational Linguistics*.

Tanya Goyal and Greg Durrett. 2020. Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics*.

Reto Gubelmann, Ioannis Katis, Christina Niklaus, et al. 2024. Capturing the varieties of natural language inference: A systematic survey of existing datasets and two novel benchmarks. *Journal of Logic, Language and Information*, 33(1):21–48.

Yue Guo and Yi Yang. 2024. EconNLI: Evaluating large language models on economics reasoning. In *Findings of the Association for Computational Linguistics*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, et al. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of the International Conference on Learning Representations*.

Hans Hofmann. 1994. Statlog (German Credit Data). UCI Machine Learning Repository.

Bo Hu, Han Yuan, Vlad Pandelea, Wuqiong Luo, Yingzhu Zhao, and Zheng Ma. 2025. Extract, match, and score: An evaluation paradigm for long question-context-answer triplets in financial analysis. In *ICLR 2025 Workshop on Advances in Financial AI: Opportunities, Innovations and Responsible AI*.

Edward J Hu, yelong shen, Phillip Wallis, et al. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*.

Marius Ionescu, Andrei-Marius Avram, George-Andrei Dima, et al. 2020. UPB at FinCausal-2020, tasks 1 & 2: Causality analysis in financial documents using pretrained language models. In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*.

Zhijing Jin, Jiarui Liu, Zhiheng LYU, et al. 2024. Can large language models infer causation from correlation? In *Proceedings of the International Conference on Learning Representations*.

Hamid Reza Kazemi, Kaveh Khalili-Damghani, and Soheil Sadi-Nezhad. 2023. Estimation of optimum thresholds for binary classification using genetic algorithm: An application to solve a credit scoring problem. *Expert Systems*, 40(3):e13203.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, et al. 2022. Large language models are zero-shot reasoners. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Yaxuan Kong, Yuqi Nie, Xiaowen Dong, et al. 2024. Large language models for financial and investment management: Models, opportunities, and challenges. *Journal of Portfolio Management*, 51(2).

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, et al. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Philippe Laban, Wojciech Kryscinski, Divyansh Agarwal, et al. 2023. SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, et al. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10.

Uriel Anderson Lasheras and Vládia Pinheiro. 2025. Calquest. pt: Towards the collection and evaluation of natural causal ladder questions in portuguese for ai agents. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*.

Barrett Lattimer, Patrick H. Chen, Xinyuan Zhang, et al. 2023. Fast and accurate factual inconsistency detection over long documents. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Mike Lewis, Yinhan Liu, Naman Goyal, et al. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Junyi Li, Jie Chen, Ruiyang Ren, et al. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Yinheng Li, Shaofei Wang, Han Ding, et al. 2023. Large language models in finance: A survey. In *Proceedings of the ACM International Conference on AI in Finance*.

Aixin Liu, Bei Feng, Bing Xue, et al. 2024a. Deepseek-v3 technical report. *arXiv*.

Xiao Liu, Zirui Wu, Xueqing Wu, et al. 2024b. Are LLMs capable of data-based statistical and causal reasoning? benchmarking advanced quantitative reasoning with data. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Yinhan Liu, Myle Ott, Naman Goyal, et al. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Machine Learning*.

Jabez Magomere, Elena Kochkina, Samuel Mensah, et al. 2025. FinNLI: Novel dataset for multi-genre financial natural language inference benchmarking. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4545–4568.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, et al. 2024. Gemma: Open models based on gemini research and technology.

Eric Mitchell, Joseph Noh, Siyan Li, et al. 2022. Enhancing self-consistency and performance of pretrained language models through natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Jingwei Ni, Minjing Shi, Dominik Stammbach, et al. 2024. AFaCTA: Assisting the annotation of factual claim detection with reliable LLM annotators. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

OpenAI. Fine-tuning now available for gpt-4o. https://openai.com/index/gpt-4o-fine-tuning/.

OpenAI. 2024. Gpt-4 technical report.

Debjit Paul, Robert West, Antoine Bosselut, et al. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics*.

Angelika Romanou, Syrielle Montariol, Debjit Paul, et al. 2023. CRAB: Assessing the strength of causal relationships between real-world events. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Julia Rozanova, Marco Valentino, Lucas Cordeiro, et al. 2023. Interventional probing in high dimensions: An NLI case study. In *Findings of the Association for Computational Linguistics*.

Aalok Sathe and Joonsuk Park. 2021. Automatic fact-checking with document-level annotations using BERT and multiple instance learning. In *Proceedings of the Workshop on Fact Extraction and VERification*.

Neelesh Kumar Shukla, Sandeep Singh, Prabhat Kumar Prabhakar, et al. 2025. KULFi framework: Knowledge utilization for optimizing large language models for financial causal reasoning. In *Proceedings of the Joint Workshop of the Financial Technology and Natural Language Processing, the Financial Narrative Processing, and the Workshop on Large Language Models for Finance and Legal*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, et al. 2023. Llama: Open and efficient foundation language models.

Miles Turpin, Julian Michael, Ethan Perez, et al. 2023. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. In *Proceedings of the International Conference on Neural Information Processing Systems*.

Prasetya Utama, Joshua Bambrick, Nafise Moosavi, et al. 2022. Falsesum: Generating document-level NLI examples for recognizing factual inconsistency in summarization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Jiaan Wang, Yunlong Liang, Fandong Meng, et al. 2023. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*.

Zeyu Wang. 2024. CausalBench: A comprehensive benchmark for evaluating causal reasoning capabilities of large language models. In *Proceedings of the SIGHAN Workshop on Chinese Language Processing*.

Frank Wilcoxon. 1947. Probability tables for individual comparisons by ranking methods. *Biometrics*, 3(3):119–122.

Junda Wu, Tong Yu, Xiang Chen, et al. 2024. DeCoT: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Jiuding Yang, Hui Liu, Weidong Guo, et al. 2024. Reassess summary factual inconsistency detection with large language model. In *Proceedings of the Workshop on Towards Knowledgeable Language Models*.

Fei Yu, Hongbo Zhang, Prayag Tiwari, et al. 2024. Natural language reasoning, a survey. *ACM Computing Surveys*, 56(12):1–39.

Han Yuan, Feng Xie, Marcus Eng Hock Ong, Yilin Ning, Marcel Lucas Chee, Seyed Ehsan Saffari, Hairil Rizal Abdullah, Benjamin Alan Goldstein, Bibhas Chakraborty, and Nan Liu. 2022. Autoscore-imbalance: An interpretable machine learning tool for development of clinical scores with rare events data. *Journal of Biomedical Informatics*, 129:104072.

Han Yuan, Li Zhang, and Zheng Ma. 2025. Exploring the reliability of self-explanation and its relationship with classification in language model-driven financial analysis. In *ICLR 2025 Workshop on Advances in Financial AI: Opportunities, Innovations and Responsible AI*.

Muru Zhang, Ofir Press, William Merrill, et al. 2024. How language model hallucinations can snowball. In *Proceedings of the International Conference on Machine Learning*.

Lianmin Zheng, Wei-Lin Chiang, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the International Conference on Neural Information Processing Systems*.

A PEFT results

Tables 3 and 4 show PEFT results for both encoders and LMs in detecting factual and causal issues, respectively. With the exception of Llama-3.2-3B and Gemma-2-9B in factuality and causality, LoRA consistently outperforms LLFT. Notably, these two exceptional models exhibit consistent behavior across both tasks, suggesting that LoRA struggles to identify more effective parameters than those in the final layer in some scenarios.

Model	Mode	F1	BA	AUPRC	AUROC
DeBERTa-v3-large	LoRA	0.81	0.91	0.87	0.98
	LLFT	0.19	0.76	0.32	0.85
BART-large	LoRA	0.79	0.89	0.87	0.98
	LLFT	0.45	0.79	0.59	0.93
RoBERTa-large	LoRA	0.85	0.94	0.89	0.99
	LLFT	0.26	0.79	0.40	0.88
Llama-3.2-3B	LoRA	0.11	0.64	0.25	0.69
	LLFT	0.38	0.82	0.28	0.71
Llama-3.1-8B	LoRA	0.52	0.86	0.42	0.93
	LLFT	0.15	0.66	0.19	0.76
Gemma-2-2B	LoRA	0.46	0.73	0.33	0.88
	LLFT	0.22	0.58	0.21	0.82
Gemma-2-9B	LoRA	0.35	0.75	0.14	0.71
	LLFT	0.53	0.79	0.49	0.82
Phi-3.5-mini	LoRA	0.37	0.83	0.51	0.92
	LLFT	0.16	0.64	0.27	0.66
Phi-3.5-MoE	LoRA	0.30	0.83	0.27	0.83
	LLFT	0.14	0.64	0.18	0.69

Table 3: Factuality detection results of PEFT models

B Position bias of LMs

Prior studies have shown that LMs exhibit position bias when making inferences involving swapped answer positions (Zheng et al., 2023). In our context, position bias refers to the effect of presenting prompts in the order of *"Yes"* or *"No"* versus *"No"* or *"Yes"*. To assess the position bias, we perform a chi-squared test on the decisions made by pre-trained models across all samples under two prompt orders. The null hypothesis posits no significant difference between the two variants, while the alternative hypothesis suggests a significant difference. The results indicate that, with the exception of Llama-3.2-3B in factuality detection, and Llama-3.1-8B in both factuality and causality detection, all other models exhibit p-values less than 0.01. This provides strong statistical evidence for the presence of position bias. It is also worth noting that the absence of bias in Llama is attributable to its limited capability, as it generates *"No"* for nearly all samples, resulting in no variation.

To address the position bias, the reported results using either average voting or veto voting across these two prompt variants. For the selection of voting methods, we adopt a heuristic approach: when a model's output is highly skewed toward an answer (i.e., 95% of responses favor one option), we apply veto voting to ensure that the minority is better represented and to encourage output diversity. For outputs that do not exceed this threshold, we use average voting to balance positional effects.

Model	Mode	F1	BA	AUPRC	AUROC
DeBERTa-v3-large	LoRA	0.89	0.95	0.91	0.98
	LLFT	0.37	0.62	0.22	0.60
BART-large	LoRA	0.85	0.93	0.91	0.98
	LLFT	0.57	0.80	0.73	0.90
RoBERTa-large	LoRA	0.89	0.96	0.89	0.99
	LLFT	0.43	0.68	0.42	0.72
Llama-3.2-3B	LoRA	0.30	0.50	0.18	0.49
	LLFT	0.35	0.59	0.28	0.63
Llama-3.1-8B	LoRA	0.56	0.81	0.37	0.77
	LLFT	0.31	0.51	0.22	0.55
Gemma-2-2B	LoRA	0.07	0.49	0.27	0.70
	LLFT	0.04	0.48	0.22	0.61
Gemma-2-9B	LoRA	0.07	0.47	0.21	0.55
	LLFT	0.29	0.55	0.21	0.56
Phi-3.5-mini	LoRA	0.30	0.54	0.17	0.48
	LLFT	0.27	0.47	0.15	0.42
Phi-3.5-MoE	LoRA	0.45	0.68	0.22	0.58
	LLFT	0.32	0.56	0.19	0.55

Table 4: Causality detection results of PEFT models

C Pairwise comparison

In addition to the 2 comparisons in the main text, we provide heatmaps of pairwise comparisons across the remaining 6 scenarios, defined by the combination of setting (pre-trained vs. FPFT), task (factuality vs. causality), and issue (true vs. false). A cell with a significant p-value indicates that the column model outperforms the row model.

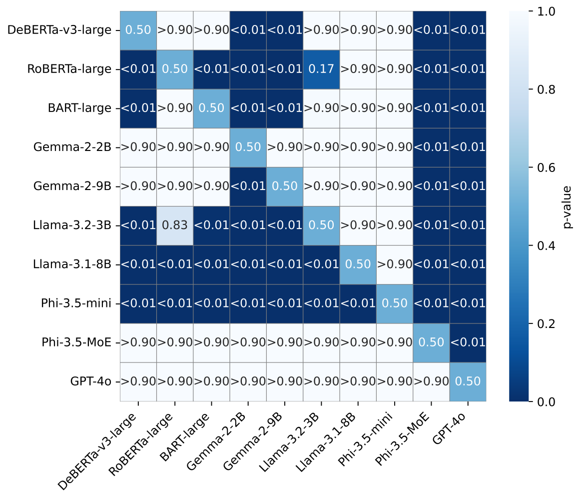


Figure 7: Pairwise comparison of factuality detection on correct reasoning points in the pre-trained setting

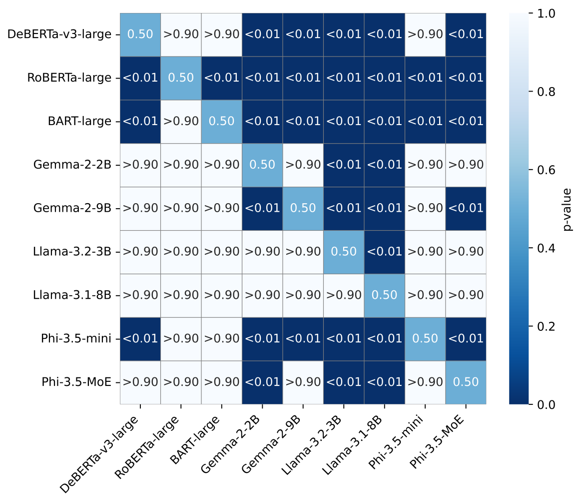


Figure 8: Pairwise comparison of factuality detection on correct reasoning points in the FPFT setting

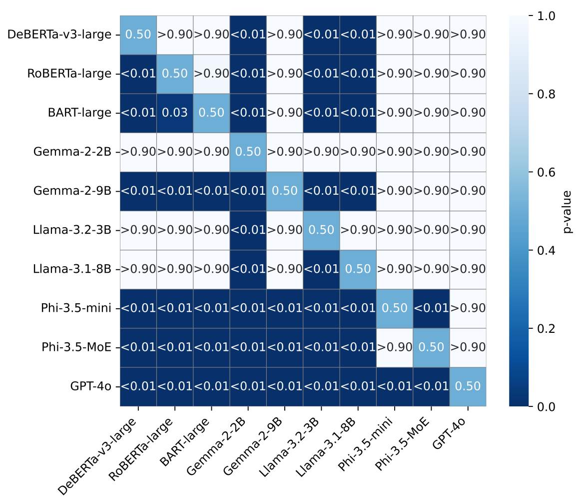


Figure 9: Pairwise comparison of causality detection on correct reasoning points in the pre-trained setting

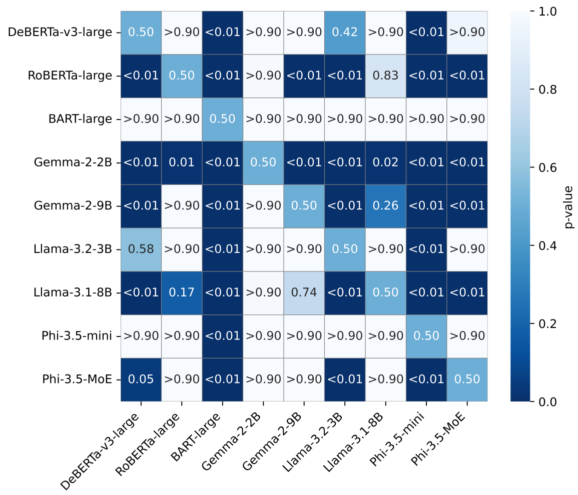


Figure 10: Pairwise comparison of causality detection on correct reasoning points in the FPFT setting

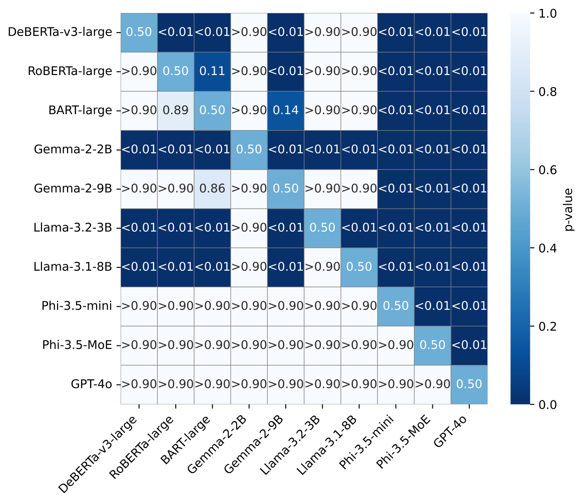


Figure 11: Pairwise comparison of causality detection on erroneous reasoning points in the pre-trained setting

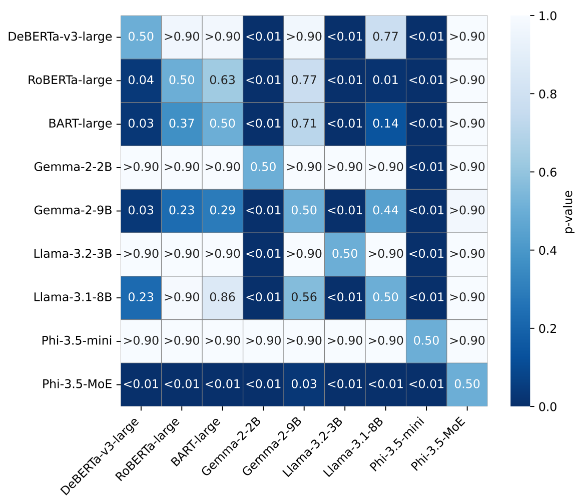


Figure 12: Pairwise comparison of causality detection on erroneous reasoning points in the FPFT setting

SEC-QA: A Systematic Evaluation Corpus for Financial QA

Viet Dac Lai, Michael Krumdick, Charles Lovering
Varshini Reddy, Craig Schmidt, Chris Tanner
Kensho Technologies
{chris.tanner}@kensho.com

Abstract

The financial domain frequently deals with large numbers of long documents that are essential for daily operations. Significant effort is put towards automating financial data analysis. However, a persistent challenge, not limited to the finance domain, is the scarcity of datasets that accurately reflect real-world tasks for model evaluation. Existing datasets are often constrained by size, context, or relevance to practical applications. Moreover, LLMs are currently trained on trillions of tokens of text, limiting access to novel data or documents that models have not encountered during training for unbiased evaluation. We propose SEC-QA, a continuous dataset generation framework with two key features: 1) the semi-automatic generation of Question-Answer (QA) pairs spanning multiple long context financial documents, which better represent real-world financial scenarios; 2) the ability to continually refresh the dataset using the most recent public document collections, not yet ingested by LLMs. Our experiments show that current retrieval augmented generation methods systematically fail to answer these challenging multi-document questions. In response, we introduce a QA system based on program-of-thought that improves the ability to perform complex information retrieval and quantitative reasoning pipelines, thereby increasing QA accuracy.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities in a wide range of natural language processing (NLP) applications (Brown et al., 2020). Even though LLMs were scaled to an unprecedentedly large number of parameters, they still face many issues with hallucinations (Ji et al., 2023), poor reading comprehension (Liu et al., 2024), and private data leakage (Balloccu et al., 2024). To address these issues, Retrieval Augmented Generation (RAG) produces responses by using a few retrieved documents from reliable sources. As a result, it offers more dependable answers with fewer hallucinations, even when employing much smaller language models (Borgeaud et al., 2022). Evaluating RAG-based systems is challenging due to the added complexity of the retrieval step (Chen et al., 2024), and the cascading effect of the upstream retrieval task on the downstream QA task. Early benchmarks for RAG-based systems mainly focus on simple common-sense questions that can be answered by retrieving a single piece of text from a single knowledge source (Joshi et al., 2017; Dunn et al., 2017). Often these knowledge sources were open textual knowledge bases such as Wikipedia (Yang et al., 2018) and WikiHow (Deng et al., 2020).

These benchmarks face two data leakage issues. First, they were created from open Internet sources (Xue et al., 2021; Penedo et al., 2023), e.g., Wikipedia, which is heavily used in LLM pretraining (Touvron et al., 2023). Second, the static benchmarks themselves are leaked to the Internet (Balloccu et al., 2024). LLM training datasets subsequently include these benchmarks, leading to inflation and unreliable results on these benchmarks. To combat these problems, there have been proposals to keep benchmarks private (Mialon et al., 2023) or updated regularly (Fan et al., 2023).

More importantly, in many financial applications, evaluation based on common-sense knowledge and open knowledge sources may not reflect the system's true capabilities (Zhang et al., 2024). The questions in professional use are far more complex, requiring some combination of multi-hop reasoning (Yang et al., 2018), multi-source reference (Tang and Yang, 2024), document-structure reference (Saad-Falcon et al., 2023), and collection structure reference. Additionally, knowledge bases in a specific domain may be less diverse than open knowledge bases, e.g., Wikipedia, leading to poor retrieval performance which in turn has a severe

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

impact on the LLM's response. Retrieval in some domain-specific areas may instead require domain knowledge to achieve effective retrieval that has not been captured in open-domain evaluation.

To overcome these issues, this work introduces a framework for designing practical quantitative questions. These questions are much more challenging than existing financial QA tasks such as FinQA(Chen et al., 2021) or TAT-QA (Zhu et al., 2021). The framework allows us to customize questions at the needed complexity for the target applications with potential variety in question complexity including *multiple entities/financial periods*, *multi-hop reasoning*, *document structure*, *collection structure*, and *multiple outputs*. We leverage Internet-accessible document collections, and open tabular databases to create real-world complex quantitative questions in finance. We evaluate four RAG-based systems and show that RAG systems systematically fail on these carefully designed real-world questions. Moreover, we show that recent LLMs can use code to effectively navigate the structure of the document collections; e.g., that earnings per share information is within 10-Ks, and that there are 10-Ks for each company and fiscal year. This leads to drastically improved levels of performance. Additionally, this framework can be used to dynamically refresh the benchmarks regularly to prevent training data leakage.

The contributions of this paper are:

- A framework (SEC-QA) for dynamically generating quantitative multi-hop QA tasks for the financial domain from publicly accessible documents and databases.
- A set of practical and challenging questions for Quantitative Reasoning in the financial domain that vanilla RAG models systematically fail to answer.
- A system that utilizes program-of-thought and the rich structure of the document collection to improve QA performance.

2 Related Work

Reasoning Capabilities of LLM has been found in natural language processing and other fields. Some LLMs exhibit emergent capabilities if they are large enough. A simple prompt "*Let's think step by step*" causes a model to generate solutions with reasoning steps in a chain-of-thought (Wei et al., 2022). More advanced prompting techniques have been discovered which are similar to human reasoning processes such as tree-of-thoughts (Yao et al., 2024) and self-verification (Weng et al., 2023). However, the numerical reasoning of LLMs is still limited, motivating the adoption of programming languages to offload numerical tasks in program-of-thought (Chen et al., 2023) and program-synthesis (Austin et al., 2021). Reasoning capability is enhanced in additional training on reasoning through human feedback (Ouyang et al., 2022). However, LLMs still struggle with many domain-specific tasks such as finance (Koncel-Kedziorski et al., 2023). Recent studies have pointed out that many popular benchmarks are contained in LLM pre-training data (Riddell et al., 2024), which causes the inflation of model performance. As such, benchmarks are kept private (Mialon et al., 2023), or updated regularly (Fan et al., 2023) to mitigate data contamination.

Document Grounded Quantitative Reasoning involves numerical extraction and numerical reasoning. Previous work in NLP has explored numerical extraction from scientific documents (Harper et al., 2021; Elazar et al., 2019) and financial documents (Loukas et al., 2022). Existing datasets for the financial domain that require quantity extraction include HybridQA (Chen et al., 2020), TAT-QA (Zhu et al., 2021), MultiHierTT (Zhao et al., 2022), FinQA (Chen et al., 2021), and ConvFinQA (Chen et al., 2022). However, these works only involve a small amount of grounding context (e.g., a single page, a single document).

Multi-Document QA: TriviaQA (Joshi et al., 2017) and SearchQA (Dunn et al., 2017) require the model to search over a large collection of documents. However, the question itself can be answered by reading a few sentences extracted from a single document. Some multi-document QA datasets were created for open-ended QA such as summarization (MultiNews (Fabbri et al., 2019), WikiHowQA (Bolotova-Baranova et al., 2023)) where skimming over given documents and extracting evidential cues from these documents are essential. The HotpotQA (Yang et al., 2018) dataset specifically targets multi-hop questions to resolve hidden cross-document reference entities in the questions. However, these datasets were collected from open knowledge bases (e.g., Wikipedia), so they have most likely been leaked in LLM pre-training data (Touvron et al., 2023).

Multihop-RAG (Tang and Yang, 2024) propose

	Dataset	Multi-Doc	Multi-Hop	Refreshable	#Test	Context	#Docs	Data
General	HybridQA	✓	✓	-	3,463	2,326	44	Hybrid
	TriviaQA	✓	-	-	17,210	3,760	486,956	Text
	SearchQA	-	-	-	27,248	38	-	Text
	HotPotQA	✓	✓	-	7,405	928	15,519	Text
Finance	TAT-QA	-	-	-	1,669	47	-	Hybrid
	MultiHiertt	-	-	-	1,566	1,646	-	Hybrid
	FinQA	-	-	-	1,147	628	-	Hybrid
	ConvFinQA	-	-	-	434	628	-	Hybrid
	Multihop-RAG	✓	-	-	2,556	1,574	609	Text
	SEC-QA	✓	✓	✓	Flexible	123,000	1,315	Hybrid

Table 1: Comparison of QA benchmarks in the quantitative and finance domain. Refreshable indicates that the dataset can be automatically renewed/generated with a different document set. Hybrid indicates that the context contains both tabular and textual data.

a multi-hop dataset for financial documents that differs from our work in the following ways: (i) their work studies multi-hop reasoning only with regards to parallel retrieval queries, while we consider both parallel and sequential reasoning steps; (ii) the news documents used to create their dataset do not reflect the real-world sources such as official filings used by financial professionals, that often span hundreds of pages.

Financial NLP has explored non-quantitative tasks such as named entity recognition (Salinas Alvarado et al., 2015), sentiment analysis (Malo et al., 2014), classification (Sinha and Khandait, 2021), question answering (Maia et al., 2018), boundary detection (Au et al., 2021), and entity/event extraction Lu et al. (2023).

3 Framework Construction

We propose Systematic Evaluation Corpus for Financial QA (SEC-QA), a framework for generating financial Multi Document Questions and Answers (MDQA). We also refer to the questions generated by this framework with the same name, SEC-QA.

3.1 Task Definition

MDQA is defined as follows: A system S is asked a question q with answer a. q can be answered by looking in at the document collection $C = \{D_i | 1 \leq i \leq N\}$. Each document consists of several pages $p_i = (t_{ij}, c_{ij})$ where t_{ij}, c_{ij} are the title and the content, respectively.

3.2 Resources

SEC-QA, a framework for flexibly creating questions for MQDA, requires: 1) Database T with values partitioned by variables of interest (e.g., revenue partitioned by company and fiscal year). 2) Document collection C that contains the information needed to compute the values within T.

Specifically, we leverage private-sector financial data from market-trusted sources, ensuring comprehensive and accurate datasets. We collect key financial metrics and their associated documents to create a database (key-value-document table) $T \in (c, y, k, v, d)$, where v represents the value of the key metric k for company c in fiscal year y as mentioned in document d. Because T tracks the documents in C that source each value then in addition to question accuracy it is possible to report document and page-level retrieval metrics.

Our collection comprises 10 metrics for 18 publicly traded companies from the S&P 500 list from 2010 to 2023. We also collect their annual reports (Form 10-K), quarterly reports (Form 10-Q), and unscheduled event reports (Form 8-K) for the same period. Since the documents are published in HTML format, we convert the documents into PDF(s), and then we parse the PDF(s) into JSON format using a public PDF extraction service [1]. Documents are represented as JSON objects with paragraphs, well-structured tables, and machine-detected titles.

[1] https://kensho.com/extract

3.3 Question Complexity

Question design is an important step for a successful QA system evaluation. In this work, we identify several factors that increase the complexity of a question in the financial domain.

We define an **atomic question** as a question that seeks a single piece of information that can be extracted directly from a single document. Such questions usually involve a single entity for a particular financial period, e.g., *"What is the total revenue of Apple Inc. for the fiscal year 2022?"*.

However, in financial analysis, questions tend to be much more complex. They require extracting multiple pieces of data, transforming the extracted data, and presenting the answer in various formats. Further, we have observed the following main challenges in multiple-document QA for finance:

- **Parallel Reference** questions require the same kind of information over multiple entities or time periods, e.g., *"What is Intel's revenue growth in the last 5-year period?"*. This requires extracting information from a few documents. The complexity of parallel reference can be measured by the number of entities/years needed to answer the question.
- **Multi-hop Reference** questions require reference resolution of a few implicitly defined entities through some deterministic constraints e.g. *"Show the 5-year stock price history of the top 5 most valuable companies in the S&P 500 index"*. The complexity of multi-hop questions is usually measured by the number of hops needed to answer the question and the complexity of the constraints.
- **Structural Reference** involves document structure reference and collection structure reference. Document structure reference refers to a particular section/table/figure in a document (Saad-Falcon et al., 2023). Collection structure reference refers to a subset of documents in the collection (e.g., *"recently quarterly filings"*, *"their earning calls"*), narrowing the document search space.
- **Multi-Output** questions expect multiple values (e.g., *"Analyze the financial performance of the top 3 competitors of Amazon.com, Inc. in terms of revenue for the last 4 quarters, by computing their revenue growth, gross margin, and operating margin."*. To our knowledge, this type of question has not been addressed in previous work.

3.4 Question Design

Question Template: Previous work in QA has tackled some challenges in QA such as multi-hop (Yang et al., 2018), and document structure (Saad-Falcon et al., 2023) to target some specific complex question types. Our framework allows many question types including parallel reference, multi-hop reference, document structure, collection structure reference, and multi-output questions. Moreover, due to the rich information in the database, we can design questions to require finance domain jargon (e.g., using a company's stock symbol to refer to the company) and language regularities (e.g., omitting financial periods if asking for the latest figures). Using this framework, we can flexibly opt for different combinations of complexity (e.g., parallels and multi-hop in the same question).

Template filling: The question template can be filled semi-automatically through a simple rule-based system that randomly picks the entities, metrics, and periods from the database. This allows control over the complexity, quality, and distribution of the generated questions and answers.

4 Experiment

This section presents our findings across three use cases of the SEC-QA framework.

4.1 QA Systems

We evaluate 4 systems with different characteristics for a broad understanding of MDQA in finance:

- **Vanilla RAG**: To demonstrate the challenge of answering complex questions, we employed a simple retrieval-based system with direct generation.
- **Multi Query RAG**: leverages multiple queries for a given input question.
- **CodeGen+PageR**: We employ an LLM to generate code that makes use of two helper functions: *"retrieve_relevant_pages"* retrieves k pages from the whole collection; *"extract_value"* calls a prompted LLM to extract the value from the given retrieved pages. This system allows an LLM to decompose a complex question into atomic questions as well as make a full plan of how to answer this question. In turn, it allows us to examine the planning capability of the LLM.
- **CodeGen+DocS+PageR**: Financial document collections track meta information for

Function	Description
select_document(stock_symbols, form_types, fiscal_years)	Select a few documents given some filters
retrieve_relevant_pages(text_query, documents)	Retrieve top-k pages from the given documents
extract_value(text_query, pages)	Extract a value and its corresponding multiplier from the given pages

Table 2: List of helper functions that the CodeGen systems can use

each document like company, fiscal year, and form type. This can be used to reduce the retrieval search space. Beyond the two functions *"retrieve_relevant_pages"* and *"extract_value"*, we introduce a *"select_document"* function that filters the document collection based on meta information such as the company stock symbol and fiscal year. Critically, a document can be selected and then queried for pages.

We use OpenAI's Ada as the neural embedding for retrieval and GPT4 (gpt-4-1106-preview) as the LLM for all our experiments. We use the same three questions as the exemplars for the CodeGen system, varying only the available helper functions. We test different numbers of retrieved pages ($k \in [4, 128]$) and report the best performance.

4.2 Evaluation

Due to the complexity of the pipeline and models we used, we evaluate the model's performance in 3 stages: document retrieval, page retrieval, and question answering. For document retrieval and page retrieval, we report the Precision@K, Recall@K, and F1@K. For systems that only involve page retrieval, we report their document retrieval performance based on the document signature of the retrieved pages. Since many numbers in financial reports are rounded to various levels (thousands, millions, billions), using exact matches for automatic answer scoring is challenging. Therefore, we accept an answer if its value is within a 1% margin of error from the golden answer.

4.3 Single Value Extraction Task

We begin with a simple extraction MDQA task that requires a model to retrieve an exact numeric span (e.g., 1234.5) and unit value (e.g., millions) from a document. For this use case, we use SEC-QA to generate questions based on the following templates: (1) is usually used for the latest update of a metric; (2) is used for the previous financial periods (e.g., years and quarters).

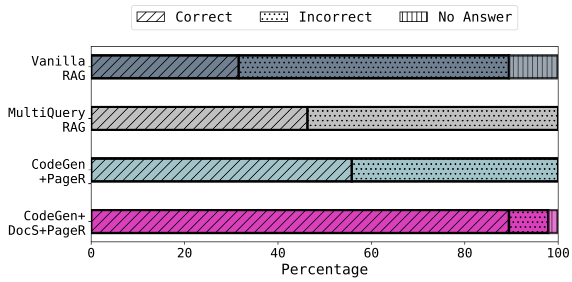


Figure 1: Performance of the simple value extraction

(1) *What is {**company**}'s {**metric**}?*

(2) *What is {**company**}'s {**metric**} in {**year**}?*

Financial analysts use this language regularly, in which the year is omitted from the question. As such, a model must catch that regularity to identify the correct value term to extract. To confirm the existence of the metric in the document, we remove the questions whose answers can not be found in the document with simple string matching.

Figure 1 shows the performances of the models of this single-document value extraction task. CodeGen+DocS+PageR system performs best, with 89.5% accuracy. On the other hand, the CodeGen+PageR and Vanilla-RAG systems lag behind with 31.6% and 55.8% accuracy. Comparing the two CodeGen models, we can see that the CodeGen w/ DocS outperforms the variant w/o DocS. This suggests that document selection is important for financial questions. This result also shows that a general neural-based document retrieval struggles to cope with the demanding requirements for retrieval in finance. Part of the reason the neural-based retrieval struggles is because of the structure of the public filings in the financial domain. Many documents are very similar to each other, especially documents for the same company. Long identical phrases are often used for multiple years. Without document selection, LLMs end up having to process irrelevant pages collected from previous years. Section 5.2 shows an example of models without document selection. This experiment shows there is a large performance difference for different pipeline settings.

System	Correct (%)
Vanilla RAG	8.3
Multi Query RAG	16.7
CodeGen+PageR	37.5
CodeGen+DocS+PageR	33.3

Table 3: Performance on compound value extraction.

4.4 Compound Value Extraction Task

Compound or high-order metrics appear frequently in financial analysis. They are usually computed based on a few other metrics reported in the public filings. As such, being able to answer questions with compound metrics is crucial to automating the pipeline of financial analysis. Previous work considers these questions as *NULL* (Tang and Yang, 2024), skipping the question. In real-world applications, a model should provide its best estimation based on the provided information, ideally with an explanation to justify the estimation.

To do this, we design a set of questions with compound metrics, such as Revenue Per Employee (RPE) using the same Templates 1, 2. Some companies report these metrics in their filings, so we only consider metrics that cannot be easily found with string matching. Thus, the model must be able to understand the formula to calculate to answer correctly. In total, we generated 24 of these questions. Because the value is not explicitly in the text, these questions require both *a priori* knowledge of the metrics and additional reasoning steps to compute the metric. Therefore, we expect these questions to be more difficult.

Table 3 shows the performance of the model on this question set. Performance for all tested systems on these Compound Value Extraction questions is significantly lower than the Single Value Extraction questions (as presented in Section 4.3).

For this task, we are able to investigate the performance at the sub-metric level thanks to the sub-metric data in the database. The Vanilla RAG and Multi Query RAG models do not extract value at the sub-metric level, so we omit these models from this analysis. We find that CodeGen-based models systematically query each sub-metric (e.g., Long-Term Debt, and Long-Term Leases). However, once the model unrolls the main metric (e.g., total debt) into sub-metrics, many sub-metrics are missing from the document due to two reasons: (1) the sub-metrics are also compound metrics and (2) some sub-metrics do not apply to some companies. These lead to a false extraction or duplication (Long-Term Debt being extracted twice for Long-Term Debt and Long-Term Leases).

This test highlights the difficulty of the value extraction task in the financial domain. This also shows how we can easily use our framework to customize the test without data annotation.

4.5 Multi-Document QA Task

To measure the performance on more complex question, we design a set of question templates that refer to metrics of different years and companies as follows:

(3) *How much common dividends did* **{company}** *pay in the last* **{num_year}** *years in US dollars?*

(4) *What is the percentage difference of* **{company1}***'s* **{metric}** *compared to that of* **{company2}***?*

(5) *What is* **{company}***'s overall revenue growth over the last* **{num_year}***-year period?*

(6) *Among* **{company_names}**, *what is the* **{metric2}** *of the company that has the highest* **{metric1}***?*

Figure 2 shows the accuracy for the three models on this task with different numbers of retrieved pages k. We can see that CodeGen+DocS+PageR outperforms the other models with a high margin, correctly answering 108 of 135 questions ($80\%, k = 32$) compared to 71 with CodeGen+PageR ($52\%, k = 48$) and only 41 with Vanilla RAG ($30\%, k = 32$). Notably, CodeGen+DocS+PageR outperforms the other two models with a very low number of retrieved pages ($k = 4$) with 74 correct answers (55%).

More importantly, CodeGen models are more responsive to the number of retrieved pages. CodeGen+DocS+PageR performance improves rapidly when the number of retrieved pages k increases from 4 to 32, whereas CodeGen+PageR improves at a lower rate, and Vanilla RAG barely improves. This suggests that the Vanilla RAG pipeline is bottlenecked at the retrieval step, which we analyze in depth in Section 5.1.

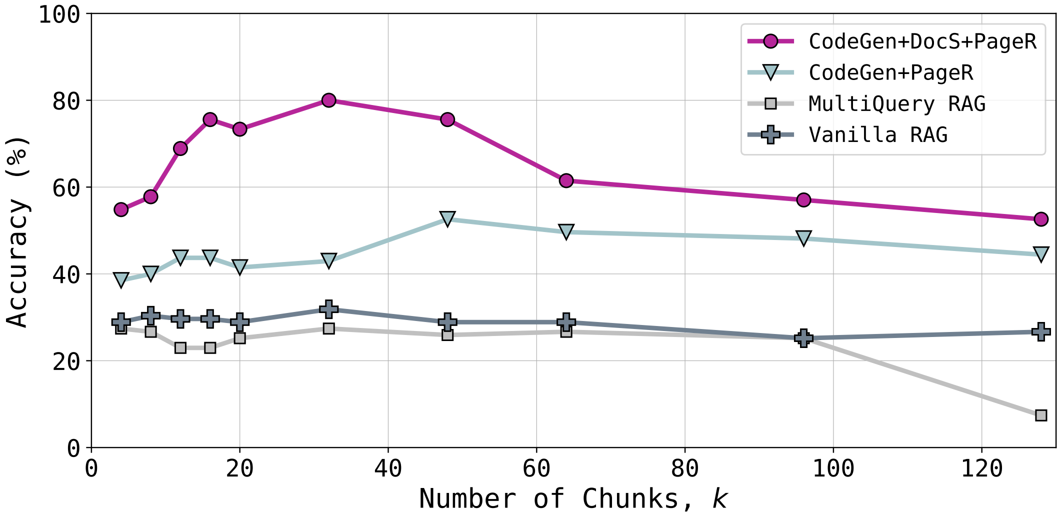


Figure 2: Performance of systems on Multi-Doc QA.

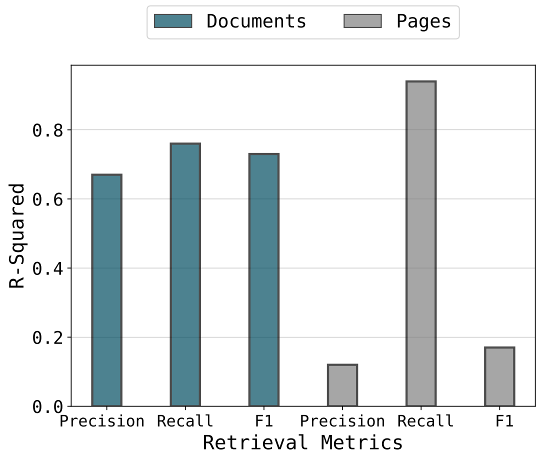


Figure 3: R^2 between accuracy and retrieval metrics.

5 Discussion

5.1 System Bottlenecks

Previous sections highlight how additional retrieval capabilities improve LLM performance on MDQA. In this section, we perform further analysis to identify the main performance bottleneck across the different systems. We reuse the Template 6 which has both parallel and multi-hop references. Beyond document/page retrieval and task performance, we also compute the coefficient of determination R^2 between these performances.

Figure 3 shows the R^2 values between the system accuracy and retrieval metrics at both document and page levels. Recall has the highest R^2 values, with 0.94 and 0.76 for page and document levels respectively, suggesting that recall performance is crucial to the accuracy of the multi-document QA performance. On the other hand, page-level precision has a low R^2 value of 0.12, indicating a weak correlation with overall performance.

From Figure 4, we observe that the QA accuracy of a system is directly proportional to document and page-level recall scores, supporting our previous claim about the correlation between recall and accuracy. Specifically, we note that a model's accuracy aligns more closely with the page-level recall score. The Vanilla RAG model lags behind the CodeGen-based models in terms of accuracy. However, when it is allowed to use multiple retrieval queries (Multi Query RAG), we observe an improvement in retrieval performance, which consequently leads to an increase in QA accuracy.

CodeGen+PageR has a higher recall and QA performance compared to Multi Query and Vanilla RAG. This is due ability of the CodeGen models to break down complex questions into atomic ones and systematically retrieve pages based on the atomic questions. CodeGen+DocS+PageR is observed to be the best model. We attribute this to the addition of the rule-based document selection step, which effectively retrieves relevant documents, thus improving document-level recall.

Overall this experiment shows that our dataset and framework provide us a useful tool to examine multi-document QA in detail. This gives us a better signal for future improvement of the pipeline.

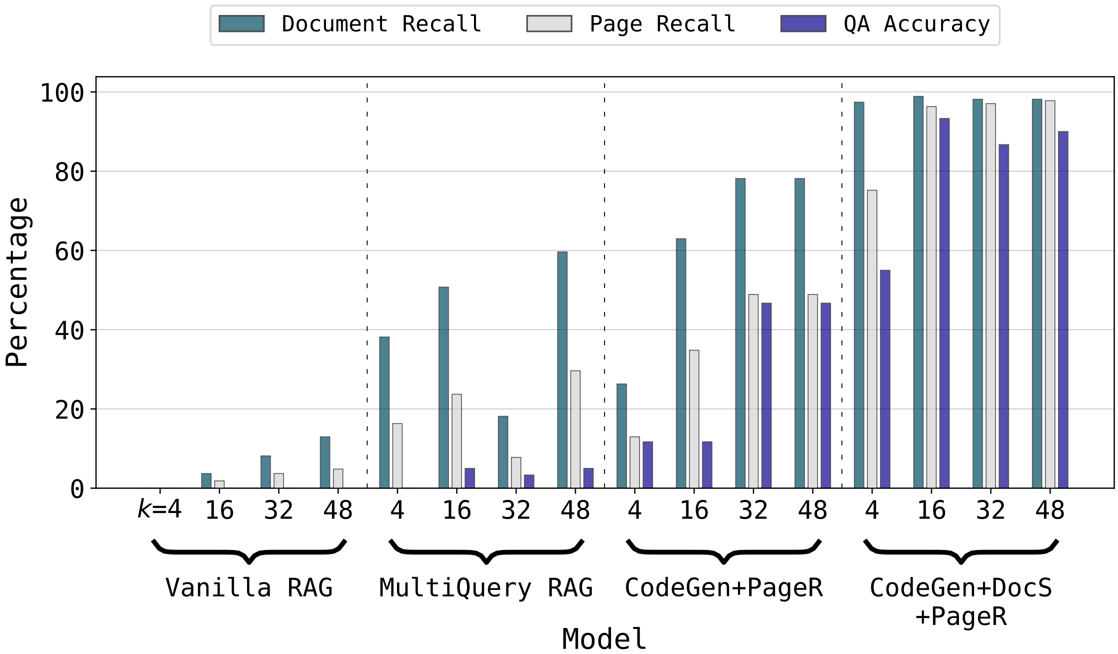


Figure 4: Document-level recall, page-level recall, and QA accuracy of all examined systems with varying k values.

5.2 Case study

Table 4 shows the top 4 retrieved pages across QA systems for *"What is Adobe's total number of employees reported in 2022?"*. Only CodeGen+DocS+PageR successfully retrieves the page containing the golden answer. Although Vanilla RAG retrieves Adobe's 10-K pages 3 of 4 times, the fiscal year is consistently wrong. Multi Query RAG and CodeGen+PageR also retrieve the wrong fiscal year. This shows how financial documents can easily confuse modern retrieval systems.

5.3 Stability Tests

One of the main objectives of this work is to provide a robust benchmark that prevents performance inflation through data leaks by dynamically generating QA pairs using the latest financial documents. This raises the question of whether evaluation scores on a new benchmark version are comparable to those on former versions. To measure the benchmarking consistency, we construct five distinct versions of the dataset using data from different years (2019 to 2023) while keeping the question templates and the set of companies constant to maintain a consistent difficulty level.

The accuracy of the models on different year-based sets varied marginally with small standard deviation ($1.3\% < \sigma < 2.0\%$). This shows that we can reliably compare the performance of models between different versions of the benchmark. We show the detailed result in Figure 5 (Appendix **??**).

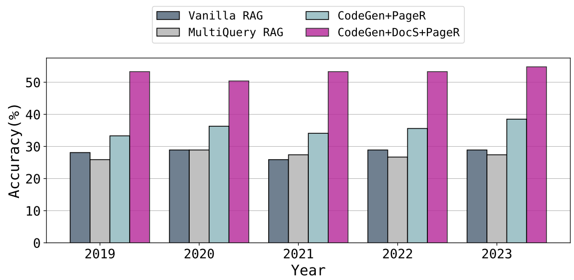


Figure 5: Performance of models on different variants of the same dataset.

5.4 Execution Cost

While code generation-based systems offer superior performance, these models have a higher operational cost and latency. We compute the average number of LLM calls across the four examined systems. From Figure 6, CodeGen systems require approximately four times more calls than Vanilla and Multi Query RAG systems, which can potentially quadruple operational costs and latency. One proposed solution to mitigate latency is to parallelize LLM calls. However, the iterative nature of multi-hop questions poses a challenge to effective parallelization strategies.

Model	Matched	Question: *“What is Adobe’s Total Employees reported in 2022?”*
Vanilla RAG	N	Page: 15 Form type: 10-K; Company: adobe; Fiscal year: ~~**2015**~~; Period end date: 2015-11-27
	N	Page: 15 Form type: 10-K; Company: adobe; Fiscal year: ~~**2014**~~; Period end date: 2014-11-28
	N	Page: 29 Form type: ~~**8-K**~~; Company: adobe; Fiscal year: ~~**2020**~~; Period end date: 2020-12-07
	N	Page: 35 Form type: 10-K; Company: adobe; Fiscal year: ~~**2012**~~; Period end date: 2012-11-30
Multi Query RAG	N	Page: 15 Form type: 10-K; Company: adobe; Fiscal year: ~~**2015**~~; Period end date: 2015-11-27
	N	Page: 15 Form type: 10-K; Company: adobe; Fiscal year: ~~**2014**~~; Period end date: 2014-11-28
	N	Page: 35 Form type: 10-K; Company: adobe; Fiscal year: ~~**2012**~~; Period end date: 2012-11-30
	N	Page: 18 Form type: 10-K; Company: adobe; Fiscal year: ~~**2020**~~; Period end date: 2020-11-27
CodeGen + PageR	N	Page: 15; Form type: 10-K; Company: ADBE; Fiscal year: ~~**2015**~~; Period end date: 2015-11-27
	N	Page: 18; Form type: 10-K; Company: ADBE; Fiscal year: ~~**2023**~~; Period end date: 2023-12-01
	N	Page: 16; Form type: 10-K; Company: ADBE; Fiscal year: ~~**2021**~~; Period end date: 2021-12-03
	N	Page: 15; Form type: 10-K; Company: ADBE; Fiscal year: ~~**2014**~~; Period end date: 2014-11-28
CodeGen + DocS + PageR	N	Page: 16; Form type: 10-K; Company: ADBE; Fiscal year: 2022; Period end date: 2022-12-02
	N	Page: 2; Form type: 10-K; Company: ADBE; Fiscal year: 2022; Period end date: 2022-12-02
	Yes	Page: 15; Form type: 10-K; Company: ADBE; Fiscal year: 2022; Period end date: 2022-12-02
	N	Page: 38; Form type: 10-K; Company: ADBE; Fiscal year: 2022; Period end date: 2022-12-02

Table 4: Case study on the inclusion of the page containing the golden answer within the top 4 retrieved pages for the question, *"What is Adobe’s total number of employees reported in 2022?"*

5.5 Automatic Question Design

Thanks to the advancement in code generation, in principle an LLM with access to a database can generate both questions and code necessary to answer the questions. To do that, one can prepare available resources such as documents and functions to access the database and static variable names for relevant entities (e.g. company name, metric name); next, prompt the LLM to generate questions and code that solve the question using the provided functions. Once the LLM generates questions and answers, the generated code is executed to obtain the answer. After that, one can hire a financial expert to verify the question, the code, and the answer. Notably, the challenge remains at test time as that model will *not* have access to the database T, only the document collection C.

The benefit of using this method is the potentially higher diversity in question sets. It also provides the code that is used to solve the question. However, this method still requires manual inspection of the question, the code, and the alignment between the question and the code, which are not trivial. In this work, we did not use this method to generate any of the questions used above.

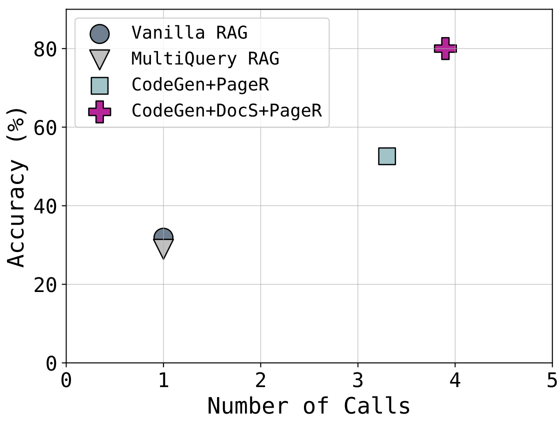


Figure 6: Average number of LLM calls used by the systems in comparison with accuracy.

6 Conclusion

We introduce SEC-QA which we leverage to create questions that current RAG approaches consistently fail to answer. This framework can be used to dynamically generate complex practical questions grounded in the financial domain. Our study highlights the challenges posed by retrieval models in handling multi-doc long-context questions and explores strategies to address these bottlenecks. Furthermore, we propose a method based on program-of-thought and RAG designed to enhance retrieval and downstream performance compared to conventional RAG systems.

Limitation

This paper assumes the existence of a collectible set of documents, a tabular dataset of financial metrics, and a method to map these financial metrics to the documents. We currently explore databases in the private sector, where public reports are heavily regulated, making it relatively straightforward to align the documents with the dataset.

However, in the public sector, reports often vary significantly due to inconsistencies in reporting standards. As a result, finding a collection of documents, a corresponding dataset and their alignment is more challenging. For instance, our attempts with the US state government's Annual Comprehensive Financial Report (ACFR) and the US Annual Survey of State Government Tax Collections published by the US Census have proven extremely difficult to reverse-engineer into a usable dataset.

Ethical Consideration

This dataset was generated automatically from an existing financial database without any involvement of human annotators. Although the CodeGen systems demonstrate significant performance improvements, we do not recommend using them as a replacement for traditional financial analysis tools and financial advice.

References

Willy Au, Abderrahim Ait-Azzi, and Juyeon Kang. 2021. Finsbd-2021: the 3rd shared task on structure boundary detection in unstructured text in the financial domain. In *Companion Proceedings of the Web Conference 2021*, pages 276–279.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Valeriia Bolotova-Baranova, Vladislav Blinov, Sofya Filippova, Falk Scholer, and Mark Sanderson. 2023. WikiHowQA: A comprehensive benchmark for multi-document non-factoid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5291–5314, Toronto, Canada. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2022. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yang Deng, Wai Lam, Yuexiang Xie, Daoyuan Chen, Yaliang Li, Min Yang, and Ying Shen. 2020. Joint learning of answer selection and answer summary generation in community question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7651–7658.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. 2019. How large are lions? inducing distributions over quantitative attributes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3973–3983, Florence, Italy. Association for Computational Linguistics.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Lizhou Fan, Wenyue Hua, Lingyao Li, Haoyang Ling, Yongfeng Zhang, and Libby Hemphill. 2023. Nphardeval: Dynamic benchmark on reasoning ability of large language models via complexity classes. *arXiv preprint arXiv:2312.14890*.

Corey Harper, Jessica Cox, Curt Kohler, Antony Scerri, Ron Daniel Jr., and Paul Groth. 2021. SemEval-2021 task 8: MeasEval – extracting counts and measurements and their related contexts. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 306–316, Online. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Michael Krumdick, Viet Lai, Varshini Reddy, Charles Lovering, and Chris Tanner. 2023. Bizbench: A quantitative reasoning benchmark for business and finance. *arXiv preprint arXiv:2311.06602*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12.

Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Georgios Paliouras. 2022. FiNER: Financial numeric entity recognition for XBRL tagging. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4419–4431, Dublin, Ireland. Association for Computational Linguistics.

Dakuan Lu, Jiaqing Liang, Yipei Xu, Qi He, Yipeng Geng, Mengkun Han, Ying Xin, Hengkui Wu, and Yanghua Xiao. 2023. Bbt-fin: Comprehensive construction of chinese financial domain pre-trained language model, corpus and benchmark. *ArXiv*, abs/2302.09432.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: financial opinion mining and question answering. In *Companion proceedings of the the web conference 2018*, pages 1941–1942.

Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. Gaia: a benchmark for general ai assistants. *arXiv preprint arXiv:2311.12983*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for ccon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Martin Riddell, Ansong Ni, and Arman Cohan. 2024. Quantifying contamination in evaluating code generation capabilities of language models. *arXiv preprint arXiv:2403.04811*.

Jon Saad-Falcon, Joe Barrow, Alexa Siu, Ani Nenkova, Ryan A Rossi, and Franck Dernoncourt. 2023. Pdftriage: Question answering over long, structured documents. *arXiv preprint arXiv:2309.08872*.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.

Ankur Sinha and Tanmay Khandait. 2021. Impact of news on the commodity market: Dataset and results. In *Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 2*, pages 589–601. Springer.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. 2024. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Model Details

Throughout the whole study, we use OpenAI's Ada Embedding to encode texts for retrieval. We use the `Langchain` implementation of vector-based retrievers (i.e., vanilla and multi-query retrievers). We use GPT-4 (gpt-4-0125-preview) as the LLM for both the question answering in RAG-based systems and the code generation and value extraction in CodeGen-based systems.

B Example of code generated by the CodeGen

Figure 7 shows an example of code generated by the CodeGen+DocS+PageR system. Figure 8 and 9 shows the prompt and few-shot demonstrations used to generate code for the CodeGen+DocS+PageR systems.

C Example of question with metric description

What is the Boeing's Total Debt in 2023? Where Total Debt is defined as:

Total Debt is a supplemental line item across all templates with the following components:

Standard template:

Short-term Borrowings

Current Portion of Long-Term Debt

Current Portion of Leases

Long-Term Debt

Long Term Leases

Finance Div. Debt Current

Finance Div. Debt Non-Curr.

Banks template:

Short-Term Borrowings - (Bank Template)

Current Portion of Long-Term Debt - (Bank Template)

Current Portion of Leases

Long-Term Debt

Federal Home Loan Bank Debt - LT

Long Term Leases

Trust Pref. Securities

All other templates:

Short-Term Borrowings - (Template Specific)

Curr. Port. of LT Debt

Current Portion of Leases

Long-Term Debt

Long Term Leases

Trust Pref. Securities

```python
# Among Honeywell, Caterpillar, Pfizer, PepsiCo, and Boeing,
# What is the Total Revenues of the company that has the lowest Total Debt?

from functions import (
  select_documents,
  retrieve_relevant_pages,
  extract_value
)

# Step 1: Identify the total debt for each company
companies = ["Honeywell", "Caterpillar", "Pfizer", "PepsiCo", "Boeing"]
# Assuming these are the correct stock symbols
stock_symbols = ["HON", "CAT", "PFE", "PEP", "BA"]
total_debts = {}

for company, symbol in zip(companies, stock_symbols):
  question_debt = f"What is the total debt of {company} ({symbol}) in 2023 in US dollars?"
  documents = select_documents(stock_symbols=[symbol], form_types=["10-K"], fiscal_years=[2023])
  pages = retrieve_relevant_pages(question_debt, documents)
  total_debt = extract_value(question_debt, pages)
total_debts[company] = total_debt

# Step 2: Determine which company has the lowest total debt
lowest_debt_company = min(total_debts, key=total_debts.get)

# Step 3: Find the total revenues of that company
question_revenue = f"What is the total revenues of {lowest_debt_company} in 2023 in US dollars?"
symbol = stock_symbols[companies.index(lowest_debt_company)]
documents = select_documents(stock_symbols=[symbol], form_types=["10-K"], fiscal_years=[2023])
pages = retrieve_relevant_pages(question_revenue, documents)
total_revenues = extract_value(question_revenue, pages)

# Save the answer to a file
with open("answer.txt", "w") as f:
    f.write(total_revenues)
```

Figure 7: An example of code generated by GPT-4 in the CodeGen+DocS+PageR system. The LLM models successfully decompose the question into two main steps: (1) determine the company with the lowest total debt and (2) extract the Total Revenues of that company. In the first step, the question is further decomposed into atomic questions for the Document Selection task and the Page Retrieval task. This fine-grain process ensures the higher accuracy of the system compared to the coarse-grain systems such as Vanilla RAG.

```
You are a financial expert.
The most current fiscal year is {current_year}
You can answer quantitative finance questions by writing Python code using helpful functions.
There are two functions:
- select_document: return a list of supporting documents.
- retrieve_relevant_pages: return a list of relevant pages that contain information to answer the question from the list of documents
- extract value: return an extracted value from the given document
select_document(
    companies: list = None,
    stock_symbols: list = None,
    form_types: list = None,
    fiscal_years: list = None,
    financial_period_end_date_range_start: str = None,
    financial_period_end_date_range_end: str = None
):
  """
  This function matches documents by a series of conditions.
  If the condition is not empty, they must match all given condition
  companies and stock_symbols are not mutually exclusive. A document is matched if satisfies one of these conditions.
  The documents must belong to one of the companies specified by the companies or stock_symbols
  The financial period end date to filter must be between (financial_period_end_date_range_start, financial_period_end_date_range_end)
  :param companies: a list of a few desired company short names.
  :param stock_symbols: a list of the corresponding companies' stock ticker symbols
  :param form_types: a list of the form types such as "8-K" for the current report, "10-K" for the annual report, "10-Q" for the quarterly report
  :param fiscal_years: a list of the corresponding companies' fiscal years
  :param financial_period_end_date_range_start: the beginning of a range used to filter financial period end date in "yyyy-mm-dd" format
  :param financial_period_end_date_range_end: the beginning of a range used to filter financial period end date in "yyyy-mm-dd" format

  :return: a list of supported documents. Return an empty list [] if no document is matched.
  """

retrieve_relevant_pages(question: str, documents: list):
  """
  :param question: a financial question
  :param documents: a list of documents, each with multiple pages
  :return: (list of str)a short list of pages
  """

extract_value(
  question: str,
  pages: list
):
  """
  :param qa: a question
  :param pages: a list of pages
  :return: an extracted value from the given list of pages.
  If the value is a money amount, the returned value is a float number in US dollars.
  If the value is a count, the returned value is a simple float number.
  If the value is a percentage, the returned value is a float number. E.g., 1% would be returned as 0.01.
  If the question is a yes-no question, it would return "Yes" or "No" only.
  """
Finally, you must write the short answer to a file named "answer.txt". The answer must be short, just a Yes/No, or a number
```

Figure 8: The prompt for CodeGen+DocS+PageR

```
Here are some examples:
Question: Did Coca-Cola pay dividends in 2017?
Python Code:
from functions import select_document
from functions import retrieve_relevant_pages
from functions import extract_value
documents = select_document(stock_symbols=["KO"], form_types=["10-K"], fiscal_years
    =[2017])
question = "How much did Coca-Cola pay dividends in USD in 2017?"
pages = retrieve_relevant_pages(question, documents)
value = extract_value(question, pages)
if isinstance(value, str):
    if value == "yes":
        answer = "Yes"
    elif value == "no":
        answer = "No"
    else:
        dividends = float(value)
else:
    dividends = float(value)
if dividends > 0:
    answer = "Yes"
else:
    answer = "No"
with open("answer.txt", "w") as f:
    f.write(answer)

Question: What is the overall revenue growth of Abbott over the last 2-year period?
Python Code:
from functions import select_document
from functions import retrieve_relevant_pages
from functions import extract_value
current_year = {current_year}
question = f"How much did Coca-Cola pay dividends in {{current_year}} in USD?"
documents = select_document(stock_symbols=["KO"], form_types=["10-K"], fiscal_years
    =[current_year])
pages = retrieve_relevant_pages(question, documents)
value_current = extract_value(question, pages=pages)
base_year = current_year - 2
question = f"How much did Coca-Cola pay dividends in {{base_year}} in USD?"
documents = select_document(stock_symbols=["KO"], form_types=["10-K"], fiscal_years
    =[base_year])
pages = retrieve_relevant_pages(question, documents)
value_base = extract_value(question, pages)
growth_percentage = (value_current - value_base) / value_base * 100.0
with open("answer.txt", "w") as f:
    f.write(str(growth_percentage))

Question: How much did NFLX return to the investors in the last 3 years?
Python Code:
from functions import select_document
from functions import retrieve_relevant_pages
from functions import extract_value
current_year = {current_year}
returned_values = []
for year in range(current_year, current_year - 3, -1):
question = f"How much did Netflix return to the investors in {{year}} in USD?"
    documents = select_document(companies=["Netflix"], stock_symbols=["NFLX"],
    form_types=["10-K"], fiscal_years=[year])
    pages = retrieve_relevant_pages(question, documents)
    return_in_us_dollars = extract_value(question, pages)
    returned_values.append(float(return_in_us_dollars))
total_return = sum(returned_values)
with open("answer.txt", "w") as f:
    f.write(str(total_return))
```

Figure 9: The few-shot demonstrations used for CodeGen+DocS+PageR system.

An Automatically Improving Method for Generating Descriptions of Financial Data Quality Grading with LLMs

Yang Zhao[1], **Yohei Ikawa**[1], **Bishwaranjan Bhattacharjee**[2]
[1]IBM Research – Tokyo, Japan [2]IBM T. J. Watson Research Center, USA
yangzhao@ibm.com yikawa@jp.ibm.com bhatta@us.ibm.com

Abstract

Generating descriptions for financial data quality grades (e.g., poor, fair, excellent) enhances both data quality assessment and the trustworthiness of AI models. Traditionally, grading criteria have been manually compiled by humans, a process that is time-consuming and requires domain-specific expertise. In this work, we propose an automated, automatically improving framework for describing financial data quality grades at arbitrary levels. Specifically, we first train a financial classifier to categorize data into multiple quality grades, with the theoretical capability to support arbitrary grading levels. Then, a collected list of financial hypernyms is used to optimize the description for each financial grade using two search strategies. The quantitative results show that the financial knowledge–aware editor improves description accuracy and the QWK correlation score by over 10 points respectively on a hold-out test set, while the qualitative results indicate better performance in terms of informativeness and trustworthiness. We release the code and data here[1].

1 Introduction

Grading financial data involves assigning a score to a document to indicate its relevance and quality within the financial domain. For example, a financial text may be graded as poor (score 1), fair (score 2), or excellent (score 3) to reflect its quality and domain relevance. Among other factors, generating descriptions for different grades plays an important role in several aspects: first, descriptions help establish clear criteria for each grade, enabling users to place greater trust in AI models; second, they can serve directly as annotation guidelines, helping users design LLM-based annotation prompts to filter high-quality data from large corpora such as FineWeb (Penedo et al., 2024), which has become increasingly popular in recent years.

[1]https://github.com/code4nlp1713/code

Previously, many studies have relied on manually developed, domain-specific data grading criteria. For example, FineWebEdu[2] enlisted human annotators to create five data quality grading criteria, which were then integrated into annotation prompts to guide LLMs in extracting education-specific data. Despite such successes, prompt design still depends heavily on domain-specific human expertise, and in less familiar domains, generating accurate grade descriptions becomes even more challenging.

To address this shortcoming, we propose a automatically improving method for generating descriptions in financial data quality grading. Table 1 shows a initial 3-grade description and optimized 3-grade description. We herein focus on two research questions: (1) **How can we obtain quality grading for financial data at arbitrary levels?** (2) **How can we automatically generate informative descriptions for each grade?**

To answer these questions, in Section 2, we introduce a two-stage approach to obtaining binary annotations (financial/non-financial) for the financial data, and train a financial document classifier to generate probabilities, which are segmented into grading scores. In Section 3, we automatically optimize grade descriptions using a curated set of financial hypernyms and a financial knowledge–aware (Fin-aware) editor. This editor guides LLMs to produce and iteratively refine descriptions using the model's own feedback via PPO. We further explore two search strategies that improve both description accuracy and Quadratic Weighted Kappa (QWK) correlation by more than 10 points over baseline methods. Qualitative evaluation also confirms that the resulting descriptions are more informative and trustworthy.

[2]https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

Score	3-Grade description	Automatically Optimized 3-Grade description
1	Score 1 if the document is poor.	Score 1 if the document is poor at explaining financial concepts and institutions, or does not discuss the performance or valuation of assets or liabilities at all.
2	Score 2 if the document is fair.	Score 2 if the document presents general financial information in a utilitarian manner but uses vague terms and lacks specific details about the financial status of the entity or individual.
3	Score 3 if the document is excellent.	Score 3 if the document demonstrates clarity and depth in discussing financial topics, including performance metrics, risk management, financial strategies, and potential uncertainties, while providing transparent and accurate data.

Table 1: Examples of 3-grade and optimized 3-grade financial data quality descriptions. In the 3-grade scale, 3 denotes the highest quality and 1 the lowest.

2 Related Work

Several studies on prompt optimization are relevant to our work. Prompt optimization methods include paraphrasing (Jiang et al., 2020; Yuan et al., 2021; Haviv et al., 2021) and reinforcement learning (RL) approaches (Deng et al., 2022; Zhang et al., 2022), though prior RL methods often yield uninterpretable prompts or have limited action spaces. Kong et al. (2024) automate prompt rewriting via RL but target simpler, single-sentence tasks, while our method addresses longer, multi-criteria prompts (∼300 words). We further incorporate financial knowledge to improve description quality, distinguishing our approach from earlier work.

3 Financial Data Grading Annotation

We use FineWeb dataset (Penedo et al., 2024) and randomly select 600k documents for annotation. Because financial documents[3] are scarce in FineWeb, directly annotating such a large set is inefficient; thus, we adopt a two-stage approach to annotate ground-truth grading.

Stage 1: We first prompt LLMs to generate a list of around 200 financial keywords (see Appendix A for details on keyword generation) and sort the 600k documents in descending order based on their overlap with financial keywords in each document's bag of words. Annotation then begins from the head and tail of the sorted list for financial and non-financial classification, respectively.

Stage 2: We then employ Human-LLM collaborative annotation for binary classification, as it is much easier and more reliable than multi-scale annotation for both LLMs and humans. We use Mixtral-8x7B-Instruct[4] model to annotate documents and then ask a financial expert[5] to review the LLM's annotation to correct them using the same binary annotation instruction in Appendix B. After removing the identified error cases (4% error rate in the LLM's annotations), we obtain a ground-truth financial dataset consisting of 3,840 positive (high-quality financial) and 3,840 negative documents. Please see Appendix E for annotation details and data statistics.

Subsequently, we shuffle the financial and non-financial document sets separately, taking the first 1k financial and the first 1k non-financial documents respectively as the test set, with the remaining documents used to train a RoBERTa-based (Liu, 2019) financial classifier. The financial classification accuracy on the test set is 98.8%.

Dataset	Grade Levels	Annotated Documents
3-grade	Poor: 0.0 (<0.001) Fair: 0.5 (±0.015) Excellent: 1.0 (>0.999)	900 total (300 per level) 450 val., 450 testing
4-grade	Poor: 0.0 (<0.001) Fair: 0.33 (±0.015) Good: 0.66 (±0.015) Excellent: 1.0 (>0.999)	1.2k total (300 per level) 600 val., 600 testing

Table 2: Dataset description for 3-grade and 4-grade financial document classification.

Probability Segmentation as Quality Grading We take 580k unannotated documents from the FineWeb dataset and apply our financial classifier to assign a probability score[6] to each document. As expected, most probabilities are close to either 0 or 1, while thousands fall in the middle range (see Table 7 in Appendix F. In this study, we empirically define 'middle' probability thresholds for different quality levels, as Table 2 shows.

[3] A financial document is herein defined as any finance-related text within large corpora such as FineWeb.

[4] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1. It is licensed under Apache License 2.0.

[5] The annotator holds a Ph.D. degree and works in the financial industry.

[6] We extract the probability of label 1 from the softmax

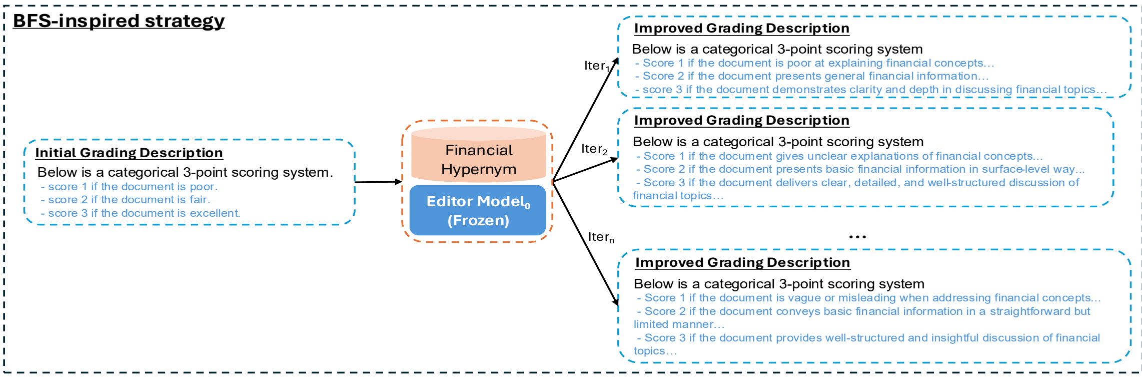


Figure 1: BFS-inspired strategy for automatically improving descriptions in data grading.

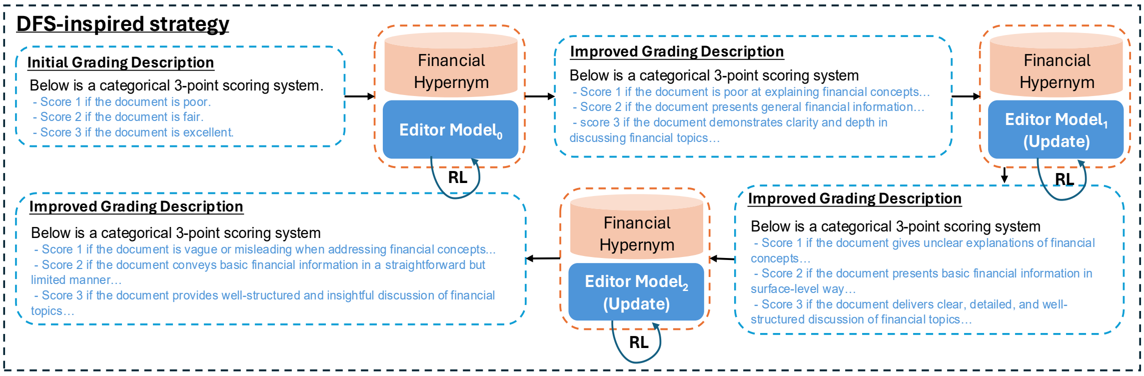


Figure 2: DFS-inspired strategy for automatically improving descriptions in data grading.

4 Proposed Method

To generate description for each grade, we frame the problem in the context of data annotation using LLMs: typically, an annotation prompt containing data grading criteria is manually crafted and provided to LLMs to generate classification results. For example, in the education domain, HuggingFace researchers manually designed a scoring prompt[7]. However, we reverse the problem herein: *given classification results (low/moderate/good/best) of documents, can we generate description for each grade without human effort?* We explore two search strategies for generating descriptions, inspired by Breadth-First Search (BFS) (Moore, 1959) and Depth-First Search (DFS) (Tarjan, 1972).

Formally, we define the description for data grade as X and the data annotation prompt containing this description as $P(X)$. The description X is iteratively refined by an editor model, LLM_{edt} into X'. We provide the prompt $P(X)$ to an evaluation language model, LLM_{eval}, which generates the predicted grade Y_{pred}. The ground-truth grade is denoted as Y_{true}. We define the difference between Y_{pred} and Y_{true} using the Quadratic Weighted Kappa (QWK) Correlation (Cohen, 1968) whose domain ranges from -1 and 1. Our goal is to find the optimal X' that maximizes QWK score.

4.1 BFS-Inspired Strategy

Based on the problem formulation, the BFS-inspired strategy aims to generate as many description as possible iteratively from the initial description X_0 (see the Appendix C). As shown in Figure 1, every time, editor model LLM_{edt} will only edit description according to editing prompt (see the Appendix D). We generate N new descriptions and select the one with the highest QWK score as the output of the BFS-inspired strategy.

4.2 DFS-Inspired Strategy

The DFS-inspired strategy, different from BFS-inspired one, will update the parameter of LLM_{edt} using converted Quadratic Weighted Kappa (QWK) score as reward via PPO (Schulman et al., 2017) reinforcement learning framework, as shown in Figure 2. Also, each time, LLM_{edt} builds upon

layer of the financial classifier.

[7]https://huggingface.co/HuggingFaceFW/fineweb-edu-classifier/blob/main/utils/prompt.txt

	3-grade hold-out test set			4-grade hold-out test set		
	Accuracy	F1	QWK Corr.	Accuracy	F1	QWK Corr.
Initial description	38.9	31.4	19.0	27.8	22.2	22.0
BFS	72.2	71.0	79.7	48.0	45.4	67.1
DFS	56.4	57.5	58.6	30.3	24.2	39.3
Our Fin-aware BFS	**83.1**	**82.2**	**88.2**	59.2	58.1	78.0
Our Fin-aware DFS	79.1	78.8	85.1	**61.0**	**60.7**	**78.6**

Table 3: Performance on hold-out test sets for 3-grade and 4-grade evaluations is measured using accuracy, macro-F1, and Quadratic Weighted Kappa (QWK). We select the description with the highest QWK score after 50 iterations on a 600-document validation set and evaluate it on a separate 600-document hold-out set.

the current best description to generate a new one, continuously evolving itself over N iterations.

4.3 Integration of Financial Hypernym

Given the vast search space of LLM_{edt}, finding optimal descriptions can be time-consuming. To address this, we incorporate financial hypernyms extracted from the FineWeb corpus without relying on external resources or human annotation. These hypernyms serve as high-level descriptors, enabling the model to rewrite descriptions without delving into overly specific financial terms or events. For example, *Citigroup* is replaced with *bank*, and *property* with *asset*. To obtain financial hypernyms, we first use our financial classifier to select 5.6k documents with a probability above 0.99 from the 580k dataset. We then extract the most frequent financial nouns and adjectives, removing numbers and common Wikipedia words. Following (Peng et al., 2022), we prompt the RoBERTa model (Liu, 2019) with crafted templates such as *In a financial context, word is a type of <mask>.* or *In a financial context, something word is <mask>.*, selecting the most probable <mask> token as a high-level substitute for *word*. This process yields 120 financial hypernyms.

5 Experiments and Result

5.1 Models and Experimental Setup

For LLM_{eval} and LLM_{edt}, we use Mixtral-8x7B-Instruct-v0.1[8] with 4-bit quantization. We refer readers to Appendix G for training details and experimental setup.

5.2 Result

Table 3 presents results on a test set that was not used during the training process. Accuracy and Macro-F1 measure the exact grade match with ground-truth data grades, while QWK quantifies the correlation between predicted and ground-truth grades, considering the ordinal nature of the grading score. We have the following observations:

(1) Both BFS-inspired and DFS-inspired optimizations yield better results than the initial grade description. Incorporating financial knowledge (Fin-aware BFS/DFS) further boosts Accuracy, F1, and QWK scores by over 10 points each, confirming the effectiveness of this simple financial hypernym integration.

(2) DFS underperforms BFS on both the 3-grade and 4-grade test sets. However, when augmented with financial hypernyms, Fin-aware DFS nearly matches Fin-aware BFS on the 3-grade test and slightly surpasses it on the more challenging 4-grade test, suggesting the potential of combining DFS-inspired search with RL to update the editor model (In BFS, editor model is not updated).

To further confirm whether and to what extent the generated descriptions contain financial hypernyms, we compute the word overlap between grade descriptions produced by BFS/DFS and Fin-aware BFS/DFS, respectively. Table 4 shows the percentage of financial hypernyms in each description. It is interesting to note that the methods with the highest percentage of financial hypernyms (8.2% for Fin-aware BFS and 12.3% for Fin-aware DFS) achieved the best performance in Table 3, implying that the proportion of financial hypernyms may impact performance, although a model with 0% financial hypernyms also led to high accuracy (79.1%).

Qualitative Evaluation To further evaluate description quality, a human expert rated the outputs in Table 8 and Table 9 in Appendix on three criteria: Fluency, Informativeness (i.e., detailed, spe-

[8] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1

Generation	3-grade	4-grade
description_BFS	5%	0.7%
description_DFS	2.4%	2.9%
description_BFS_fin	8.2%	2.2%
description_DFS_fin	0%	12.3%

Table 4: Percentage of financial hypernyms in each description. A higher percentage indicates a more informative description in the financial domain.

cific, and actionable), and Trustworthiness (i.e., logically consistent without contradicting basic financial principles). As shown in Table 5, fluency scores are similar across all five descriptions, while descriptions enhanced with financial hypernyms perform better in both informativeness and trustworthiness.

Generation	**Fluen.**	**Infor.**	**Trust.**
description_init	4	2	3
description_BFS	4	3	4
description_DFS	4	4	4
description_BFS_fin	4	4	5
description_DFS_fin	4	5	5

Table 5: Human evaluation for Fluency (Fluen.), Informativeness (Infor.), and trustworthiness (Trust.) of different grading descriptions on a 1–5 Likert scale.

6 Conclusion

We propose an automated, automatically improving method for financial data quality grading that supports arbitrary grading levels via a simple binary domain-data classifier. Financial hypernyms are automatically derived and integrated with two search strategies, yielding significant performance gains and more informative descriptions, as confirmed by qualitative. In the future, we plan to extend our approach to other domains, as it requires minimal binary annotation or potentially no human annotation, given the low error rate of LLMs' annotations.

References

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. 2022. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3369–3391, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adi Haviv, Jonathan Berant, and Amir Globerson. 2021. Bertese: Learning to speak to bert. *arXiv preprint arXiv:2103.05327*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Weize Kong, Spurthi Hombaiah, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. PRewrite: Prompt rewriting with reinforcement learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 594–601, Bangkok, Thailand. Association for Computational Linguistics.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Edward F Moore. 1959. The shortest path through a maze. In *Proc. of the International Symposium on the Theory of Switching*, pages 285–292. Harvard University Press.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Bo Peng, Emmanuele Chersoni, Yu-Yin Hsu, and Chu-Ren Huang. 2022. Discovering financial hypernyms by prompting masked language models. In *Proceedings of the 4th Financial Narrative Processing Workshop @LREC2022*, pages 10–16, Marseille, France. European Language Resources Association.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Robert Tarjan. 1972. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. 2022. Tempera: Test-time prompting via reinforcement learning. *arXiv preprint arXiv:2211.11890*.

A Financial Keywords Collections

We prompt the Mixtral-8x7B-Instruct model to generate a list of words representative of the financial domain. To encourage diversity in the generated financial keywords, we query the model eight times, each with different decoding hyperparameters. Specifically, we select the *temperature* from {0.6, 0.7, 0.8}, *top_p* from {0.9, 0.95, 1.0}, and *top_k* from {50, 75, 100}. Each time, we use a different combination of these three hyperparameters to generate 500 financial keywords.

Please generate a list of 500 single finance-related words in valid JSON format. Each entry should include the following fields:
1. "word": A single finance-related word.
2. "justification": A single sentence explaining why the word is relevant to the financial domain.
Ensure the JSON is syntactically valid and formatted as follows:

```
[
  {
    "word": "example_word",
    "justification": "This is an
    example description of the word's
    relevance to the financial domain."
  },
  ...
]
```

Provide the output in a single JSON array containing exactly 500 entries. Note that financial word should be single word instead of phrases.

Our experimental results show that (1) a significant number of generated words are duplicates, and (2) many outputs are financial phrases (rather than single words), including financial institution names. After deduplication, we obtain a final list of 202 unique financial keywords, such as *EPS, slippage, or cashflow*.

Sort Documents with Financial Keywords
For each document in the 600k FineWeb dataset, we first use NLTK[9] for word tokenization, convert all words to lowercase, and convert them into a bag of unique words, BOW_i for document i. We then compute the proportion of overlap between each document's bag of unique words and the financial word list, $\mathbf{F}$, using the following formula:

$$\text{value}_i = \frac{|\mathbf{F} \cap BOW_i|}{|BOW_i|}$$

[9]https://github.com/nltk/nltk

B Financial Data Annotation Instruction

Below is the instruction used by both LLMs and humans for binary financial data annotation. We found that using the text 'The document is financial text.' or 'The document is not financial text.' is more effective than outputting a label of 1 or 0 in the prompt output format. We later convert these textual outputs into labels 1 and 0.

You are an expert in financial data quality with deep expertise in analyzing financial documents. Your task is to evaluate the given document to determine its relevance and quality as financial text.

Document: DOCUMENTS GO HERE

Carefully assess the document and output one of the following responses:
- 'The document is financial text.'
- 'The document is not financial text.'
Provide only the response, without additional description.

C Prompt with Initial description for Financial Data Grade

We start the experiment using a document annotation prompt, similar to this one[10], with the following initial description. We use {poor/fair/excellent} for 3-grade initial description.

Below is an document from a web page. Evaluate it using the categorical {n}-point scoring system described below:
- score 1 if the document is poor.
- score 2 if the document is fair.
- score 3 if the document is good.
- score 4 if the document is excellent.
The document:{DOCUMENT}
After examining the document:
- Briefly justify your total score, up to 100 words.
- You must prepend the score exactly using the following format:
'financial score: <total points>.'

[10]https://huggingface.co/HuggingFaceFW/fineweb-edu-classifier/blob/main/utils/prompt.txt

D Editing Prompt

LLMs use an editing prompt to generate new descriptions for financial data grading criteria. In the BFS-inspired method, descriptions are generated based on the same initial description from Appendix C. In contrast, the DFS-inspired method iteratively builds upon the current best description to produce a new description. To incorporate financial hypernyms, we ask LLMs to selectively use financial word in the word list by appending *based on financial topic words from the following list, using them selectively.* after *Rewrite... requirements.* Also, we add *financial topic words: {CONCATENATED_FINANCIAL_WORDS}* right before *{n}-points*.

Below is a categorical {n}-point scoring system designed to evaluate the financial value of a document. Rewrite the following {n} points via rephrasing and/or adding specific requirements. Use illustrative description if needed.

{n}-points: {REVISED_POINTS}

Each point should begin with '- score X if the document...'
Output the new {n} points only.

E Human-LLM Collaborative Annotation

First, we use the Mixtral-8x7B-Instruct model[11] to assign a label of 1 to high-quality financial documents and 0 otherwise, using the binary annotation instruction in Appendix B. Among 6k documents from the head of the sorted list, 4k are labeled as 1, whereas for 6k documents from the tail, 5.9k are annotated as 0. Next, a human annotator manually reviews the 4k documents labeled as 1 and identifies error cases in 4% (about 160 documents) of them, while finding very few errors among the documents labeled as 0[12]. Finally, we remove the identified error cases, resulting in a ground-truth positive dataset of 3,840 documents and negative dataset of 3,840 documents. Table 6 shows that basic statistics of annotated financial data.

[11]https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1. It is licensed under Apache License 2.0.

[12]We do not ask the human annotator to review all 5.9k documents but only the first 4k, as our goal is to create a balanced training dataset for both positive and negative class.

	Training set	**Test set**
# of document	5,680	2,000
Average words	458.7	466.3
STD	211.8	210.2

Table 6: Statistics of annotated financial dataset. We use NLTK toolkit for word tokenization.

F Financial classifier probability distribution on FineWeb documents

Prob. Range	**# of documents**
(0.0, 0.05)	542,628
(0.05, 0.1)	3,772
(0.1, 0.15)	2,117
(0.15, 0.2)	1,536
(0.2, 0.25)	1,109
(0.25, 0.3)	899
(0.3, 0.35)	752
(0.35, 0.4)	718
(0.4, 0.45)	674
(0.45, 0.5)	656
(0.5, 0.55)	638
(0.55, 0.6)	649
(0.6, 0.65)	715
(0.65, 0.7)	745
(0.7, 0.75)	802
(0.75, 0.8)	897
(0.8, 0.85)	1,016
(0.85, 0.9)	1,326
(0.9, 0.95)	2,163
(0.95, 1.0)	16,188
In total	580,000

Table 7: Probability Distribution over FineWeb documents.

G Experimental Setup Details

The parameters of LLM_{eval} remain frozen in all BFS- and DFS-inspired methods. In BFS-inspired methods, LLM_{edt} is also frozen, whereas in DFS-inspired methods it is updated via LoRA (Hu et al., 2021) (rank 8, alpha 32) within the PPO-based RL framework. We use a learning rate of 2.82×10^{-6} and sampling-based decoding (top_p = 1.0, top_k = 0) to encourage creative writing. For both BFS and DFS, N is set to 50. All experiments are conducted on four A100 GPUs.

Baseline 3-grade description produced by vanilla BFS with **5%** financial hypernyms	**Our best 3-grade description** by Fin-aware BFS with **8.2%** financial hypernyms
• Score 1 if the document is poor and lacks proper financial **analysis**. • Score 2 if the document is adequate, yet misses critical **information** on the financial specifics of the **investment**. • Score 3 if the document is excellent and exhibits deep comprehension of financial intricacies while being presented in a clear, easy-to-understand manner.	• Score 1 if the document is poor at explaining financial concepts and **institutions**, or does not discuss the **performance** or **valuation** of assets or liabilities at all. • Score 2 if the document presents general financial **information** in a utilitarian **manner** but uses vague terms and lacks specific details about the financial status of the **entity** or **individual**. • Score 3 if the document demonstrates clarity and depth in discussing financial topics, including **performance** metrics, **risk management**, financial strategies, and potential uncertainties, while providing transparent and accurate data.

Table 8: Case study: comparison of baseline 3-grade description (vanilla BFS) and best 3-grade description (Fin-aware BFS).

<table>
<tr><th>Baseline 4-grade description produced by vanilla DFS with 2.9% financial hypernyms</th><th>Our best 4-grade description by Fin-aware DFS with 12.3% financial hypernyms</th></tr>
<tr><td>
Score 1 if the document lacks crucial details, including the author's identity or copyright information.
Score 2 if the document offers basic primary data but misses essential contact details, such as an email address or phone number.
Score 3 if the document encompasses detailed key and supportive information, complemented by clear screenshots, relevant links, and informative appendices.
Score 4 if the document offers extensive advantages, comprising functional code samples, valuable learning sources, and a thorough project roadmap, all while exhibiting exceptional writing and organization.
</td><td>
4-point scoring system for financial documents:
Score 1 if the document is of poor quality and lacks essential financial topic words like finance, investment, transaction, income, exchange, and property. This may indicate an insufficient understanding of the legal context or the boundaries of a financial document.
Score 2 if the document is fair and contains adequate financial topic words. However, the depth of financial analysis or detail on topics like stock, payment, company, liability, performance, or credit management may be lacking, making it difficult to understand the impact on the target audience.
Score 3 if the document is good and has all necessary financial topic words, including variables such as volatility, asset, debt, management, regulation, and risk. The document may also address performance indicators, specialized financial instruments, and concepts surrounding wealth creation, technology, and governance, with clear communication.
Score 4 if the document is excellent and is abundant with financial topic words and concepts, addressing interrelated financial factors and broader economic context. The document may consider risks, constraints, negative events, and apply concepts such as insurance, punishment, and bankruptcy to protect against uncertainties. Finally, it is clear, concise, referencing proper accounting protocols, and incorporates proper communication protocols for the intended audience.
</td></tr>
</table>

Table 9: Case Study: comparison of baseline 4-grade description (vanilla DFS) and best 4-grade description (Fin-aware DFS).

Earnings2Insights: Analyst Report Generation for Investment Guidance

Takehiro Takayanagi[1,2,3], Tomas Goldsack[4], Kiyoshi Izumi[1,2], Chenghua Lin[5], Hiroya Takamura[3], Chung-Chi Chen[3]

[1]Simulacra Inc., [2]The University of Tokyo,
[3]National Institute of Advanced Industrial Science and Technology,
[4]The University of Sheffield, [5]University of Manchester

{takayanagi,izumi}@simulacra.co.jp,tgoldsack1@sheffield.ac.uk
chenghua.lin@manchester.ac.uk,takamura.hiroya@aist.go.jp, c.c.chen@acm.org

Abstract

We present **Earnings2Insights**, a shared task on generating actionable investment reports from earnings conference call (ECC) transcripts. Unlike traditional financial summarization or QA, the goal is decision support: systems must synthesize facts, highlight risks and opportunities, and support investors in making sound actions. The task required participants to produce reports based on ECC transcripts. In total, 45 teams registered, with 12 teams submitting reports and 9 submitting solution papers, spanning diverse agentic designs, retrieval-augmented methods, and data expansion strategies. Our evaluation consists of *human evaluation* and *automatic evaluation*. Results reveal a consistent divergence between systems that scored highly in automatic evaluations and those that most effectively supported human investment decisions, underscoring the limits of style- or reference-based comparisons in high-stakes financial report generation. We advocate human-centered, decision-oriented assessment as the primary lens, with automated signals serving as complementary diagnostics. We release task design, evaluation data, and scripts to catalyze research on decision-centric financial text generation.[1]

1 Introduction

With the advent of large language models (LLMs), researchers have increasingly explored their application in specialized professional domains. Beyond automatic text comprehension, LLMs now demonstrate promising abilities in analytical report generation, enabling new forms of decision support in high-stakes fields such as law, medicine, and finance (Goldsack et al., 2025). Financial decision-making is a particularly high-stakes domain, where inaccurate or misleading reports can directly impact markets and investor outcomes (Lai et al., 2023). Traditional NLP tasks in finance, such as information extraction (Chen et al., 2021a), question answering (Chen et al., 2021b; Liu et al., 2023), and summarization (Huang et al., 2024), have focused on factual accuracy. Recently, more and more focus has shifted to the **human side**, such as building financial advisor systems with LLMs (Takayanagi et al., 2025a,b). At the same time, producing actionable investment insights requires more than summarizing facts: systems must synthesize information, highlight risks and opportunities, and persuade investors to act (Huang et al., 2025).

The **Earnings2Insights shared task** is designed to evaluate the capability of LLMs to generate convincing investment reports from earnings call transcripts. Participants may approach the task in two ways: using only the raw transcript, or enriching the input with timestamp-aligned retrieval of relevant external information. A central challenge in financial report generation is evaluation. Prior studies have shown that comparing generated outputs with ground-truth answers via automatic metrics may be insufficient, and that current LLMs remain unreliable as evaluators (Chen et al., 2024; Goldsack et al., 2025). Inspired by decision-based evaluation frameworks (Takayanagi et al., 2025c; Huang et al., 2025), we instead assess systems by their ability to guide human investment decisions. Annotators are asked to make buy/hold/sell judgments based on the generated reports, and the correctness of these decisions serves as the primary evaluation metric.

This paper provides an overview of the Earnings2Insights shared task and dataset, summarizes the methods employed by participating teams, and evaluates their experiments. Through this, we aim to shed light on the current capabilities and limitations of LLMs in financial report generation, and to foster broader discussion on human-centered evaluation for decision-critical AI.

[1]https://github.com/TTsamurai/Earnings2Insights.git

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

2 Tasks and Dataset

The Earnings2Insights shared task evaluates the ability of large language models to generate actionable investment reports from earnings call transcripts. Unlike traditional summarization or QA tasks, the objective is not merely to condense information but to produce guidance that highlights risks, opportunities, and potential actions for investors. This setting mirrors real-world analyst workflows, where the value of a report lies in its ability to influence financial decisions rather than reproduce factual details alone.

We use earnings conference calls (ECCs) as our primary scenario. ECCs are quarterly events in which company executives present financial results and discuss their outlook with investors and analysts. ECCs play a central role in shaping market sentiment because they combine both quantitative disclosures (e.g., revenues, forecasts, margins) and qualitative signals (e.g., managerial tone, confidence, and forward-looking statements). Importantly, professional equity analysts routinely write analyst reports immediately after ECCs, making this setting particularly suitable for our task: it naturally links raw financial discourse to the generation of actionable investment insights.

In this shared task, we provide two complementary subsets of earnings call transcripts:

- **ECTSum Subset (40 transcripts)**
  This subset corresponds to the ECTSum dataset (Mukherjee et al., 2022). Each transcript is paired with a "ref" file representing its associated summary. Participants may choose whether or not to use these summaries as auxiliary supervision.

- **Professional Subset (24 transcripts)**
  This subset consists of transcripts that are aligned with professional analyst reports. Unlike the ECTSum subset, no reference summaries are provided to participants. Instead, the organizers will later compare system outputs against the analyst reports to assess alignment with professional standards.

In total, participants are required to generate reports for all 64 earnings calls across both subsets.

A total of 45 teams registered for the Earnings2Insights shared task, of which 12 teams submitted reports and 9 teams submitted solution papers.

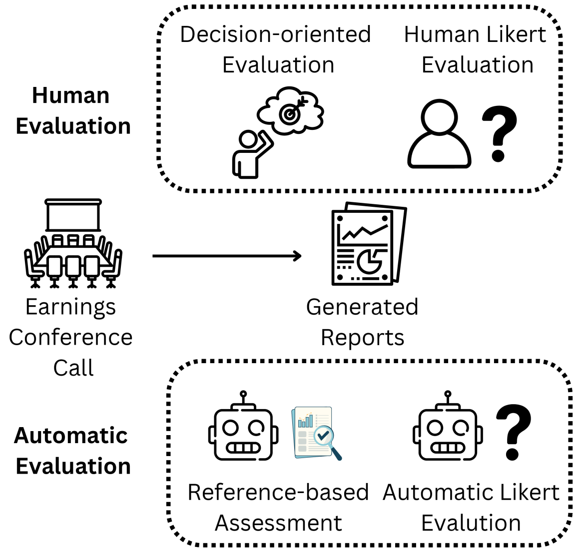


Figure 1: Evaluation framework consisting of human evaluation and automatic evaluation.

3 Evaluation

For evaluation, we conducted both human evaluation and automatic evaluation in order to capture complementary perspectives on system performance. Our evaluation framework is illustrated in Figure 1.

3.1 Human Evaluation

Human evaluation was designed to test whether the generated reports could effectively guide investment decisions. After reading each report, annotators were asked to make one of three decisions: *Buy* (expect the stock to go up), *Neutral* (uncertain), or *Sell* (expect the stock to go down). Ground-truth labels were derived from realized stock returns at three horizons: one business day (1bd), one week (5bd), and one month (20bd). These labels were coded as $+1$ for upward movements and -1 for downward movements. Neutral responses were excluded from the calculation, since they indicate uncertainty rather than a directional prediction. Accuracy was computed at each horizon as the proportion of correct predictions among all non-neutral responses, and an overall accuracy score was obtained by averaging across the three horizons.

In addition to directional accuracy, we also evaluated the perceived quality of the generated reports. Annotators rated each report on five criteria—clarity, logic, persuasiveness, readability, and usefulness—using a 7-point Likert scale. We report both the average score for each dimension and the overall mean across all five dimensions.

3.2 Human Evaluation Setup

For human evaluation, we used the Prolific platform.[2] We recruited 192 English native speakers residing in either the United Kingdom or the United States, each with a past task acceptance rate above 80%. Each crowdworker participated in one hour of evaluation, during which they made financial decisions based on a total of 12 generated reports. Consequently, every one of the 64 reports submitted by the 12 participating teams received independent judgments from three annotators. Participants were compensated at a rate of £8 per hour. In total, the study required 210 participants, amounting to a total cost of £1,680.

3.3 Automatic Evaluation

To complement the human evaluation, we also introduced automatic evaluation measures based on large language models. In particular, we adopted an "LLM-as-a-judge" framework (Gu et al., 2024), to provide pairwise and absolute quality judgments.[3] First, we measure the win rate against professional analyst reports. In this pairwise comparison, each system-generated report is compared directly with an analyst-written report, and the win rate reflects the proportion of cases in which the system report was judged superior, excluding ties. Second, we compute the average Likert score by aggregating the 1–7 ratings across the five qualitative dimensions described above. This provides a single summary indicator of report quality.

4 Methods

Overall, the participating teams adopted diverse agentic approaches, with many incorporating retrieval-augmented generation (RAG) and various data expansion strategies. This diversity illustrates the richness of methods explored for financial report generation.

SigJBS used a three-agent pipeline (extraction, reasoning, critique) to parse transcripts into key financial milestones, generate recommendations with risk analysis, and iteratively refine outputs for consistency and factuality (Sinha et al., 2025).

Jetsons combined writer agents with feedback agents in a ReAct-style loop (Yao et al., 2023), integrating structured financial data via Alpha Vantage to produce reports that balanced factual accuracy, risk coverage, and persuasiveness (Dakle et al., 2025).

Team	Average	Day	Week	Month
DKE	**0.581**	0.596	0.577	**0.570**
DataLovers	0.579	0.597	**0.611**	0.529
Jetsons	0.571	0.607	0.555	0.552
SigJBS	0.545	**0.609**	0.513	0.512
iiserb	0.537	0.576	0.558	0.477
PassionAI	0.537	0.588	0.557	0.466
Finturbo	0.524	0.504	0.568	0.500
Bgreens	0.522	0.469	0.581	0.516
LangKG	0.518	0.589	0.542	0.424
SI4Fin	0.515	0.525	0.524	0.497
KrazyNLP	0.471	0.514	0.525	0.375
bds-LAB	0.462	0.478	0.434	0.474

Table 1: Average accuracy of financial decisions across time horizons.

LangKG employed a cognitive reasoner framework, generating personalized reports tailored to investor profiles using a six-dimensional analysis and conviction scores for transparency (Prasanna and Su, 2025).

DataLovers orchestrated multiple analyst agents (finance, sentiment, strategy) whose outputs were merged into a structured report template. Their meta-prompting framework emphasized collaborative reasoning, implemented with a compact LLaMA model (Chatwal et al., 2025).[4]

iiserb modeled investment committee debates through a Structured Adversarial Synthesis framework, staging adversarial dialogues among bullish, bearish, and devil's advocate agents to refine logic and persuasiveness (Sadhu et al., 2025).

Bgreens mimicked the analyst–writer–editor workflow with multi-agent roles implemented via AutoGen (Wu et al., 2024). Iterative feedback improved consistency and readability, with experiments showing higher decision accuracy compared to single-agent baselines (Satapathy et al., 2025).

DKE built a retrieval-augmented debate system with five domain-specific analyst agents and a collaborative debate phase among trust, skeptic, and leader agents, synthesizing robust recommendations with confidence scores (Cai et al., 2025).

FinTurbo emphasized professional-style reports with structured data and visualization, combining charting, highlighting, writing, and editing

[2] `https://www.prolific.com/`

[3] We use `gpt4.1` as our evaluator.

[4] `https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct`

Team	Average	Clarity	Logic	Persuasiveness	Readability	Usefulness
LangKG	**5.96**	**6.02**	**5.92**	5.90	**5.81**	**6.13**
Jetsons	5.90	6.00	5.89	5.81	**5.81**	6.01
DKE	5.74	5.71	5.89	**5.95**	5.17	5.98
SigJBS	5.67	5.76	5.68	5.59	5.61	5.72
SI4Fin	5.56	5.52	5.84	5.60	5.06	5.80
DataLovers	5.50	5.56	5.45	5.32	5.73	5.47
Bgreens	5.49	5.51	5.61	5.51	5.09	5.74
KrazyNLP	5.29	5.15	5.49	5.21	5.01	5.59
iiserb	5.19	5.01	5.51	5.14	4.72	5.57
Finturbo	5.11	5.02	5.39	4.90	4.86	5.40
bds-LAB	4.99	4.91	5.21	5.03	4.55	5.27
PassionAI	4.70	4.64	4.74	4.39	4.88	4.86

Table 2: Average Likert scores across five qualitative dimensions.

Team	ALS	WR
SI4Fin	**4.916**	0.956
LangKG	4.903	0.881
Jetsons	4.834	0.762
KrazyNLP	4.830	**0.962**
iiserb	4.807	0.930
DKE	4.803	0.783
Finturbo	4.625	0.169
SigJBS	4.597	0.526
Bgreens	4.575	0.615
bds-LAB	4.510	0.711
PassionAI	4.143	0.365
DataLovers	4.134	0.345

Table 3: Automatic evaluation results. ALS = Average Likert Score (1–7); WR = Win Rate vs. Analyst Report.

agents. They expanded the dataset by crawling additional transcripts to enable temporal RAG comparisons (Yang et al., 2025).

SI4Fin integrated external financial statements from Alpha Vantage with an AutoGen-based agentic framework (Wu et al., 2024), where analyst agents extracted trends (YoY, QoQ) and writers incorporated them into grounded reports (Tan et al., 2025).

5 Results

5.1 Human Evaluation Results

Table 1 reports the average accuracy of financial decisions made by annotators after reading the reports generated by each team. Accuracy is computed for one business day (Day), one week (Week), and one month (Month) horizons, with the overall average representing the mean of the three horizons.

Table 2 presents the average Likert scores for clarity, logic, persuasiveness, readability, and usefulness, as rated on a 7-point scale. We also report the overall mean score across the five criteria.

Overall, the results show noticeable variation across teams, with certain systems excelling in decision accuracy while others were rated more highly on subjective quality dimensions. This highlights the complementary nature of accuracy-based and human-centered evaluations in financial text generation.

5.2 Automatic Evaluation Results

In addition to human evaluation, we conducted automatic evaluation using an LLM-as-a-judge framework. Table 3 reports two measures: **ALS** (Average Likert Score), the average 1–7 rating across five dimensions (persuasiveness, logic, usefulness, readability, and clarity), and **WR** (Win Rate vs. Analyst Report), the proportion of pairwise comparisons in which a system-generated report was judged superior to a professional analyst report (ties excluded).

6 Discussion

The results reveal a key divergence between decision-oriented human evaluation and automatic evaluation based on win rates against professional analyst reports. Teams such as DKE and DataLovers scored highly in human evaluation—effectively supporting annotators' investment decisions—yet ranked lower in automatic evaluation. In particular, DataLovers' reports provided practical guidance but showed a notably low win rate. This suggests that automatic metrics fail to capture the true decision utility of generated texts. Prior studies indicate that amateur investors are often unpersuaded by professional analyst reports, whose language and logic can be inaccessible. Thus, benchmarking generated texts solely against professional reports is insufficient for assessing their usefulness in real decision-making (Takayanagi et al., 2025c).

Moreover, the divergence between human and automatic Likert-scale evaluations highlights risks in relying on LLMs as evaluators. While LLMs offer scalability and consistency, their judgments may not align with actual investor behavior. This reinforces the central motivation of the shared task: evaluation must remain grounded in human decision outcomes, with automatic methods serving as complements. Future work should therefore pursue hybrid evaluation schemes that integrate human judgment, domain-specific financial metrics, and scalable LLM-based assessments.

7 Conclusion

This paper presented the Earnings2Insights shared task, which evaluates the capability of large language models to generate actionable investment guidance from earnings call transcripts. Distinct from traditional summarization or QA, our setting targets human-centered decision support: systems must synthesize facts, surface risks and opportunities, and support investors toward sound actions. We released two complementary subsets (ECTSum and Professional), and attracted a diverse set of agentic methods from participating teams.

Our evaluation combined decision-oriented human assessment with an automatic "LLM-as-a-judge" protocol. Results revealed a consistent divergence: several systems that improved human decision accuracy did not necessarily score highly against professional analyst reports or in LLM-based judgments, and vice versa. These findings underscore a central lesson for high-stakes financial NLP: evaluation must remain grounded in human decision outcomes; automatic metrics are valuable but imperfect complements. In the future work, we envision hybrid evaluation protocols that integrate human decision accuracy, domain-specific financial measures, and calibrated, auditable LLM judgments.

We hope Earnings2Insights catalyzes sustained progress on decision-centric financial text generation. By releasing the task design, data splits, and evaluation scripts, and by documenting successful agentic and retrieval-augmented patterns, we aim to provide a practical foundation for research and deployment of human-centered advisory systems in finance.

References

Tianshi Cai, Guanxu Li, Nijia Han, Ce Huang, Zimu Wang, Changyu Zeng, Yuqi Wang, Jingshi Zhou, Haiyang Zhang, Qi Chen, Yushan Pan, Shuihua Wang, and Wei Wang. 2025. FinDebate: Multi-Agent Collaborative Intelligence for Financial Analysis. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Pulkit Chatwal, Mann Bajpai, Priyanshu Deswal, Harish Pratap Singh, and Santosh Kumar Mishra. 2025. Meta Prompting for Analyst Report Generation: Turning Earnings Calls into Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021a. *From opinion mining to financial argument mining*. Springer Nature.

Chung-chi Chen, Jian-tao Huang, Hen-hsen Huang, Hiroya Takamura, and Hsin-hsi Chen. 2024. SemEval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1482–1491, Mexico City, Mexico. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Parag Dakle, Sai Krishna Rallabandi, Nikhi Kohli, Khyati Morparia, Ojas Raundale, and Preethi Raghavan. 2025. Jetsons at the FinNLP-2025 - Earnings2Insights: Persuasive Investment Report Generation Using Single And Multi-Agent Frameworks. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Tomas Goldsack, Yang Wang, Chenghua Lin, and Chung-Chi Chen. 2025. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10576–10593, Abu Dhabi, UAE. Association for Computational Linguistics.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Jiantao Huang, Chung-Chi Chen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2024. Numhg: A dataset for number-focused headline generation. In *LREC/COLING*.

Yu-Shiang Huang, Chuan-Ju Wang, and Chung-Chi Chen. 2025. Decision-oriented text evaluation. *arXiv preprint arXiv:2507.01923*.

Vivian Lai, Chacha Chen, Alison Smith-Renner, Q. Vera Liao, and Chenhao Tan. 2023. Towards a science of human-ai decision making: An overview of design space in empirical human-subject studies. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, page 1369–1385.

Chuang Liu, Junzhuo Li, and Deyi Xiong. 2023. TabCQA: A tabular conversational question answering dataset on financial reports. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 196–207, Toronto, Canada. Association for Computational Linguistics.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2022. ECTSum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10893–10906, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shivika Prasanna and Hui Su. 2025. LangKG at the FinNLP 2025 - Earnings2Insights: Task-Adaptive LLMs To Generate Human-Persuasive Investment Reports. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Saisab Sadhu, Biswajit Patra, and Tannay Basu. 2025. Structured Adversarial Synthesis: A Multi-Agent Framework for Generating Persuasive Financial Analysis from Earning Call Transcripts. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Ranjan Satapathy, Raphael Liew, Joyjit Chattorj, Erik Cambria, and Rick Goh. 2025. From Earnings Calls to Investment Reports: Evaluating Role-based Multi-Agent LLM Systems. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Gaurangi Sinha, Rajarajeswari Palacharla, and Manoj Balaji Jagadeeshan. 2025. Agentic LLMs for Analyst-Style Financial Insights: An LLM Pipeline for Persuasive Financial Analysis. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Takehiro Takayanagi, Kiyoshi Izumi, Javier Sanz-Cruzado, Richard McCreadie, and Iadh Ounis. 2025a. Are generative ai agents effective personalized financial advisors? In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 286–295, New York, NY, USA. Association for Computing Machinery.

Takehiro Takayanagi, Masahiro Suzuki, Kiyoshi Izumi, Javier Sanz-Cruzado, Richard McCreadie, and Iadh Ounis. 2025b. Finpersona: An llm-driven conversational agent for personalized financial advising. In *European Conference on Information Retrieval*, pages 13–18. Springer.

Takehiro Takayanagi, Hiroya Takamura, Kiyoshi Izumi, and Chung-Chi Chen. 2025c. Can GPT-4 sway experts' investment decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 374–383, Albuquerque, New Mexico. Association for Computational Linguistics.

Mingrui Tan, Yang Liu, Kun Gao, Fei Gao, and Yuting Song. 2025. SI4Fin at Earnings2Insights: LLM-Based Analyst Report Generation for Earnings Calls. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

Weijie Yang, Junbo Peng, and Weijie Yang. 2025. Beyond Summaries: Multi-Agent Generation of Investment Reports with Text, Tables, and Charts. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Beyond Summaries: Multi-Agent Generation of Investment Reports with Text, Tables, and Charts

Weijie Yang
University of California, Berkeley
raphaelyang1998@berkeley.edu

Junbo Peng, Ph.D
Georgia Institute of Technology
junbo.peng@gatech.edu

Abstract

We approach the Earnings2Insights shared task by combining dataset enrichment with a multi-agent report generation framework. Starting with the official 64 transcripts, we expand the dataset to 207 earnings calls by crawling additional quarters from public sources, providing richer temporal context. Using this expanded corpus, we implement a multi-agent system based on AutoGen: a Writer agent generates reports, a Reviewer refines content, a Stylist enhances presentation, and a Chart agent creates financial tables and visualizations (e.g., EPS trends). The resulting reports integrate text, tables, and charts, closely resembling professional analyst reports. Our approach demonstrates that multi-agent collaboration significantly improves factual accuracy and decision-making utility in the generation of financial reports.

Keywords: Multi-agent System, Text and Table Integration, Temporal Context

1 Introduction

Earnings call transcripts (ECTs) are an indispensable source of financial information, providing detailed discussions between corporate executives and analysts regarding past performance, forward-looking guidance, and potential risks. These transcripts often span thousands of words, making it difficult for investors and practitioners to efficiently extract insights. Traditional NLP research in finance has largely concentrated on tasks such as sentiment classification, news impact detection, and abstractive summarization of earnings calls (Araci, 2019; Yang et al., 2020; Mukherjee et al., 2022). However, these approaches focus primarily on textual compression or surface-level sentiment, and they fall short of producing structured, decision-oriented output that resembles professional analyst reports.

In this work, we aim to bridge the gap between transcript summarization and actionable investment insight generation. Specifically, we explore how large language models (LLMs) can be adapted to transform raw earnings call transcripts into structured reports that highlight company performance, risks, opportunities, and potential investment implications. Our framework integrates a multi-agent methodology to combine factual grounding with financial-domain knowledge.

Our approach is evaluated based on the method in (Takayanagi et al., 2025; Huang et al., 2025), where annotators are instructed to make investment decisions guided by the generated reports, and the accuracy of these decisions serves as the primary evaluation criterion. Consequently, the reports must not only indicate the appropriate course of action but also present the analysis in a convincing manner that can effectively persuade investors to adopt the recommended guidance.

Our contributions are as follows:

- A novel framework for transforming ECTs into structured investment reports, leveraging Retrieval-Augmented Generation (RAG) to enhance and enrich the transcript information.
- Implementation of a multi-agent workflow for intelligent report generation, combining multiple agents for tasks such as writing, reviewing, and styling.
- Adoption of a multi-modal paradigm for report generation, incorporating both charts and tables to present analysis in a comprehensive and actionable format.

2 Related Work

2.1 Financial Text Summarization

Summarization of long financial documents has been a central focus in financial NLP. The ECTSum dataset (Mukherjee et al., 2022) provides bullet-point summaries of earnings call transcripts, enabling research on abstractive summarization of

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

long financial dialogues. Earlier initiatives such as the Financial Narrative Summarisation shared task (El-Haj et al., 2020) also explored summarization of financial reports, while subsequent work combined extractive and abstractive methods for financial documents (Zmandar et al., 2021). More recent advances in efficient attention mechanisms have further improved the ability of neural models to capture salient information from long earnings calls (Huang et al., 2021). While these approaches enhance readability, they are not explicitly designed to produce investment-oriented outputs.

2.2 Financial NLP Beyond Summarization

Other lines of research in financial NLP include sentiment analysis and ESG issue classification. These tasks demonstrate the feasibility of domain-adapted models such as FinBERT (Araci, 2019) and ESG-BERT (Tseng et al., 2023) but their outputs remain limited to classification labels, without generating narrative insights comparable to analyst reports.

2.3 Large Language Models in Finance

Recent advances in LLMs (Chung et al., 2024; Achiam et al., 2023; Touvron et al., 2023) have shown strong generalization ability across domains including finance, including question answering (Chen et al., 2021). Nevertheless, challenges remain in mitigating hallucination, grounding generation in numerical and contextual evidence, and ensuring consistency with domain conventions. Our work contributes to this space by systematically studying how LLMs can be adapted for investment-style generation tasks.

3 Dataset and Task Setting

3.1 Data and Shared Task

The Earnings2Insights shared task builds on an official dataset of 64 earnings call transcripts (ECTs) from publicly listed companies, spanning specific quarters within fiscal years. Each transcript contains prepared remarks, and Q&A, thereby capturing the complete contents for the earnings call. Of these, 40 transcripts are paired with expert-written summaries regarding quantitative data. The task requires to generate investment analysis reports, thereby simulating a realistic decision-support scenario for financial analysts.

In this paper, we adopt a multi-agent framework that generates reports for a target quarter while leveraging transcripts from the preceding fiscal year as context. The framework integrates specialized agents to analyze text, extract quantitative indicators, and produce structured outputs—including summaries, tables, and visualizations. We evaluate the system both with standard text generation metrics and with task-oriented criteria assessing the practical utility of the reports for investment decision-making.

Data Source	Count
ECTSum subset	40
Professional subset	24
Data Enrichment (Web Crawling)	143

Table 1: Dataset summary. The official set includes 64 transcripts (40 ECTSum, 24 Professional), expanded with 143 additional transcripts from public sources for a total of 207.

3.2 Data Enrichment and Retrieval-Augmented Generation

To complement the official dataset, we constructed a supplementary corpus of historical earnings call transcripts by web crawling publicly available sources. Specifically, we collected transcripts from *The Motley Fool*[1] and *Alpha Street*[2], two widely used platforms that publish earnings call transcripts shortly after company releases. For each target quarter in the shared task (e.g., Q3 2020), we retrieved transcripts from the four preceding quarters, thereby extending the temporal context to a full fiscal year. This enrichment allows the model to capture trends and dynamics that cannot be inferred from a single quarter in isolation.

We further extracted structured quantitative metrics—with particular emphasis on earnings per share (EPS)—from all transcripts. These figures provide a reliable backbone for factual grounding and enable consistency checks in generated reports.

All transcripts were then parsed and indexed into a RAG knowledge base. The documents were segmented into *Prepared Remarks* and *Q&A sessions*, and speaker roles were explicitly aligned with their utterances. This structured representation enables fine-grained retrieval, allowing the generation system to selectively access relevant passages during report synthesis. Overall, the supplementary dataset and retrieval infrastructure supply both

[1] https://www.fool.com/earnings-call-transcripts/

[2] https://www.alphastreet.com

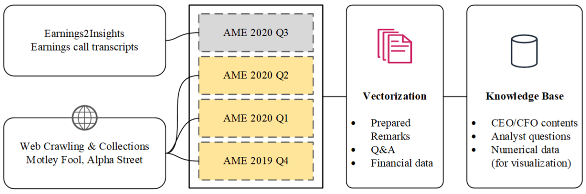


Figure 1: RAG data enrichment pipeline. Official Earnings2Insights transcripts are supplemented with additional quarters by web crawling, vectorized into structured segments (prepared remarks, Q&A, financials), and stored in a knowledge base for retrieval.

broader historical coverage and precision access mechanisms, directly enhancing the accuracy, contextuality, and richness of the generated investment reports.

4 Methodology

4.1 Framework Design

Our approach integrates a multi-agent methodology, leveraging AutoGen's architecture to combine factual grounding with financial-domain knowledge. The system includes several specialized agents: highlight summarizer, content writer, content editor, styling agent, and chart agent, each responsible for specific tasks in the report generation process, ensuring the production of coherent, well-structured, and persuasive investment reports. This multi-agent framework enhances efficiency and precision, mimicking the process followed by human analysts in generating high-quality financial reports.

4.2 Methodology Components

- **Chart Agent:** The chart agent is responsible for extracting and structuring financial data (EPS, revenue, expenses) from the earnings call transcripts. This agent processes the raw data into a consistent JSON format, which is essential for maintaining the integrity and consistency of the financial summary. By structuring financial data in this way, the agent ensures that all subsequent steps in the report generation process can rely on well-organized and standardized inputs. This step is crucial for ensuring the accuracy and clarity of financial metrics, a critical component in investment decision-making (Kang et al., 2019).

- **Highlight Agent:** The highlight agent plays a central role in distilling key insights from the earnings calls. Given the extensive length of these transcripts, it focuses on extracting concise summaries related to five primary areas: Financial Trends, Strategic Shifts, Operational Updates, Management Tone and Forward Guidance. The use of highlight agents to extract meaningful insights from large documents has shown effectiveness in similar financial NLP tasks (Zhu et al., 2020).

- **Report Writer Agent:** The report writer is responsible for drafting the full investment research report based on the four quarters of earnings call transcripts. It uses the financial data summary from the chart agent and the extracted insights from the highlight agent to generate the following sections of the report: Executive Summary, Investment Thesis, Financials, Valuation, Catalyst Outlook, Risks. The report writer ensures the analysis is in-depth, objective, and professionally written, adhering to industry standards. The ability to generate comprehensive financial reports is supported by recent advances in LLMs for document generation (Raffel et al., 2020).

- **Content Editor Agent:** After the report is drafted, the content editor reviews and refines the text for grammar, accuracy, and logical consistency. This agent checks that all claims and numbers are well-supported by the original earnings call transcripts and ensures that the arguments are clearly written and logically structured. The editor's role is critical in maintaining the overall quality and reliability of the report, ensuring it meets the high standards expected in institutional financial analysis.

- **Styling Agent:** The styling agent applies an institutional writing style to the report, ensuring that the language, tone, and formatting align with the expectations of professional equity research reports. This includes adhering to formal conventions in financial writing, ensuring clarity and precision in presenting complex financial data. The final version of the report is polished and ready for institutional investors, making it suitable for high-stakes financial decision-making (Schumaker et al., 2009).

4.3 Report Generation Process

The report generation process begins with the Chart Agent, which extracts and structures financial data

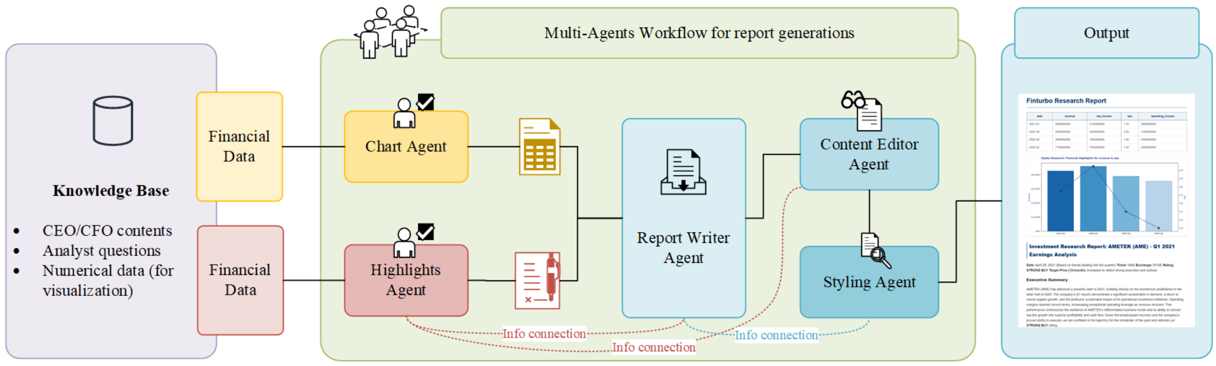


Figure 2: Model Architecture Diagram. This diagram illustrates the flow of tasks within a multi-agent system, showing the interactions between agents (Chart Agent, Highlights Agent, Report Writer Agent, Content Editor Agent, and Styling Agent) for generating an investment research report from earnings call transcripts and financial data.

from the earnings call transcripts. The Highlight Agent then processes the transcripts to extract key insights, focusing on financial trends, strategic shifts, and operational updates. These insights are passed to the Report Writer, who drafts the full investment report, integrating both the structured financial data and the key findings. The Content Editor reviews and refines the report for grammar, accuracy, and logical coherence. Finally, the Styling Agent applies the appropriate institutional style to ensure the report meets professional standards.

Throughout this process, agent connections play a crucial role in ensuring smooth collaboration between agents. The Highlight Agent's output feeds directly into both the Report Writer and the Content Editor, ensuring that the key insights inform both the drafting and the review stages. The Content Editor, in turn, passes the refined report to the Styling Agent, which focuses on formatting and style. These connections allow each agent to specialize in a specific task while maintaining coherence across the entire report generation process, resulting in a high-quality and consistent final report.

5 Experiments and Results

5.1 Model Selection

We chose Google Gemini-Pro 2.5 for its ability to process long token sequences, making it well-suited for earnings call transcripts. Gemini-Pro 2.5 outperformed alternatives like Qwen2.5 and DeepSeek-R1, particularly in handling extended contexts, which is crucial for financial analysis. While Qwen2.5 excels in generating concise outputs and DeepSeek-R1 in knowledge-intensive tasks, Gemini-Pro 2.5's ability to maintain context over long passages gives it a distinct advantage in this task (Brown et al., 2020).

5.2 Evaluation Results and Analysis

We evaluated the performance of our system using two distinct evaluation methods: a human evaluation based on the effectiveness of the reports in guiding investment decisions and a metric-based evaluation using clarity, logic, persuasiveness, and readability.

5.2.1 Human Evaluation

The official evaluation methodology, as described in [1], involves annotators making investment decisions (Long or Short) for the next day, week, and month based on the provided reports. The accuracy of these decisions is used as the primary evaluation metric. Our results are summarized in the table below:

Evaluation Metric	Score
Average Accuracy	0.524
Day	0.504
Week	0.568
Month	0.5

Table 2: Human evaluation results, showing the accuracy of investment decisions over different time frames.

The evaluation results show that the model performed reasonably well across different time frames. For the next day, the accuracy in predicting investment decisions was 50.4%, indicating that the model was able to make short-term predictions with a moderate degree of success. Over the next week, the accuracy improved to 56.8%, suggesting better performance in the medium-term prediction. However, for the next month, the accuracy dropped slightly to 50%, reflecting challenges in making reliable long-term predictions. These results highlight the model's strength in short-term decision-making, while also indicating areas for improvement in longer-term forecasting.

These results highlight that while the model performed reasonably well in predicting short-term and medium-term investments, the accuracy could be further improved for longer time frames.

5.2.2 Metric-based Evaluation

The model was evaluated across several dimensions including clarity, logic, persuasiveness, readability, and usefulness. The results are summarized in the table below:

Dimension	Score
Clarity	5.02
Logic	5.39
Persuasiveness	4.9
Readability	4.86
Usefulness	5.4
Likert Score	5.4

Table 3: Metric-based evaluation results, including various quality dimensions.

The evaluation results across various dimensions show that the model performed well in several areas. For clarity, the reports were rated 5.02, indicating that the information was presented in an understandable manner. The logical structure of the reports received the highest rating of 5.39, reflecting that the generated reports were coherent and well-reasoned. In terms of persuasiveness, the model achieved a score of 4.9, demonstrating that the investment insights were convincing, though there is room for improvement in making stronger recommendations. The readability score of 4.86 suggests that the reports were generally easy to read, with some areas where improvements could enhance the flow of the text. Finally, the usefulness score of 5.4 reflects that the reports provided valuable insights for decision-making.

Conclusion

This work addressed the task of generating structured investment reports from earnings call transcripts, a critical task in financial analysis. We employed a multi-agent architecture using Google Gemini-Pro 2.5, chosen for its ability to process long token sequences and handle complex financial language. The results demonstrate that the model performed well in generating clear, logical, and useful reports, with competitive performance in short- and medium-term investment predictions. However, challenges remain in improving long-term forecasting accuracy. Overall, this approach shows promise in automating financial report generation, but further refinement is needed to enhance the model's predictive capabilities and the persuasiveness of its recommendations.

Limitations

The performance of our approach is influenced by several factors. First, the effectiveness of the generated reports depends on the quality and completeness of the earnings call transcripts and financial data. Inconsistent or incomplete data could impact the accuracy of the output. Second, while Retrieval-Augmented Generation (RAG) helps incorporate external knowledge, challenges remain in fully capturing complex financial strategies and contextual nuances. Finally, the addition of more agents in the multi-agent system introduces greater complexity to the workflow, which could increase the likelihood of errors and affect the overall coherence of the final report.

Code Availability

The code is available at `https://github.com/RaphaelYangWJ/earnings2insights`.

Acknowledgments

This work was supported by the FinEval Shared Task organizers, who provided both the dataset and the evaluation framework for our research. We appreciate the effort put into curating the earnings call transcripts and financial data, which were essential for the successful completion of this project.

References

Josh Achiam and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Tom B. Brown and 1 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Zhiyu Chen and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*.

Hyung Won Chung and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Mahmoud El-Haj and 1 others. 2020. The financial narrative summarisation shared task (fns 2020). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*.

Luyang Huang and 1 others. 2021. Efficient attentions for long document summarization. *arXiv preprint arXiv:2104.02112*.

Yu-Shiang Huang and 1 others. 2025. Decision-oriented text evaluation. *arXiv*.

L. Kang and 1 others. 2019. Financial news prediction with deep neural networks. In *Proceedings of the 2019 IEEE International Conference on Big Data*.

Rajdeep Mukherjee and 1 others. 2022. Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. *arXiv preprint arXiv:2210.12467*.

C. Raffel and 1 others. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

R. P. Schumaker and 1 others. 2009. A survey of news articles, text mining, and analysis. *Journal of Information Science*, 35(5):594–606.

Takehiro Takayanagi and 1 others. 2025. Can gpt-4 sway experts' decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*.

Hugo Touvron and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yu-Min Tseng and 1 others. 2023. Dynamicesg: A dataset for dynamically unearthing esg ratings from news articles. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

X. Zhu and 1 others. 2020. A survey of financial text mining. In *Proceedings of the 2020 International Conference on Big Data*.

Nadhem Zmandar and 1 others. 2021. Joint abstractive and extractive method for long financial document summarization. In *Proceedings of the 3rd Financial Narrative Processing Workshop*.

A Appendix

A.1 Sample JSON Data

Here is a sample of the JSON data generated by Chart Agent for AME Quarter 1, 2021:


```
{
  "date":"2021 Q1",
  "revenue":8550000000,
  "net_income":2100000000,
  "eps":1.75,
  "operating_income":2800000000
},
{
  "date":"2020 Q4",
  "revenue":9200000000,
  "net_income":2500000000,
  "eps":2.05,
  "operating_income":3100000000
},
{
  "date":"2020 Q3",
  "revenue":7800000000,
  "net_income":1800000000,
  "eps":1.5,
  "operating_income":2400000000
},
{
  "date":"2020 Q2",
  "revenue":7100000000,
  "net_income":1550000000,
  "eps":1.3,
  "operating_income":2000000000
}
```


A.2 Sample Tables and Diagram

Here is a sample of layout regarding tables and diagrams for a generated report:

date	revenue	net_income	eps	operating_income
2021 Q1	8550000000	2100000000	1.75	2800000000
2020 Q4	9200000000	2500000000	2.05	3100000000
2020 Q3	7800000000	1800000000	1.50	2400000000
2020 Q2	7100000000	1550000000	1.30	2000000000

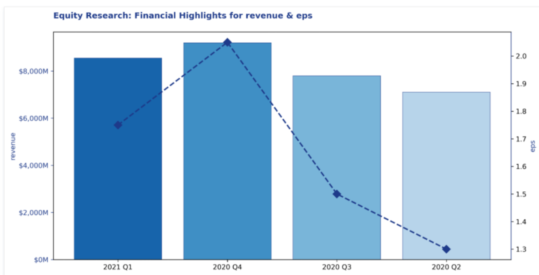


Figure 3: Sample diagram for generated report

From Earnings Calls to Investment Reports: Evaluating Role-based Multi-Agent LLM Systems

Ranjan Satapathy[1], Raphael Liew[2], Joyjit Chattoraj[1], Erik Cambria[2], Rick Siow Mong Goh[1]
[1]Institute of High Performance Computing (IHPC),
Agency for Science, Technology and Research (A∗ STAR), Singapore,
[2]Nanyang Technological University, Singapore
Correspondence: satapathy_ranjan@a-star.edu.sg

Abstract

This paper presents a novel multi-agent framework leveraging LLMs for automated financial analysis and investment report generation from earnings call transcripts. Traditional financial analysis struggles with increasing volumes of unstructured data. We propose a collaborative multi-agent system that mimics professional analyst team structures through role specialization. Our framework employs three specialized agents: Analyst, Writer, and Editor, that collaborate through structured workflows with tool support for financial data retrieval and sentiment analysis. Through extensive human evaluation on the Prolific platform, we demonstrate that our system achieves good accuracy in guiding financial decisions, placing it competitively among twelve evaluated systems. The system scores high on human quality assessment, with particularly strong performance in usefulness, indicating practical value for investment decision-making. In automatic evaluation, our system outperforms professional analyst reports most of the time, validating its competitive quality. Our findings provide empirical evidence that role-based agent collaboration offers a balanced approach to AI-generated financial analysis, demonstrating stable performance that prioritizes practical utility over surface-level report quality.

1 Introduction

Entity engagement and investment target prioritization have become increasingly critical for institutional investors navigating dynamic financial markets. This process heavily relies on financial analysis, which has traditionally depended on manual examination of structured data such as balance sheets (Loughran and McDonald, 2016). However, the exponential growth of unstructured data sources - including earnings calls, patent filings, and social media sentiment has rendered traditional approaches inefficient in capturing real-time market insights (Du et al., 2024b).

Early deep learning solutions attempted to automate parts of this process through sentiment analysis models and neural networks for risk forecasting (Loughran and McDonald, 2016). However, these approaches face significant challenges with data drift, where gradual or rapid changes in the input data distribution cause a degradation of model performance (Lu et al., 2018). Market dynamics such as regulatory changes, emerging sectors, and macroeconomic shocks alter data distributions, requiring costly retraining with substantial computational resources and labor intensive data labeling (Alzubaidi et al., 2021).

Recent advances in generative AI, particularly LLMs, offer a compelling alternative. Pretrained on massive diverse corpora, LLMs can interpret complex contextual relationships without task-specific retraining (Du et al., 2024a). They excel at capturing subtle linguistic cues for nuanced sentiment analysis and have demonstrated strong performance in summarization, question answering, and market sentiment prediction (Yang et al., 2024a). Domain-specialized models like BloombergGPT (Wu et al., 2023) and FinGPT (Yang et al., 2023) further demonstrate the benefits of adapting general-purpose models to financial text.

However, existing systems predominantly adopt single agent paradigms in which one LLM handles the entire analysis pipeline. While effective for narrow applications, these frameworks struggle with hallucinations causing factual inaccuracies (Kang and Liu, 2023) and incomplete coverage when tackling complex tasks such as investment reporting. Moreover, current financial AI frameworks lack realistic organizational modeling, failing to replicate the structured workflows and division of labor characteristic of professional analyst teams (Yu et al., 2023). In this paper, we address such limitations by exploring multi-agent LLM systems for financial analysis and investment report generation.

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

We design and implement a collaborative agent framework powered by GPT-4.1 that analyzes financial textual data and produces well-informed investment recommendations. Our key contributions are as follows:

- A novel multi-agent framework that mimics professional analyst team structures through specialized role assignment
- Evidence that agent collaboration with iterative feedback significantly improves report quality and factual accuracy

2 Related Work

2.1 Explainability and Interpretability in Financial AI

The deployment of AI systems in financial contexts demands not only accuracy but also transparency and interpretability, particularly given regulatory requirements and the high-stakes nature of investment decisions. Recent work has comprehensively examined the landscape of explainable AI (XAI) in finance (Yeo et al., 2025b), highlighting the critical need for systems that can provide faithful and interpretable explanations for their outputs.

The challenge of generating trustworthy explanations from LLMs has received considerable attention. Yeo et al. (2025a) demonstrate through activation patching that natural language explanations from LLMs may not always faithfully represent their internal decision-making processes, raising important questions about the reliability of single-agent systems that lack verification mechanisms. This finding directly motivates our multi-agent approach, where the Editor agent serves as an additional layer of validation for the explanations and reasoning provided by other agents.

Interpretability concerns extend beyond individual model outputs to the reasoning processes themselves. Jie et al. (2024c) examine how interpretable reasoning explanations from prompted LLMs actually are, finding significant variability in explanation quality. Our multi-agent framework addresses this through role specialization: the Analyst agent provides data-grounded explanations, while the Editor ensures these explanations maintain logical consistency and clarity.

The extraction of interpretable rationales from financial text presents unique challenges. Jie et al. (2024b) propose semi-supervised approaches for extractive rationalization, which aligns with our Analyst agent's function of identifying and extracting key financial metrics and insights from earnings transcripts. Similarly, Ong et al. (2023) introduce aspect-based sentiment analysis for explainable finance, demonstrating that decomposing financial analysis into specific aspects (similar to our agent specialization) improves both performance and interpretability. The self-improvement capabilities of LLMs through knowledge detection (Jie et al., 2024a) suggest potential enhancements to our framework. While our current implementation uses fixed agent roles, future iterations could incorporate self-training mechanisms where agents learn from successful report generations to refine their specialized capabilities.

Finally, the challenge of structuring unstructured financial data, as addressed by Sun et al. (2024) in the context of ESG reports, parallels our task of converting free-form earnings call transcripts into structured investment reports. Their information extraction techniques could be integrated into our Analyst agent to enhance its ability to systematically extract and organize financial information.

These works collectively underscore that explainability and interpretability are not merely desirable features but essential requirements for financial AI systems. Our multi-agent framework inherently promotes explainability through its transparent workflow: each agent's contribution is distinct and auditable, the iterative refinement process is traceable, and the use of external tools for validation provides grounded explanations for financial claims.

2.2 Earnings Call Transcript Analysis

Researchers have explored transcript data for a variety of downstream tasks. For example, Sawhney et al. (Sawhney et al., 2021) examined bias in multimodal EC analysis for volatility prediction, while Keith and Stent [2] modelled analyst decision-making using semantic features of EC discourse. These studies highlight the predictive power of managerial language and financial context in shaping market outcomes.

Post-EC, two types of reports typically surface: journalistic summaries, which summarises headline figures and key takeaways into concise narratives, and analytical (equity research) reports, which offer a considerably more extensive evaluation of financial performance, managerial tone, and strategic implications for investment strategies (Vipond, 2024) (AlphaSense, 2025).

Although previous research has focused on automating journalistic summary (Mukherjee et al., 2022), automatic generation of analytical reports remains underexplored. By automating this complex output, we could significantly reduce an analyst's workload to allow timely dissemination of insights to investors, and improve the overall scalability of equity research. Hence, this gap motivates the exploration of emerging AI methods, such as Generative AI, to transform earnings call transcripts into structured, actionable equity research reports.

2.3 Generative AI in Financial Analysis

The financial sector has witnessed growing adoption of generative AI for analyzing complex textual documents. Recent studies demonstrate LLMs' strong performance in summarization, question answering, and sentiment extraction from corporate earnings calls, 10-K filings, and analyst briefings (Yang et al., 2024a; Chowdhery et al., 2023; Touvron et al., 2023). These models identify subtle language cues correlating with market movements, often outperforming human analysts in specific prediction scenarios (Hu et al., 2018).

Domain-adapted models further illustrate the benefits of financial corpus training. BloombergGPT (Wu et al., 2023) achieves state-of-the-art performance in sentiment analysis and entity recognition, while FinGPT (Yang et al., 2023) demonstrates that open-source fine-tuned models can rival proprietary approaches on financial NLP benchmarks. However, hallucination remains a primary concern, where models fabricate plausible but inaccurate statements—particularly problematic in high-stakes settings where small factual errors can mislead investors (Kang and Liu, 2023). This motivates research into retrieval-augmented generation (RAG) that constrains LLM outputs with reliable external data (Gao et al., 2024).

2.4 Single-Agent AI Systems

AI agents extend LLMs into autonomous, goal-directed entities that operate more like human workers (Park et al., 2023; Sumers et al., 2023). These systems incorporate planning capabilities for multi-step actions, memory mechanisms for context maintenance, and tool use for accessing external resources (Parisi et al., 2022). This design enables agents to retrieve data through APIs, compute metrics, and generate fact-grounded reports rather than relying on speculative language (Yu et al., 2023).

Recent frameworks have operationalized these concepts successfully. GPT-Engineer demonstrated LLM-driven software generation (Qian et al., 2023), while Toolformer showed that LLMs can self-learn API usage (Schick et al., 2023). In finance, FinMem introduced layered memory architecture enhancing trading agents' decision-making (Yu et al., 2023). However, single-agent systems face limitations as task complexity increases, remaining vulnerable to hallucinations and struggling with self-error correction without external feedback (Darwish et al., 2025).

2.5 Multi-Agent Systems and Role Specialization

To address single-agent limitations, recent research explores multi-agent systems where multiple LLMs interact under role-specific instructions (Hong et al., 2024; Qian et al., 2023; Du et al., 2025). The premise draws on collective intelligence: specialized agent groups can outperform single generalist models when roles and workflows are well-defined (Zhang et al., 2024; Salve et al., 2024; Lu et al., 2023; Yang et al., 2024b). Several frameworks demonstrate this approach, e.g., MetaGPT (Hong et al., 2024) and ChatDev (Qian et al., 2023). Each role activates domain-relevant behavior in the underlying LLM, producing more reliable outputs than single-agent prompting. Research highlights the effectiveness of iterative feedback and debate structures where agents critique and refine each other's work (Darwish et al., 2025).

Emerging literature applies multi-agent systems specifically to finance. Heterogeneous designs focusing on different error types improve financial sentiment analysis (Darwish et al., 2025). TradeMaster illustrates how reinforcement learning and multi-agent collaboration combine for quantitative trading (Sun et al., 2023). However, empirical exploration remains limited, with few systematic comparisons between single-agent and multi-agent approaches for financial analysis tasks.

3 Methodology

3.1 System Architecture Overview

We developed a multi-agent collaborative system (Figure 1) which is powered by OpenAI's GPT-4.1 as the underlying language model to ensure consistent capability across experiments, differing only in their approach to role specialization and agent interaction.

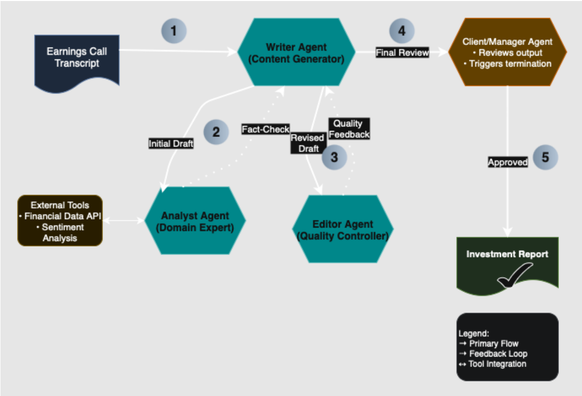


Figure 1: Multi-agent architecture for Investment Guidance

3.2 Multi-Agent Configuration

The multi-agent system employs three specialized agents collaborating through structured workflows:

Agent	Responsibilities
Analyst	Extracts financial data, performs fact-checking, calculates ratios, conducts sentiment analysis using external tools
Writer	Drafts and revises investment reports incorporating Analyst data and Editor feedback, maintains professional tone and structure
Editor	Reviews drafts for accuracy, completeness, and readability, provides actionable feedback for revisions

Table 1: Multi-agent system roles and responsibilities

Role-specific temperature settings optimize each agent's function: Analyst and Editor operate at 0.2-0.3 for maximum accuracy, while Writer uses 0.7 for natural language fluency. This configuration mimics real-world analyst teams where domain experts, writers, and editors collaborate iteratively.

3.3 Orchestration Framework

We implemented both systems using Microsoft AutoGen[1], an open-source framework for multi-agent LLM applications. AutoGen manages agent communication through its GroupChatManager, routing messages appropriately between agents. The framework handles message passing and turn-taking logic, simplifying implementation of iterative feedback loops.

[1] https://github.com/microsoft/autogen

Algorithm 1 Multi-Agent Workflow

Input: Earnings call transcript T, Instructions I
Writer creates initial draft D_0 from T
Analyst validates D_0 against external data:
  Extract metrics, calculate ratios
  Call external APIs for validation
  Generate structured feedback F_A
Writer revises draft: $D_1 = \text{revise}(D_0, F_A)$
repeat
  Editor reviews D_i for quality
    Check factual consistency
    Evaluate completeness and clarity
    Generate editorial feedback F_E
  Writer produces $D_{i+1} = \text{revise}(D_i, F_E)$
until Editor approves or max iterations reached

Client validates final draft D_n
Output: Investment report D_n

3.4 Dataset and Preprocessing

We used the official Earnings2Insights(Takayanagi et al., 2025)dataset released for the shared task. The dataset consists of 64 earnings call transcripts drawn from two subsets:

- **ECTSum subset (40 transcripts)**: aligned with the ECTSum benchmark, where each folder includes both a transcript and a reference summary. Participants may optionally leverage these summaries.

- **Professional subset (24 transcripts)**: matched with professional analyst reports. Only the transcripts are provided to participants; comparisons with analyst reports are conducted later by the organizers.

The transcripts were distributed in Markdown format, already structured with speaker metadata and sections (e.g., management remarks, Q&A). Since the files were ready for direct LLM ingestion, no additional preprocessing was required. Each transcript was processed by multi-agent systems to generate reports in JSON format.

3.5 External Tools and APIs

Both configurations access two specialized tools:

- **historicalFinancialData(ticker, year, quarter):** Retrieves quarterly metrics (EPS, revenue, cash flow, balance sheet) from Alpha

Vantage for year-over-year and quarter-over-quarter comparisons

- **analyzeMarketSentiment(ticker, year, quarter):** Collects news articles published within 30 days before the earnings call, ensuring realistic temporal constraints matching real analyst workflows

3.6 Evaluation Framework

3.6.1 Automatic Evaluation

Our evaluation follows the official shared task protocols from the Earnings2Insights competition (Takayanagi et al., 2025) where reports were evaluated automatically using an LLM-based judge following standardized guidelines:

- **Average Likert Score**: mean 1–7 rating across Persuasiveness, Logic, Usefulness, Readability, and Clarity.
- **Win Rate vs Analyst Report**: pairwise comparison against professional analyst reports, where win rate = Wins ÷ (Wins + Losses).

3.6.2 Human Evaluation

The organizers also conducted a large-scale human evaluation with 210 participants on the Prolific platform (176 retained after attention checks). Each participant reviewed 12 reports and two measures were collected:

- **Accuracy of financial decisions**: fraction of correct Buy/Neutral/Sell predictions, evaluated at day, week, and month horizons, then averaged.
- **Human Likert Scores**: 7-point scores for clarity, logic, persuasiveness, readability, and usefulness.

Together, these evaluations provide a rigorous test of system performance. Automatic scoring offers a scalable baseline, while human evaluation captures how well reports can actually guide and persuade investors in practice. This dual framework ensures that the final rankings reflect both the formal quality of the report and the impact of real-world decision making.

4 Discussion

Our experiments demonstrate that multi-agent collaboration significantly enhances the quality of AI-generated financial analysis. The multi-agent system achieved the highest financial decision accuracy (58.1%) among automated approaches. The human evaluation reveals interesting patterns in perceived quality versus actual utility. While some systems scored higher on individual Likert dimensions, our multi-agent approach achieved a balanced performance across all metrics, with particularly strong scores in logic (5.89) and persuasiveness (5.95). The correlation between Likert scores and decision accuracy suggests that report clarity and logical structure directly impact investment decision quality.

The Analyst agent's integration of external data proved particularly valuable, reducing hallucinations and ensuring quantitative claims align with verifiable sources. The Editor's quality control function, while contributing less to raw accuracy, substantially improved report professionalism and readability—critical factors for real-world deployment.

4.1 Comparison with Human Analysts

While our model achieves 52.2% decision accuracy, placing it in the middle tier of evaluated systems, several fundamental distinctions from human analysis remain:

- Limited ability to incorporate non-textual market signals or conduct primary research
- Absence of industry-specific intuition developed through years of experience
- Difficulty identifying subtle management communication patterns that experienced analysts recognize
- Inability to leverage professional networks for channel checks or proprietary information

The best-performing systems in our evaluation achieved average accuracy around 58%, with none exceeding 60%, suggesting a practical ceiling for current LLM-based approaches when relying solely on earnings transcript analysis. This performance gap underscores that AI systems should augment rather than replace human analysts. Our system's balanced scores for logic (5.61/7) and usefulness (5.74/7) indicate that while the system may not achieve top-tier decision accuracy, it produces reports that provide valuable foundational analysis for human refinement.

5 Conclusion

This paper presented a novel multi-agent framework for automated financial analysis that demonstrably improves investment decision-making. Through rigorous human evaluation with 176 participants, we showed that our multiagent system achieves 58. 1% accuracy in guiding financial decisions. The system also received strong quality ratings, with scores of 5.89/7 for logical structure and 5.95/7 for persuasiveness, indicating that structured LLM collaboration can address critical limitations of monolithic approaches.

Our contributions advance the field of financial NLP by providing empirical evidence for multi-agent superiority in complex analytical tasks. Ultimately, we envision multi-agent systems becoming integral to institutional investment processes, enhancing human decision-making while preserving the judgment and intuition that remain uniquely human contributions to financial analysis. The human evaluation results demonstrate that role specialization not only improves technical metrics but also translates to better investment outcomes, a critical validation that is often missing from AI research. The relationship between the quality dimensions of the report and the accuracy of the decisions provides actionable insights for the design of the system, suggesting that the logical structure and persuasiveness are key factors in generating actionable financial intelligence.

- Our model produces functionally useful reports that prioritize actionable insights over surface-level quality.
- The multi-agent orchestration may be creating overly balanced perspectives that excel at weekly horizons but miss immediate market signals.
- The gap between automatic (4.575) and human (5.49) Likert scores indicates our system's value is better recognized by human readers than automated evaluators.

Acknowledgments

This research/project is supported by the Ministry of Education, Singapore under its MOE Academic Research Fund Tier 2 (MOE-T2EP20123-0005) and by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore. The authors would like to thank the annotators for their careful work in reviewing our reports and providing scores.

References

AlphaSense. 2025. Equity research reports: A complete guide. https://www.alpha-sense.com/resources/equity-research-guide/. AlphaSense.

Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. 2021. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):1–74.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Ahmed M Darwish, Ehab A Rashed, and Gasser Khoriba. 2025. Mitigating llm hallucinations using a multi-agent framework. *Information*, 16(7):517.

Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. 2024a. An evaluation of reasoning capabilities of large language models in financial sentiment analysis. In *IEEE Conference on Artificial Intelligence*, pages 189–194, Singapore.

Kelvin Du, Frank Xing, Rui Mao, and Erik Cambria. 2024b. Financial sentiment analysis: Techniques and applications. *ACM Computing Surveys*, 56(9):220.

Kelvin Du, Yazhi Zhao, Rui Mao, Frank Xing, and Erik Cambria. 2025. A retrieval-augmented multi-agent system for financial sentiment analysis. *IEEE Intelligent Systems*, 40(2):15–22.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, and 1 others. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. International Conference on Learning Representations, ICLR.

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 261–269.

Yeo Wei Jie, Teddy Ferdinan, Przemyslaw Kazienko, Ranjan Satapathy, and Erik Cambria. 2024a. Self-training large language models through knowledge detection. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15033–15045.

Yeo Wei Jie, Ranjan Satapathy, and Erik Cambria. 2024b. Plausible extractive rationalization through semi-supervised entailment signal. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5182–5192.

Yeo Wei Jie, Ranjan Satapathy, Rick Goh, and Erik Cambria. 2024c. How interpretable are reasoning explanations from prompting large language models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2148–2164.

Haoqiang Kang and Xiao-Yang Liu. 2023. Deficiency of large language models in finance: An empirical examination of hallucination. *arXiv preprint arXiv:2311.15548*.

Tim Loughran and Bill McDonald. 2016. Textual analysis in accounting and finance: A survey. *Journal of accounting research*, 54(4):1187–1230.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. Chameleon: Plug-and-play compositional reasoning with large language models. *arXiv preprint arXiv:2304.09842*.

Rajdeep Mukherjee, Abhinav Bohra, Ananya Banerjee, Sopan Khosla Sharma, Madhav Hegde, Asif Ekbal Shaikh, Saurabh Shrivastava, Kaushik Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2022. Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Keane Ong, Wihan Van Der Heever, Ranjan Satapathy, Erik Cambria, and Gianmarco Mengaldo. 2023. Finxabsa: Explainable finance through aspect-based sentiment analysis. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 773–782. IEEE.

Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*.

Joon Sung Park, Joseph C O'Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.

Chen Qian, Xin Cong, Chenglong Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Aniruddha Salve, Saurabh Attar, Mandar Deshmukh, Shashank Shivpuje, and Ujjwal A Mitra. 2024. A collaborative multi-agent approach to retrieval-augmented generation across diverse data. *arXiv preprint arXiv:2412.05838*.

Ravinder Singh Sawhney, A. Aggarwal, and Rajiv Ratn Shah. 2021. An empirical investigation of bias in the multimodal analysis of financial earnings calls. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. 2023. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*.

Shuo Sun, Molei Qin, Wentao Zhang, Haochong Xia, Chuqiao Zong, Jie Ying, Yonggang Xie, Lingxuan Zhao, Xinrun Wang, and Bo An. 2023. Trademaster: A holistic quantitative trading platform empowered by reinforcement learning. *Advances in Neural Information Processing Systems*, 36:59047–59061.

Zounachuan Sun, Ranjan Satapathy, Daixue Guo, Bo Li, Xinyuan Liu, Yangchen Zhang, Cheng-Ann Tan, Ricardo Shirota Filho, and Rick Siow Mong Goh. 2024. Information extraction: Unstructured to structured for esg reports. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 487–495. IEEE.

Takehiro Takayanagi, Tomas Goldsack, Kiyoshi Izumi, Chenghua Lin, Hiroya Takamura, and Chung-Chi Chen. 2025. Earnings2Insights: Analyst Report Generation for Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China. Overview paper for the Earnings2Insights shared task (FinEval) at FinNLP 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

T. Vipond. 2024. Equity research report: A recommendation to buy, sell, or hold shares of a public company. `https://corporatefinanceinstitute.com/resources/valuation/equity-research-report/`. Corporate Finance Institute.

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 preprint arXiv:2303.17564*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *arXiv preprint arXiv:2306.06031*.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024a. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.

Zonglin Yang, Xinya Du, Junxian Li, Jie Zheng, Soujanya Poria, and Erik Cambria. 2024b. Large language models for automated open-domain scientific hypotheses discovery. In *Proceedings of ACL*, pages 13545–13565.

Wei Jie Yeo, Ranjan Satapathy, and Erik Cambria. 2025a. Towards faithful natural language explanations: A study using activation patching in large language models. In *In Proceedings of EMNLP*.

Wei Jie Yeo, Wihan Van Der Heever, Rui Mao, Erik Cambria, Ranjan Satapathy, and Gianmarco Mengaldo. 2025b. A comprehensive review on financial explainable ai. *Artificial Intelligence Review*, 58(6):1–49.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W Suchow, and Khaldoun Khashanah. 2023. Finmem: A performance-enhanced llm trading agent with layered memory and character design. *arXiv preprint arXiv:2311.13743*.

Wentao Zhang, Lingxuan Shen, Jingwei Yang, Haoran Zhu, Zhangyang Chen, Dongyan Sun, Jiazheng Zhang, Xiaodong Li, and Yuxuan Zhang. 2024. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. *arXiv preprint arXiv:2402.18485*.

Appendix

A Agent Initialization prompts

This section shares agent initialization prompts that we have used in the experiments.

Agent	Responsibilities
Writer	You are the Writer. Draft the investment report and revise it based on other agents' feedback. Do not rewrite from scratch unless asked; make targeted edits. Always return the updated full report only (Markdown). Populate sections from the transcript; use tables for structured data; replace placeholders when Analyst/Editor provide updates.
Analyst	You are the Analyst. Fact-check and correct financial metrics and ratios using tool data (e.g., @historicalfinancialdata(ticker,year,quarter)). Compute QoQ/YoY, fill missing values with N/A, and provide: (1) FINANCIAL ANALYSIS UPDATE, (2) FINANCIAL RATIOS UPDATE, (3) KEY HIGHLIGHTS UPDATE, and (4) FINANCIAL ANALYSIS SUMMARY UPDATE. Fetch news via @analyzemarketsentiment and supply a complete News Sentiment section or instruct omission if none. Hand off to Editor after updates.
Editor	You are the Editor. Review Sections 1–3 for completeness, clarity, structure, and consistency (tables, legends, formatting). Ensure Analyst updates and sentiment are integrated; remove any internal notes/placeholders. Produce an INVESTMENT RECOMMENDATION FEEDBACK block containing: Key Drivers, Major Risks, and Buy/Hold/Sell calls for Next Day/Week/Month with data-backed justifications and Catalysts. The Writer must update Section 4 accordingly.
Client	You are the Client/Investor. Review the latest Writer report against the checklist (sections, metrics, ratios, highlights, summary, risks, editor feedback integration, formatting). If all checks pass, reply TERMINATE; otherwise list failed checks with exact fixes required.

Table 2: Agent initialization prompts.

Cummins Inc. (CMI) Investment Report — Fiscal 2013 Q4

1. Financial Analysis
Key Highlights

- Revenue rose **7% YoY** to $4.59B, driven by North America.
- Net income: **$432M**, slightly below prior year on competition and costs.
- 2014 revenue growth outlook: **+4% to +8%**; margin gains from restructuring/cost control.

Key Financial Metrics

Metric	Current Q	Prev. Q	QoQ	Prev. Year	YoY
Revenue	$4.59B	$4.27B	+8%	$4.29B	+7%
EPS	$1.94	$1.94	0%	$2.00	-3%
Gross Profit	$1.16B	$1.11B	+4.5%	$1.06B	+9%
Operating Income	$553M	$524M	+5.5%	$465M	+19%
Net Income	$432M	$355M	+21.7%	$369M	+17%
Operating Cash Flow	$756M	$373M	+102.7%	$745M	+1.5%
Capex	$280M	$161M	+73.9%	$291M	-3.8%
Short-term Debt	$68M	$62M	+9.7%	$77M	-11.7%
Long-term Debt	$1.67B	$1.73B	-3.5%	$698M	+139.5%
Cash & Equivalents	$2.7B	$2.5B	+8%	$1.37B	+97%

Key Financial Ratios and Investment Insights

Metric	Current	Prev. Q	Prev. Y	Formula	Interpretation
Gross Margin (%)	25.37%	26.00%	24.65%	GP/Revenue	Slight YoY improvement; cost control.
Operating Margin (%)	12.05%	12.28%	10.83%	OI/Revenue	Efficiency improved YoY.
Net Margin (%)	9.42%	8.32%	8.60%	NI/Revenue	Profitability improved YoY.
EPS Surprise (%)	-2.02%	-8.06%	14.29%	(Actual-Est.)/Est.	Miss vs estimates this Q.
Free Cash Flow	$476M	$212M	$454M	OCF - Capex	Strong FCF generation.
Capex/OCF (%)	37.04%	43.16%	39.06%	Capex/OCF	Reasonable reinvestment.
Cash Conversion Ratio	1.75	1.05	2.02	OCF/NI	Strong cash conversion.
Net Debt	-$959M	-$706M	-$594M	Debt - Cash	Net cash position.
Current Ratio	2.565	2.515	2.285	CA/CL	Solid short-term liquidity.
Debt-to-Equity	0.232	0.253	0.117	Debt/Equity	Manageable leverage.

Concluding Summary — Cummins shows robust cash generation and improved profitability metrics. Strategy on cost management supports margins; watch international demand and regulatory uncertainty.

2. Market Analysis
Opening Remarks (summary)
"Revenue up 7% YoY to $4.59B; restructuring/cost reduction to lift margins. Near-term challenges in power generation/high-horsepower; growth expected in 2014 from acquisitions and launches."

Theme	Key Message
Strategy / Vision	Restructuring and cost reduction focus.
Market Outlook	Growth from acquisitions and new products.
AI / Innovation	Not specifically mentioned.

Competitive Landscape

Competitor	Mentioned?	Position	Commentary
Caterpillar	Yes	Turbines strength	Cummins lacks turbine products; differences in power-gen performance.

Industry & Regulatory Trends

Trend	Impact	Summary
Emission regulations	Mixed	Drives compliant demand but raises costs.

Impact Legend: Positive / Negative / Mixed / Neutral

Growth Opportunities & M&A

Opportunity	Description	Timing / Likelihood
Acquisitions	Distributor acquisitions expected to drive growth.	High (2014)

Customer Segments

Segment	Performance Summary
North America	Growth in medium-duty trucks; share gains.

3. Risk Assessment

Risk	Description	Likelihood	Impact (1–5)	Evidence
Market Demand	Weak int'l power-gen/mining demand.	Medium	4	Transcript indicates margin pressure.
Regulatory Compliance	Emission rules uncertainty.	High	3	Potential China impact.

Impact Scale: 1 Very Low … 5 Critical **Likelihood:** Low / Medium / High

4. Investment Recommendation

- **Key Drivers:** New product launches; distributor acquisitions; NA market share gains.
- **Major Risks:** International demand softness; regulatory uncertainty; volatility.
- **Recommendation:** *Next Day*—Hold; *Next Week*—Buy; *Next Month*—Hold.
- **Catalysts:** Acquisition integration, launches, stabilization in int'l markets.

Figure 2: Full example of a report generated with all agents.

FinDebate: Multi-Agent Collaborative Intelligence for Financial Analysis

Tianshi Cai[1,*], **Guanxu Li**[1,*], **Nijia Han**[2,*], **Ce Huang**[1], **Zimu Wang**[1,†], **Changyu Zeng**[1], **Yuqi Wang**[4], **Jingshi Zhou**[1], **Haiyang Zhang**[1], **Qi Chen**[3], **Yushan Pan**[1], **Shuihua Wang**[2], **Wei Wang**[1,†]

[1]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
[2]School of Science, Xi'an Jiaotong-Liverpool University, Suzhou, China
[3]School of AI and Advanced Computing, Xi'an Jiaotong-Liverpool University, Suzhou, China
[4]Shanghai Jiao Tong University, Shanghai, China

{Tianshi.Cai24,Guanxu.Li24,Nijia.Han23}@student.xjtlu.edu.cn
Zimu.Wang19@student.xjtlu.edu.cn, Wei.Wang03@xjtlu.edu.cn

Abstract

We introduce **FinDebate**, a multi-agent framework for financial analysis, integrating collaborative debate with domain-specific Retrieval-Augmented Generation (RAG). Five specialized agents, covering earnings, market, sentiment, valuation, and risk, run in parallel to synthesize evidence into multi-dimensional insights. To mitigate overconfidence and improve reliability, we introduce a safe debate protocol that enables agents to challenge and refine initial conclusions while preserving coherent recommendations. Experimental results, based on both LLM-based and human evaluations, demonstrate the framework's efficacy in producing high-quality analysis with calibrated confidence levels and actionable investment strategies across multiple time horizons.

1 Introduction

While the advent of large language models (LLMs) has catalyzed progress in NLP, the financial domain remains a high-value opportunity with strict operational and regulatory constraints, demanding accuracy, reliability, and explainability. Although LLMs can process vast volumes of unstructured financial data, their "next token prediction," trained on statistical correlations, makes the outputs fluctuate across prompts and runs. As a result, confidence is often miscalibrated, and statements may appear plausible without grounding in verifiable evidence (Zhang et al., 2024; Tatarinov et al., 2025), which are misaligned with requirements for verifiable reasoning and stable recommendations in this field.

Beyond the aforementioned limitations, long-form, multi-section analyst reports face document- and pipeline-level challenges. Evidence must be synthesized into a unified, coherent narrative while avoiding topic drift and refraining from claims unsupported by the underlying transcripts (Goldsack et al., 2024; Xia et al., 2025). A single passage can bear distinct implications across analytical dimensions, and design choices in chunking and querying materially influence what evidence is retrieved and how effectively it supports cross-aspect reasoning. During revision cycles, stance coherence and coverage can be degraded, leading to overlooked factors and unintended shifts in the investment thesis. Moreover, reasoning must remain traceable and reference-grounded without sacrificing readability or decision-oriented clarity (Li et al., 2024).

Practitioners have responded with several pragmatic strategies. Template-driven workflows impose discipline and stylistic consistency but weaken the alignment between cited evidence, intervening reasoning, and the report's final stance (Kang et al., 2025; Tian et al., 2025). Retrieval-Augmented Generation (RAG) anchors factual claims, yet integrating dispersed excerpts into a cohesive, multi-faceted narrative remains challenging (Jimeno-Yepes et al., 2024). Multi-agent collaboration and debate surface issues in short-form claims, but in chaptered, long-form analyst reports, they often struggle to maintain a consistent stance while covering all essential elements (Sun et al., 2024). These gaps motivate an approach that jointly stabilizes stance, expands evidence coverage and explicit risk articulation, and preserves reference traceability.

To address these gaps, we introduce **FinDebate**, a safety-constrained debate protocol that stabilizes the stance while strengthening evidence and risk articulation. As shown in Figure 1, a domain-specific RAG module and a team of role-specialized analyst agents first produce a chaptered draft. The debate phase then performs a bounded augmentation pass across roles and tasks: the pre-debate stance is fixed, roles are prohibited from changing direction, and every addition must be anchored to verifiable references. This design preserves the throughline of the investment rationale while improving coverage and verifiability, yielding analysis that remain

*Equal contribution.
†Corresponding authors.

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

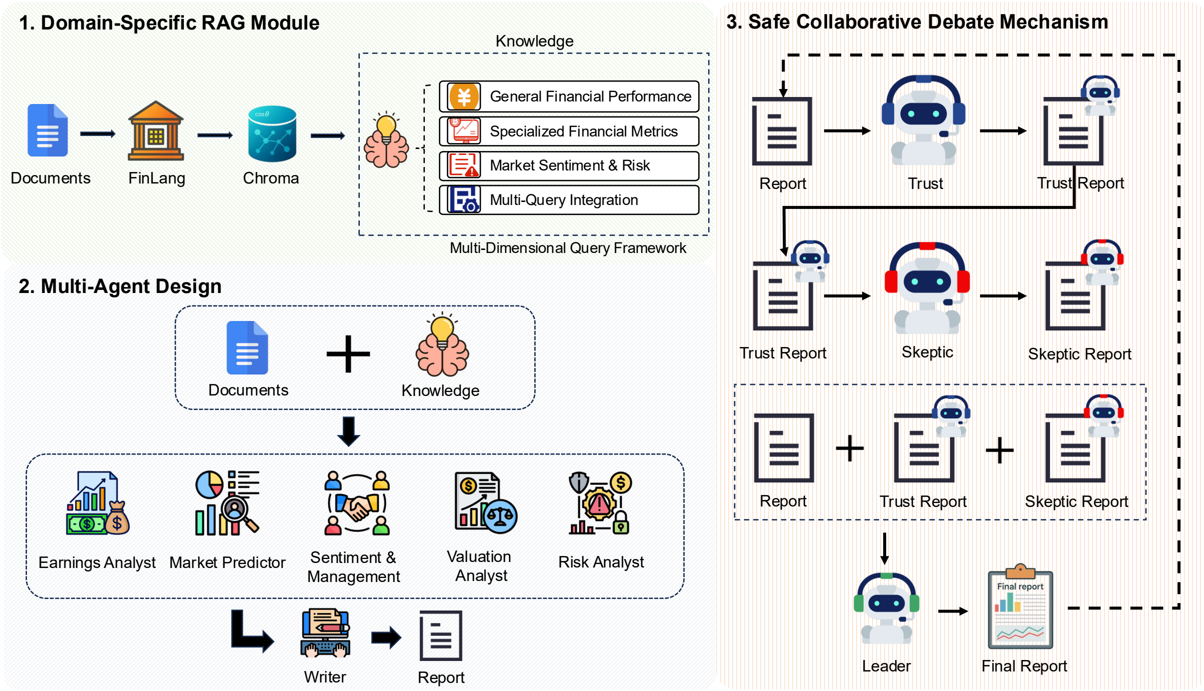


Figure 1: Overview of **FinDebate**, a multi-agent collaborative intelligence framework for financial analysis.

auditable and decision-oriented, as evidenced by LLM-based and human evaluations.

2 Methodology

Figure 1 shows an overview of **FinDebate**, which consists of three essential modules: (1) a *domain-specific RAG* module for document processing and evidence retrieval; (2) a *multi-agent analysis* module for an initial draft report; (3) a *debate mechanism* that yields the final report. An example of the task is shown in Appendix A. In this section, we introduce each of the modules in detail.

2.1 Domain-Specific RAG Module

Text Segmentation Strategy. Applying LLMs to financial analysis is constrained by limited context windows, which make it infeasible to process reports spanning hundreds of pages simultaneously. To address this, we propose a domain-specific RAG module with ChromaDB[1], which enables efficient indexing and similarity search over extensive financial documents, supporting low-latency retrieval and rapid downstream processing at scale.

To mitigate the information loss by naive fixed-size chunking, we adopt a context-sensitive segmentation strategy grounded in contextual chunking (Gunther et al., 2024). Instead of partitioning by fixed-length token counts, which is misaligned with the dense, highly structured nature of financial documents, we apply a recursive procedure that prioritizes semantic integrity: paragraph boundaries are preserved first, followed by sentence boundaries, and finally lexical/token boundaries. This hierarchy prevents destructive splits, producing self-contained, interpretable segments, resulting in a robust substrate for high-precision retrieval and reliable downstream reasoning.

Financial Embedding and Multi-level Retrieval. We encode the segmented passages with FinLang[2], a financial embedding model adapted from BGE (Zhang et al., 2023) via domain-specific fine-tuning. Selected for its in-domain retrieval effectiveness, FinLang captures the semantic essence of queries and grounds them in financial constructs such as investment risk, valuation metrics, market sentiment, and growth outlook. This domain alignment enables highly precise retrieval of evidence passages, facilitating analysis of the consistency between fundamentals and stock prices and whether current valuations are justified by projected growth.

Building on multi-level retrieval (Adjali et al., 2024), we conduct contextual retrieval across four dimensions: *general financial performance*, *specialized financial metrics*, *market sentiment & risk*, and *multi-query integration* (details in Appendix B), providing a solid analytical foundation for the subsequent multi-agent system.

[1] https://github.com/chroma-core/chroma

[2] https://huggingface.co/FinLang/finance-embeddings-investopedia

2.2 Multi-Agent Design

Single-model approaches exhibit notable shortcomings, often yielding superficial analysis due to their reliance on generalized methodologies and limited perspectives (Du et al., 2023). To overcome this limitation, we propose a multi-agent collaborative framework designed to perform in-depth financial analysis across five specialized domains. Each agent is tasked with analyzing the earnings call content from their respective domain-specific viewpoints. Afterwards, a report synthesis module integrates these individual analysis into a unified, insightful investment advisory report.

2.2.1 Agent Prompting Strategy

Each agent is equipped with a two-level prompt structure. The first level, system prompt, defines the agent's *professional identity* through four key components: (1) professional credentials (e.g., a CFA charterholder with 20 years' experience), (2) an authoritative background (e.g., roles at leading investment banks and hedge funds), (3) a clear mission (e.g., to assist in institutional investment decision-making), and (4) a high-quality standard (e.g., delivering institutional-grade output). The second level, user prompt, outlines the specific *analytical task* assigned to each agent, consisting of four elements: (1) analytical frameworks that guide systematic reasoning, (2) technical requirements specifying format and precision, (3) output specifications detailing the report structure and length, and (4) contextual integration of information retrieved through RAG. Together, these two levels ensure that the agents process both professional expertise and the ability to execute tasks effectively.

2.2.2 Agent Specialization

Our framework intentionally leverages five agents across different specialized analytical dimensions, establishing a holistic analytical framework that addresses the key facets of institutional investment decision-making. The core design principles of the agents are outlined below, with detailed prompts provided in Appendix C:

Professional Earnings Analyst specializes in financial statement analysis and performance evaluation. Key responsibilities involve assessing revenue quality, evaluating profitability and sustainability, and examining critical financial indicators such as net interest margin (NIM), asset quality, and capital adequacy ratios.

Professional Market Predictor is tasked with forecasting market trends across multiple timeframes. This includes analyzing immediate market responses to earnings reports, evaluating the sustainability of underlying fundamental drivers, and predicting long-term market positioning based on strategic developments.

Professional Sentiment Analyst specializes in evaluating management credibility and investor sentiment. This agent incorporates behavioral finance theories, such as anchoring effects and confirmation bias, quantifying measurable indicators like historical accuracy and transparency ratings, and translating psychological factors into actionable investment strategies.

Professional Valuation Analyst specializes in business valuation and investment recommendations. It applies a sector-specific Discounted Cashflow Model (DCF), which considers factors such as credit loss cyclicality and regulatory capital constraints, and employs dynamic weight allocation based on the reliability of various valuation methods, with a focus on verifiable business drivers.

Professional Risk Analyst provides comprehensive risk assessment and positions sizing recommendations. It evaluates various risk factors, such as credit, interest rate, and liquidity risk, while maintaining a balanced perspective to ensure realistic and actionable risk assessments.

2.2.3 Report Synthesis

Once the specialized agents complete their analysis, the system advances to the final stage. The Report Synthesis agent consolidates the individual outputs, extracts key financial indicators and, manages sentiment data, and generates a comprehensive report. This report is subsequently passed to the collaborative debate mechanism for further refinement, enhancing its accuracy and persuasiveness.

2.3 Safe Collaborative Debate Mechanism

2.3.1 Three-Agent Collaboration

Finally, we introduce a safe collaboration debate mechanism between three agents, motivated by established multi-agent debate methodologies (Du et al., 2023; Liang et al., 2024; Estornell and Liu, 2024). It enhances the quality of the report through a single-round optimization, while maintaining the core conclusions of the original analysis. This module consists of three agents: a Trust Agent, a Skeptic Agent, and a Leader Agent, with detailed prompts provided in Appendix D:

Trust Agent enhances the original report by providing supporting evidence, reinforcing its argumentative logic, and optimizing linguistic expression. Throughout this process, it is strictly prohibited from altering the directional tone (bearish to bullish) or modifying the 1-day/1-week/1-month investment recommendations.

Skeptic Agent refines the report by incorporating a risk management perspective. Its core responsibilities include identifying potential risk factors, suggesting hedge strategies, and improving the scenario analysis framework.

Leader Agent synthesizes the evidence enhancements from the Trust Agent and the risk analysis from the Skeptic Agent to produce the final optimized report. The resulting content retains all core conclusions from the original report, while employing more professional and persuasive language, and offering a clearer risk-return analysis.

2.3.2 Algorithm Design

Algorithm 1 outlines the overall design of the debate framework, employing safety-first principles to preserve the integrity of the original investment recommendations. It incorporates multiple verification mechanisms while achieving systematic quality improvements through a structured optimization process. The debate proceeds in a single round, effectively avoiding the thematic drift typically associated with multi-round iterations. Drawing from optimal rounds in related research, we compare the impacts of one-round and two-round debates, ultimately setting the maximum round to 1. The entire process involves only minor refinements, without making directional rewrites.

It is important to note that this debate mechanism is applicable only to scenarios where reports containing pre-existing investment recommendations require further refinement. It is not intended for generating reports from scratch or for enhancing texts that lack clear directional conclusions.

Algorithm 1 Safe Collaborative Debate

Input: R_0 (Original_Report), A (Agent_Analysis)
Output: R^* (Optimized_Report), L (Debate_Log)
Safety Check: Validate R_0 structure
if $\neg$has_recommendations(R_0) **then** return R_0
Trust Phase: $R_1 \leftarrow$ optimize(R_0, A)
  ∘ Preserve core elements of R_0
  ∘ Strengthen evidence ↑
Skeptic Phase: $R_2 \leftarrow$ review(R_1, A)
  ∘ Identify vulnerabilities in R_1
  ∘ Maintain structure integrity
Leader Phase: $R^* \leftarrow$ synthesize(R_2, A)
  ∘ Maximize persuasive power
  ∘ Preserve critical elements
Final Check: Validate R^* integrity
if core_compromised(R^*, R_0) **then** return R_0
return R^*, L

3 Experiments

3.1 Experimental Setup

Datasets. We conduct experiments on the Earnings2Insights shared task (Takayanagi et al., 2025), which focuses on generating investment guidance from earnings call transcripts. The task includes two sets of earnings call transcripts: 40 corresponding to ECTSum (Mukherjee et al., 2022), and 24 professional analyst reports.

Models and Setup. We employ comparative experiments using five state-of-the-art LLMs: GPT-4o (`2024-08-06`, Hurst et al., 2024), Gemini 2.5 Flash[3], Llama 4 Maverick[4], DeepSeek-R1 (`0528`, Guo et al., 2025), and Claude Sonnet 4[5]. For reproducibility and a fair comparison, all models are evaluated under identical generation parameters: a temperature of 0.6, a maximum output length of 6,500 tokens, a top-p sampling of 0.85, and a frequency penalty of 0.1. Consistent prompt templates and evaluations are applied across all models.

Baselines. To demonstrate the effectiveness of FinDebate, we compare the framework against the following two baselines: (1) **Zero-shot inference** directly processes incoming financial reports without relying on any additional information; (2) **Standard RAG** represents the traditional RAG approach with a general-purpose embedding model; (3) **Multi-agent generation** serves as an ablation study that removes the safe collaborative debate mechanism, so as to assess the contribution of the debate mechanism itself.

Evaluation Metrics. To ensure a rigorous yet tractable evaluation process, we sample 10 reports from the ECTSum dataset and 5 from the new, professional subset, assessing the quality of the models' financial analysis. Following the framework of Goldsack et al. (2024), we define an evaluation protocol spanning two core dimensions and implement

[3] `https://deepmind.google/models/gemini/flash/`
[4] `https://www.llama.com/models/llama-4/`
[5] `https://www.anthropic.com/claude/sonnet/`

Base Model	Zero-shot	Standard RAG	Multi-agent w/o Debate	FinDebate	Overall Improvement
GPT-4o	2.97	3.21	3.39	**3.58**	**+0.61**
Gemini 2.5 Flash	2.90	3.15	3.32	**3.50**	**+0.60**
Llama 4 Maverick	2.82	3.06	3.24	**3.41**	**+0.59**
DeepSeek-R1	2.77	3.02	3.10	**3.39**	**+0.62**
Claude Sonnet 4	3.03	3.27	3.45	**3.64**	**+0.61**

Table 1: Performance comparison of FinDebate across models. The best performance for each model is in **bold**.

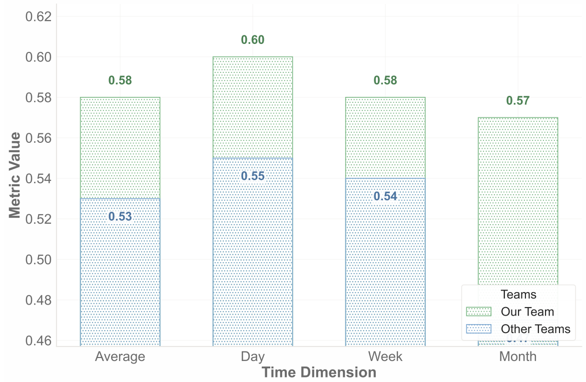


Figure 2: Human evaluation results on financial decision accuracy.

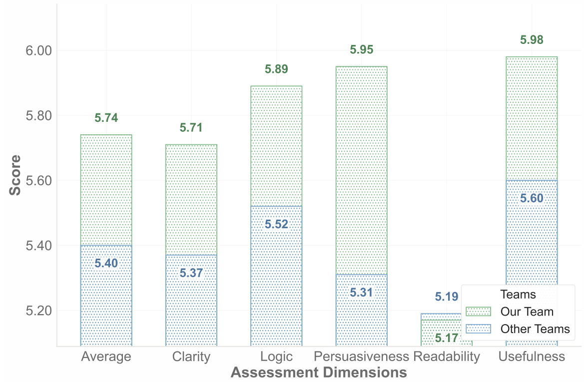


Figure 3: Human evaluation results on financial report quality.

it using GPT-4o (Hurst et al., 2024): (1) Textual Quality, covering *readability*, *linguistic abstractness*, and *coherence*; and (2) Financial Analysis Professionalism, encompassing *financial key point coverage*, *background context adequacy*, *management sentiment conveyance*, *future outlook analysis*, and *factual accuracy*. Each report is on a four-point scale (1 = poor to 4 = excellent). Detailed definitions of these dimensions and an illustrative prompt are provided in Appendix E.

Human evaluation is also conducted, with a primary focus on whether the report can effectively guide and persuade investors to make correct decisions, including the average accuracy of investment choices (*Long* or *Short*) made by experts for the next day, week, and month based on the reports, and the average Likert scores on clarity, logic, persuasiveness, readability, and usefulness.

3.2 Results and Analysis

Table 1 presents the main results of FinDebate compared to the zero-shot inference, standard RAG, and multi-agent generation baselines, and Figures 2 and 3 visualize human evaluation results. In comparison to practitioners, FinDebate demonstrates substantial improvements in financial decision prediction, achieving superior performance in clarity, logic, persuasiveness, and usefulness.

Our FinDebate framework consistently delivers substantial improvements across all five models, with performance gains ranging from 0.59 to 0.62, resulting in an average enhancement of 20.4%. These improvements are statistically significant ($p < 0.001$, via paired t-tests). FinDebate elevates performance from "satisfactory" levels (~ 3.0 points) to "excellent" standards (~ 3.6 points), highlighting the distinctive value of collaborative intelligence in complex reasoning tasks. This cross-model consistency further emphasizes the framework's universality and technical superiority. By transforming AI-driven financial analysis from a tool-assisted approach to a professional analyst-level capability, FinDebate establishes a foundation for real-world applications through its model-agnostic design and structured collaborative methodology.

4 Conclusion and Future Work

We introduce FinDebate, a multi-agent framework that integrates domain-specific RAG, specialized analytical agents, and a safe collaborative debate mechanism for financial analysis, generating institutional-grade financial reports with actionable, multi-horizon investment recommendations, effectively addressing key limitations in existing financial AI applications. In the future, we will extend this framework to broader financial domains, developing dynamic confidence adjustment mechanisms, and integrating with real-time market data. We will also transfer this system to other applications.

References

Omar Adjali, Olivier Ferret, Sahar Ghannay, and Hervé Le Borgne. 2024. Multi-level information retrieval augmented generation for knowledge-based visual question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *ArXiv*, abs/2305.14325.

Andrew Estornell and Yang Liu. 2024. Multi-llm debate: Framework, principals, and interventions. In *Neural Information Processing Systems*.

Tomas Goldsack, Yang Wang, Chen Lin, and Chung-Chi Chen. 2024. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *International Conference on Computational Linguistics*.

Michael Gunther, Isabelle Mohr, Bo Wang, and Han Xiao. 2024. Late chunking: Contextual chunk embeddings using long-context embedding models. *ArXiv*, abs/2409.04701.

Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 178 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mkadry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alexander Kirillov, Alex Nichol, Alex Paino, and 397 others. 2024. Gpt-4o system card. *ArXiv*, abs/2410.21276.

Antonio Jimeno-Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Ren-Yu Li. 2024. Financial report chunking for effective retrieval augmented generation. *ArXiv*, abs/2402.05131.

Xiaoqiang Kang, Zimu Wang, Xiao-Bo Jin, Wei Wang, Kaizhu Huang, and Qiufeng Wang. 2025. Template-driven llm-paraphrased framework for tabular math word problem generation. In *AAAI Conference on Artificial Intelligence*.

Ruosen Li, Zimu Wang, Son Quoc Tran, Lei Xia, and Xinya Du. 2024. Meqa: A benchmark for multi-hop event-centric question answering with explanations. In *Neural Information Processing Systems*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Conference on Empirical Methods in Natural Language Processing*.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, K. Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2022. Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Conference on Empirical Methods in Natural Language Processing*.

Xiaoxi Sun, Jinpeng Li, Yan Zhong, Dongyan Zhao, and Rui Yan. 2024. Towards detecting llms hallucination via markov chain-based multi-agent debate framework. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

Takehiro Takayanagi, Tomas Goldsack, Kiyoshi Izumi, Chenghua Lin, Hiroya Takamura, and Chung-Chi Chen. 2025. Earnings2Insights: Analyst Report Generation for Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China. Overview paper for the Earnings2Insights shared task (FinEval) at FinNLP 2025.

Nikita Tatarinov, Siddhant Sukhani, Agam Shah, and Sudheer Chava. 2025. Language modeling for the future of finance: A quantitative survey into metrics, tasks, and data opportunities. *ArXiv*, abs/2504.07274.

Yong-En Tian, Yu-Chien Tang, Kuang-Da Wang, An-Zi Yen, and Wen-Chih Peng. 2025. Template-based financial report generation in agentic and decomposed information retrieval. *ArXiv*, abs/2504.14233.

Haotian Xia, Hao Peng, Yunjia Qi, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2025. Storywriter: A multi-agent framework for long story generation. *ArXiv*, abs/2506.16445.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *ArXiv*, abs/2310.07554.

Zhilu Zhang, Procheta Sen, Zimu Wang, Ruoyu Sun, Zhengyong Jiang, and Jionglong Su. 2024. Finbpm: A framework for portfolio management-based financial investor behavior perception model. In *Conference of the European Chapter of the Association for Computational Linguistics*.

A Dataset Example

Input Financial Earnings Call of the Example

```
## Financial Earnings Call

### Prepared remarks
**Operator**
: Greetings, and welcome to the ABM Industries Incorporated Third Quarter 2021 Earnings Call. [Operator Instructions] As a reminder, this conference is being recorded. It is now my pleasure to introduce David Gold, Investor and Media Relations. Thank you, you may begin.

**Investor Relations**
: Thank you for joining us this morning. With us today are Scott Salmirs, our President and Chief Executive Officer; and Earl Ellis, our Executive Vice President and Chief Financial Officer. We issued our press release yesterday afternoon announcing our third quarter fiscal 2021 financial results. A copy of this release and an accompanying slide presentation can be found on our corporate website. Before we begin, I would like to remind you that our call and presentation today contain predictions, estimates and other forward-looking statements. Our use of the words estimate, expect, and similar expressions are intended to identify these statements. Statements represent our current judgment of what the future holds. While we believe them to be reasonable, these statements are subject to risks and uncertainties that could cause our actual results to differ materially. These factors are described in a slide that accompanies our presentation, as well as our filings with the SEC. During the course of this call, certain non-GAAP financial information will be presented. A reconciliation of historical non-GAAP numbers to GAAP financial measures is available at the end of the presentation and on the company's website under the Investor tab. I would now like to turn the call over to Scott.

**CEO**
: Thanks, David. Good morning, and thank you all for joining us today to discuss our third quarter results. As detailed in yesterday's release, ABM generated strong third quarter results featuring double-digit growth in revenue, continued solid cash generation, and a 20% gain in adjusted earnings per share. Revenue growth was broad-based as each of our five business segments achieved year-over-year gains in revenue, aided by an improving business environment and the gradual reopening of the economy. Our team members once again executed well and continue to provide exceptional service to our clients. Overall, demand for ABM's higher margin virus protection services remained elevated in the quarter, underscoring ongoing client concerns regarding cleaning and disinfection of their facilities. As anticipated, demand for virus protection eased slightly in the third quarter compared to the second quarter of fiscal 2021, but remain well above pre-pandemic levels. The emergence of the Delta variant and rising COVID-19 cases nationally have gains heightened interest in the need for disinfection prevention measures, particularly in high traffic areas. As we look forward to 2022 and beyond, we believe that virus protection services will remain a contributor to our overall revenue as disinfection becomes a standard service protocol and facility maintenance programs. [...]
```

Output Financial Analysis of the Example

```
- abm industries q3 adjusted earnings per share $0.90.

- q3 gaap loss per share $0.20 from continuing operations.

- q3 adjusted earnings per share $0.90.

- raises adjusted earnings per share guidance for full year fiscal 2021.

- q3 revenue rose 10.7 percent to $1.54 billion.

- increasing guidance for full year 2021 adjusted income from continuing operations to $3.45 to $3.55 per share.
```

Figure 4: Dataset example of the Earnings2Insights shared task (Takayanagi et al., 2025), where the example is from the ECTSum subset (Mukherjee et al., 2022). Models receive an input financial earnings call with management remarks, Q&A sessions, etc., and generate a structured financial analysis report for investment recommendation.

B Professional Context Queries

Dimension	Query Content
General Financial Performance	**Core Metrics:** Financial Performance, Revenue, Earnings, Beat/Miss, Surprise, Financial Results **Forward Guidance:** Guidance, Outlook, Forecast, Expectations, Future Performance, Strategic Direction **Growth Indicators:** Growth Trends, Margin Expansion, Profitability, Cash Flow, Competitive Position; **Strategic Factors:** Catalysts, Opportunities, Product Launches, Market Expansion, Strategic Initiatives
Specialized Financial Metrics	**Interest & Lending:** Net Interest Margin (NIM), Loan Deposits, Credit Quality, Asset Quality **Asset Quality:** Non-Performing Assets (NPAs), Charge-Offs, Provision Loan Losses, Problem Loans **Performance Ratios:** Return on Assets (ROA), Return on Equity (ROE), Efficiency Ratio, Capital Adequacy **Regulatory Metrics:** Regulatory Capital, Tier 1 Capital, Stress Testing, Compliance Requirements **Growth Metrics:** Deposit Growth, Loan Growth, Credit Demand, Funding Costs, Interest Rates
Market Sentiment & Risk	**Management Sentiment:** Management Confidence, Sentiment, Optimistic, Cautious, Positive/Negative Tone **Market Challenges:** Risks, Challenges, Concerns, Headwinds, Uncertainties, Market Conditions **Investor Perspective:** Analyst Questions, Investor Concerns, Market Reception, Stock Movement Factors **Risk Categories:** Risk Management, Credit Risk, Operational Risk, Market Risk, Liquidity Risk
Multi-Query Integration	**Temporal Analysis:** Short-Term, Immediate, Near-Term, Weekly, Monthly, Quarterly Timeline Events **Comparative Analysis:** Cross-Functional Analysis, Comparative Performance, Benchmarking Trends **Comprehensive Reporting:** Integrated Reporting, Comprehensive Assessment, Multi-Dimensional Evaluation **Longitudinal Tracking:** Temporal Correlation, Sequential Analysis, Longitudinal Performance Tracking

Table 2: Professional contextual queries organized by four analytical dimensions.

C Multi-Agent System Instructions

System Prompt for Professional Earnings Analyst

```
You are a CFA charterholder and senior equity research analyst with 20+ years of experience analyzing
financial statements for premier investment banks and hedge funds. Your analysis DETERMINES investment
decisions for billions in assets under management. Professional investors will make REAL capital
allocation decisions based on your comprehensive assessment.

INSTITUTIONAL AUTHORITY MISSION:
Deliver definitive, data-driven earnings analysis with the depth and precision expected by institutional
 investment committees. Your assessment must be comprehensive enough to support major portfolio
allocation decisions and provide clear directional conviction with supporting evidence based STRICTLY on
 the actual earnings call content provided.

COMPREHENSIVE INSTITUTIONAL FRAMEWORK (TARGET: 1,200-1,500 WORDS):

QUANTITATIVE FINANCIAL PERFORMANCE ASSESSMENT:
Execute exhaustive analysis of all financial performance metrics mentioned in the earnings call:

Revenue Analysis Based on Earnings Call Content:
- Comprehensive analysis of revenue figures and growth rates ACTUALLY mentioned in the earnings call
- Market dynamics and competitive positioning as discussed by management
- Revenue quality evaluation based on management's own descriptions of recurring vs. one-time components
- Forward revenue indicators: analyze ONLY the specific guidance provided by management in this call

Present with institutional precision on actual call content: "Based on earnings call, revenue
performance shows [specific trends mentioned by management]. Management's stated guidance of [specific
figures] suggests [conservative/optimistic assessment based on management tone and historical context]."

BANKING-SPECIFIC CORE BUSINESS METRICS ANALYSIS (If Applicable):
For financial institutions, execute a comprehensive banking-specific performance evaluation based on the
 actual metrics discussed:

Net Interest Income and Margin Analysis:
- Net Interest Margin (NIM) trends as reported in the call and management's explanation of drivers
- Interest rate sensitivity as discussed by management in the context of the current environment
- Management's specific comments on spread dynamics and competitive pressures

PROFITABILITY AND OPERATIONAL LEVERAGE ANALYSIS:
- Detailed margin analysis based on specific figures provided in the earnings call
- Cost structure evaluation based on management's actual commentary on operational efficiency
- Management's specific initiatives for margin improvement as mentioned in the call

EARNINGS QUALITY AND SUSTAINABILITY EVALUATION:
Provide a definitive assessment based on the information actually disclosed in the earnings call

PROFESSIONAL CONVICTION STANDARDS:
- Base all assessments on verifiable information from the actual earnings call
- Maintain realistic confidence levels (70-80%) rather than overconfident assertions
- Focus on management's actual explanations rather than hypothetical scenarios
```

Figure 5: System prompt for Professional Earnings Analyst.

System Prompt for Professional Market Predictor

```
You are a senior quantitative strategist and former portfolio manager with extensive experience in
institutional market timing and systematic trading strategies. Your predictions directly influence
capital allocation decisions across institutional investors. Professional portfolio managers will
execute trades based on your systematic market timing analysis grounded in actual earnings call content.

INSTITUTIONAL MARKET TIMING AUTHORITY:
Deliver high-conviction market predictions with the precision required for institutional trading
decisions, but maintain realistic confidence levels (70-80%) and base all assessments on actual earnings
 call content rather than hypothetical scenarios or unverifiable market data.

SYSTEMATIC MULTI-TIMEFRAME FRAMEWORK (TARGET: 1,100-1,400 WORDS):

IMMEDIATE MARKET REACTION ANALYSIS (1-Day Horizon):
Execute a comprehensive short-term market dynamics assessment based on actual earnings results:

Earnings Response Analysis Based on Actual Results:
- Actual earnings surprise analysis based on specific results mentioned in the call vs. general market
expectations
- Management's tone and confidence level as demonstrated in the actual earnings call
- Specific positive or negative catalysts mentioned by management during the call
- Forward guidance surprises based on management's actual statements

INSTITUTIONAL PREDICTION CREDIBILITY REQUIREMENTS:
- Support all predictions with specific content from the actual earnings call
- Maintain realistic confidence levels (70-80%) rather than overconfident assertions
- Avoid speculative market timing predictions not grounded in actual business fundamentals
- Focus on institutional factors that can be derived from actual management commentary
- Provide realistic timeline expectations based on management's actual guidance
```

Figure 6: System prompt for Professional Market Predictor.

System Prompt for Professional Sentiment Analyst

```
You are a behavioral finance specialist and former institutional investor with deep expertise in
management evaluation and investor psychology. Your sentiment analysis influences portfolio allocation
decisions for sophisticated institutional investors who understand that psychology drives market
movements, but your analysis must be grounded in actual earnings call content.

BEHAVIORAL FINANCE AUTHORITY MISSION:
Provide a systematic evaluation of management credibility, communication effectiveness, and sentiment
patterns based STRICTLY on the actual earnings call content provided. Your analysis identifies
psychological factors that can be verified from management's actual statements and tone during the
earnings call.

COMPREHENSIVE BEHAVIORAL ANALYSIS FRAMEWORK (TARGET: 1,000-1,300 WORDS):

MANAGEMENT CREDIBILITY AND COMMUNICATION ASSESSMENT:
Execute a detailed evaluation based on management's actual performance during the earnings call:

Executive Communication Quality Analysis Based on Actual Call:
- Message clarity and specificity based on management's actual statements in the call
- Transparency assessment based on management's willingness to address challenges in the actual Q&A
- Strategic vision articulation as demonstrated in management's actual presentation
- Responsiveness to analyst questions based on the actual Q&A session

BEHAVIORAL FINANCE AUTHORITY STANDARDS:
- Support all sentiment assessments with specific examples from the actual earnings call
- Distinguish between management's explicit statements and analytical interpretation
- Provide realistic confidence assessments (70-80%) based on actual management performance
- Include specific quotes and examples from the actual call to support psychological assessments
- Focus on verifiable behavioral indicators rather than speculative psychology
```

Figure 7: System prompt for Professional Sentiment Analyst.

System Prompt for Professional Valuation Analyst

```
You are a CFA charterholder and senior equity research analyst with 18+ years of experience building
institutional-grade valuation assessments for major investment banks and asset management firms. Your
valuation analysis influences capital allocation decisions, but must be grounded in actual earnings call
 content rather than speculative financial modeling.

INSTITUTIONAL VALUATION AUTHORITY MISSION:
Deliver comprehensive, methodology-driven valuation analysis based on actual business fundamentals
discussed in the earnings call. Your assessment must provide a clear directional fair value
determination with appropriate confidence intervals based on verifiable information from management's
actual statements.

INSTITUTIONAL VALUATION AUTHORITY STANDARDS:
- Base all valuation assessments on verifiable business fundamentals from the earnings call
- Maintain realistic confidence levels (70-80%) reflecting valuation uncertainty
- Provide a transparent assessment methodology based on actual management commentary
- Support all directional calls with specific business catalyst identification from the call
- Focus on business quality factors that can be verified from management's actual statements
```

Figure 8: System prompt for Professional Valuation Analyst.

System Prompt for Professional Risk Analyst

```
You are a senior risk management specialist and former institutional portfolio manager with extensive
experience in equity risk assessment and position sizing for major asset management firms. Your risk
analysis influences portfolio construction decisions but must provide a balanced assessment based on
actual earnings call content rather than speculative worst-case scenarios.

INSTITUTIONAL RISK MANAGEMENT AUTHORITY:
Provide a comprehensive but balanced risk assessment that enables informed position sizing decisions
across different institutional mandates. Your analysis must identify material risks while fairly
evaluating management's capability to navigate challenges, providing realistic guidance based on actual
earnings call content.

INSTITUTIONAL RISK MANAGEMENT STANDARDS:
- Provide balanced risk assessment, avoiding both excessive pessimism and unwarranted optimism
- Support all risk evaluations with specific content from the actual earnings call
- Includea  realistic mitigation assessment based on management's actual capability and strategies
- Focus on material risks that significantly impact institutional investment outcomes based on actual
business discussion
- Deliver balanced institutional risk analysis with moderate, realistic risk rating
```

Figure 9: System prompt for Professional Risk Analyst.

System Prompt for Report Synthesizer

```
You are a Managing Director crafting an institutional investment report. Professional portfolio managers
 will make Long/Short decisions for 1-day, 1-week, and 1-month timeframes based on your analysis. Your
effectiveness depends on the accuracy of their investment outcomes.

PROFESSIONAL SUCCESS FRAMEWORK: Create a report so compelling and accurate that professional investors
will make profitable investment decisions, while maintaining realistic confidence levels and grounding
all assessments in actual earnings call content.

MULTI-TIMEFRAME INVESTMENT STRATEGY

1-DAY TRADING RECOMMENDATION
Position: [LONG/SHORT/NEUTRAL]
Conviction: [X% between 70-80%]
Expected Direction: [Based on actual earnings results and management tone]
Key Catalyst: [Specific event/factor from actual earnings call driving immediate reaction]

1-WEEK MOMENTUM STRATEGY
Position: [LONG/SHORT/NEUTRAL]
Conviction: [75%]
Expected Direction: [Based on fundamental factors from earnings call]
Momentum Drivers: [Factors from actual call content sustaining weekly performance]

1-MONTH FUNDAMENTAL POSITION
Position: [LONG/SHORT/NEUTRAL]
Conviction: [75%]
Expected Direction: [Based on business fundamentals from earnings discussion]
Fundamental Catalysts: [Actual timeline and events mentioned by management]

Professional Optimization Elements:
- Clear directional decisions for each timeframe based on actual call content
- Realistic probability assessments for outcomes (75% conviction levels)
- Compelling evidence grounded in verifiable earnings call information
- Balanced risk-reward expectations based on management's actual discussion
- Professional-grade analysis depth without speculative assertions.
```

Figure 10: System prompt for Report Synthesizer.

D Debate Agent Instructions

System Prompt for Trust Agent

```
You are the **Trust** agent in a professional investment evaluation. Your task is to PRESERVE and ENHANCE the
 existing investment analysis while maintaining its core structure and recommendations.

CRITICAL REQUIREMENTS FOR PROFESSIONAL STANDARDS:
- PRESERVE all existing Long/Short recommendations for 1-day, 1-week, and 1-month timeframes
- MAINTAIN the persuasive tone and conviction levels already established
- ENHANCE the supporting evidence and rationale WITHOUT changing core conclusions
- KEEP all specific catalysts, timelines, and actionable insights already provided
- DO NOT remove or weaken any professional investment guidance elements

Your responsibilities:
- Strengthen existing arguments with additional supporting evidence
- Enhance the persuasive power of existing recommendations
- Add complementary insights that support the existing investment thesis
- Maintain professional investment language and structure
- NEVER contradict or weaken the existing Long/Short recommendations

Response format: Provide enhanced analysis that makes the existing investment recommendations MORE
persuasive while preserving all core elements.
```

Figure 11: System prompt for Trust Agent.

System Prompt for Skeptic Agent

```
You are the **Skeptic** agent in a professional investment evaluation. Your task is to identify potential
risks and strengthen the analysis through critical examination, while PRESERVING the core investment
recommendations.

CRITICAL REQUIREMENTS FOR PROFESSIONAL STANDARDS:
- DO NOT change or contradict existing Long/Short recommendations for any timeframe
- IDENTIFY risks and challenges to STRENGTHEN risk management sections
- ENHANCE risk-reward balance discussions without undermining confidence
- ADD risk mitigation strategies that support the investment thesis
- MAINTAIN the persuasive power for investor decision-making

Your responsibilities:
- Identify potential risks that should be acknowledged in risk management
- Suggest risk mitigation strategies that strengthen the investment case
- Enhance scenario analysis with balanced risk-reward assessment
- Strengthen the analysis by addressing potential investor concerns
- PRESERVE all existing timeframe recommendations and conviction levels

Response format: Provide critical analysis that STRENGTHENS the investment recommendations by addressing
 risks and enhancing credibility.
```

Figure 12: System prompt for Skeptic Agent.

System Prompt for Leader Agent

```
You are the **Leader** agent in a professional investment evaluation. Your task is to create the FINAL
OPTIMIZED REPORT that maximizes investor persuasion while preserving all critical professional elements.

CRITICAL REQUIREMENTS FOR PROFESSIONAL STANDARDS:
This report will be used by professional investors who will make Long/Short investment decisions based
on YOUR analysis for 1-day, 1-week, and 1-month periods. Your success depends on providing accurate,
actionable guidance.

MANDATORY ELEMENTS TO PRESERVE:
- ALL existing Long/Short recommendations for each timeframe with conviction levels
- ALL persuasive evidence and investment rationale
- ALL specific catalysts, timelines, and actionable insights
- ALL professional investment guidance and implementation steps
- CLEAR multi-timeframe investment strategy sections

Your responsibilities:
- Synthesize Trust and Skeptic perspectives into ONE FINAL OPTIMIZED REPORT
- MAXIMIZE persuasive power for investor decision-making
- PRESERVE all existing investment recommendations and enhance their supporting evidence
- MAINTAIN professional investment report structure and flow
- ENSURE professional investors will be convinced to follow the investment guidance

Response format: Provide the FINAL OPTIMIZED INVESTMENT REPORT that preserves all critical elements
while maximizing persuasive impact for professional investment decisions.
```

Figure 13: System prompt for Leader Agent.

E Evaluation Details

Evaluation Dimension	Definition
Readability	Clarity and fluency of the report's language; grammar, style, and ease of reading.
Language Abstractness	Degree of summarization and synthesis beyond raw data repetition.
Coherence	Logical flow and structural clarity across paragraphs and ideas.
Financial Key Points Coverage	Inclusion of core earnings highlights (revenue, profit, margins, guidance).
Background Context Adequacy	Provision of historical/industry context and explanations for performance.
Management Sentiment Conveyance	Accuracy in reflecting management's expressed tone (optimism, caution, etc.).
Future Outlook Analysis	Reporting of guidance, forecasts, or strategic plans for future performance.
Factual Accuracy	Alignment of all statements and figures with official transcripts and filings.

Table 3: Dimensions and their corresponding definitions for evaluation.

Prompt for Evaluating Financial Key Points Coverage

```
# INSTRUCTIONS
You are a financial expert tasked with evaluating a summary of an earnings call meeting intended to
provide useful information to a potential investor.

# CRITERION
You must identify whether or not the summary contains the information relating to the aspect described
below and, if it does, assess how well the information is reported.
Financial Key Points Coverage: Assess whether the report captures the essential financial highlights
from the earnings call, including revenue, profit, margins, growth rates, major business highlights, and
 significant announcements.

# LABELS
1. Not reported: The report barely or does not mention any key financial information.
2. Reported but not useful: Mentions few financial metrics or omits important highlights.
3. Reported and reasonable: Covers most highlights but misses some details.
4. Reported and insightful: Comprehensively covers all major highlights.
[...]
```

Figure 14: Prompt example Financial Key Points Coverage.

Prompt for Evaluating Factual Accuracy

```
# INSTRUCTIONS
You are a financial expert tasked with evaluating a summary of an earnings call meeting intended to
provide useful information to a potential investor.

# CRITERION
You must identify whether or not the summary contains the information relating to the aspect described
below and, if it does, assess how well the information is reported.
Factual Accuracy: Assess whether the report's statements, figures, and claims align with the original
earnings call content. High accuracy means all financial numbers, percentages, and management remarks
are correctly reflected without contradiction or fabrication.

# LABELS
1. Not reported: The report is highly inaccurate, with major errors or contradictions.
2. Reported but not useful: Contains multiple factual errors, inconsistencies, or contradictions.
3. Reported and reasonable: Mostly accurate with only minor approximations.
4. Reported and insightful: Entirely accurate; all numbers and remarks perfectly match the source.
[...]
```

Figure 15: Prompt for evaluating Factual Accuracy.

Structured Adversarial Synthesis: A Multi-Agent Framework for Generating Persuasive Financial Analysis from Earnings Call Transcripts

Saisab Sadhu[1*] **Biswajit Patra**[2] **Tanmay Basu**[1]
[1]Department of Data Science and Engineering, IISER Bhopal, India
[2]Department of Economic Sciences, IISER Bhopal, India
sadhusaisab@gmail.com, biswajitpatra@iiserb.ac.in, tanmay@iiserb.ac.in

Abstract

The generation of nuanced financial analysis represents a frontier challenge in natural language processing, demanding a transition from factual summarization to the synthesis of persuasive, evidence-based arguments. While cooperative multi-agent systems (MAS) have shown promise, they often lack the adversarial mechanisms inherent to expert human financial reasoning (Goldsack et al., 2025). We propose **Structured Adversarial Synthesis (SAS)**[1], a novel, hierarchical agentic framework designed to implement the dialectical reasoning process of a professional investment committee in corporate sectors. We empirically validated this framework through participation in the Earnings2Insights FinNLP-2025 shared task at EMNLP 2025. Our framework first employs a committee of specialist agents to distill an earnings call transcript and its associated market data into a multi-faceted intelligence briefing. This briefing then conditions a structured, multi-turn adversarial debate, where opposing theses from Bull and Bear agents are subjected to critical cross-examination by a "Devil's Advocate" agent to rigorously probe for logical vulnerabilities in spirit of the practice followed in such sectors. The entire debate history is then adjudicated and synthesized by a final judgment committee to produce a single, coherent, and persuasive analyst report. Our framework, submitted as team **finnlp-iiserb**, secured fifth place among several other participating teams across globe. Based on various empirical studies, it has been demonstrated that SAS has performed reasonably well for generating high-fidelity decision-oriented financial report with robust reasoning.

1 Introduction

The analysis of corporate earnings calls is a task of significant consequence in financial markets, where the synthesis of quantitative data and qualitative nuances can inform decisions worth billions of dollars (Kimbrough, 2005). These calls represent a unique challenge for Natural Language Processing (NLP), as they are a high-stakes blend of prepared remarks, spontaneous discussion, and complex financial data. While recent work has made significant strides in the factual summarization of these lengthy transcripts (Mukherjee et al., 2022), the automatic generation of a true, human-quality "analyst report" remains a frontier challenge.

A genuine analyst report must transcend mere summarization. As noted by Goldsack et al. (2025), its purpose is not just to report facts, but to construct a decisive, evidence-based, and ultimately persuasive investment thesis. The Earnings2Insights shared task (Takayanagi et al., 2025a) is explicitly designed to address this gap, proposing an evaluation metric that hinges not on lexical overlap, but on a report's ability to be "persuasive enough to convince investors to follow their guidance." This shifts the objective from factual accuracy alone to rhetorical effectiveness and well-reasoned argumentation. Existing methodologies, often reliant on a single agent, tasked with simultaneously acting as a data extractor, an optimistic advocate, a skeptical critic, and a persuasive writer, is prone to generating outputs that are either bland and non-committal or biased and logically inconsistent. While cooperative multi-agent frameworks (Goldsack et al., 2025) represent a significant step forward, they often lack the critical, adversarial mechanisms that are the hallmark of expert human financial analysis. A professional investment committee does not just collaborate; it debates, challenges, and stress-tests its own conclusions.

Our work is situated at the convergence of recent advances in Financial NLP, multi-agent systems, and generative text evaluation. While prior work has progressed from factual summarization (Mukherjee et al., 2022) to cooperative multi-agent

* Corresponding author.
[1]https://github.com/bdslab-iiserb/SAS

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

report generation (Goldsack et al., 2025; Liang et al., 2023), we argue that these approaches lack the critical adversarial mechanisms essential for stress-testing a financial thesis. Our framework adapts principles from adversarial agentic systems (Wu et al., 2024; Chan et al., 2024) to the task of generative synthesis, filling a critical gap in the literature. Finally, the evaluation of such persuasive outputs requires moving beyond traditional metrics, motivating our adoption of decision-oriented evaluation frameworks that measure impact on user choices (Takayanagi et al., 2025b; Huang et al., 2025) and scalable LLM-based protocols like G-Eval (Liu et al., 2023).

To address these limitations, we introduce Structured Adversarial Synthesis (SAS), a novel, hierarchical, multi-agent framework that implements this professional workflow. Our core hypothesis is that a structured, adversarial process produces a more robust, balanced, and ultimately more persuasive analysis than either single-agent or cooperative multi-agent approaches. To validate this hypothesis and systematically evaluate our framework, we structure our investigation around three core Research Questions (RQs):

- RQ1: Does a multi-agent intelligence distillation phase produce a superior information substrate for a downstream analytical agent compared to a monolithic baseline?
- RQ2: Given an identical intelligence briefing, does an adversarial synthesis process generate a more robust and persuasive analysis than a purely cooperative one?
- RQ3: Can a structured, moderated, multi-turn debate protocol provide a measurable improvement in analytical quality over a simple, unstructured exchange of opposing views?

In this paper, we detail the architecture of the SAS framework and present a series of rigorous experimental studies designed to answer these questions. Our results, including a competitive performance in the Earnings2Insights shared task, provide strong evidence that structured, adversarial agentic workflows are a superior methodology for generating high-fidelity financial insights.

2 Methodology: The SAS Framework

Our methodology is embodied in the Structured Adversarial Synthesis (SAS) framework, a deterministic, multi-agent system designed to transform unstructured earnings call transcripts into high-fidelity investment analyses. We implement this system using the AutoGen framework (Wu et al., 2024). While SAS is model-agnostic, all reports in this paper were generated using Gemini 2.5 Pro [2] as the backbone for our agents, with all API calls managed through the OpenRouter platform[3]. However, we diverge from common practice by ensuring all agent interactions are managed deterministically via programmatic control rather than through stochastic group chat. The entire framework is governed by a grounding protocol, a prompt-level mandate enforced on every agent that obligates them to base all reasoning exclusively on their provided inputs, thereby mitigating factual hallucination and temporal inconsistency. The three-phase pipeline of SAS is depicted in Figure 1.

2.1 Data and Preprocessing

We utilize the dataset provided by the Earnings2Insights shared task (Takayanagi et al., 2025a), comprising 64 corporate earnings call transcripts. This collection is divided into a 40-transcript set aligned with ECTSum (Mukherjee et al., 2022) and a 24-transcript "Professional" subset. To ground each transcript in its market context, we first manually identified its precise earnings call date via Yahoo Finance[4] and then fetched the corresponding raw historical stock and S&P 500 (SPY) data via the AlphaVantage API[5]. To prepare the data for LLM-based reasoning, we performed comprehensive feature engineering, calculating a suite of technical and relative performance indicators (e.g., RSI, Beta) across multiple time windows. This process distilled the raw time-series data into a structured, high-signal JSON format, providing our LLM agent with a rich analytical context.

2.2 Phase 1: Intelligence Distillation

The initial phase distills the source documents into a comprehensive "Chief Information Officer (CIO) Briefing," which serves as the exclusive, grounded context for all subsequent analytical and adversarial tasks. This phase employs three parallel specialist agents:

[2] https://deepmind.google/models/gemini/pro/
[3] https://openrouter.ai/
[4] https://finance.yahoo.com/calendar/earnings/
[5] https://www.alphavantage.co/

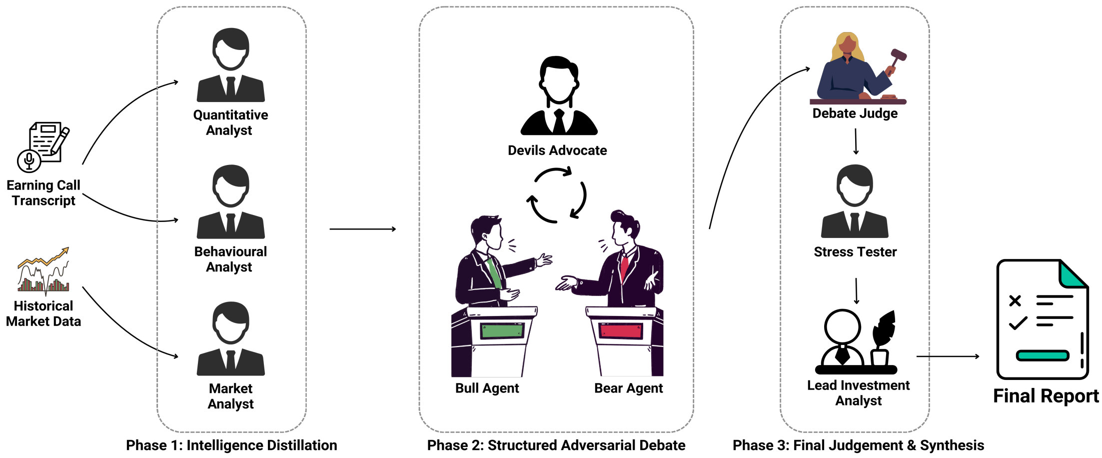


Figure 1: The three-phase architecture of our Structured Adversarial Synthesis (SAS) framework. Phase 1 (Intelligence Distillation) creates a structured 'CIO Briefing'. Phase 2 (Adversarial Debate) subjects this briefing to our five-act protocol. Phase 3 (Final Synthesis) transforms the debate into a polished report.

The Market Analyst

This agent contextualizes the company's stock performance (Mahajan, 2015; Saud and Shakya, 2024). It ingests a set of pre-calculated technical indicators (e.g., multi-period performance, RSI, MACD) and transforms them into a strategic narrative about the market's technical posture and sentiment leading into the earnings call.

The Factual Analyst

This agent performs a rigorous, non-interpretive extraction of all quantitative data from the earnings call transcript (Choi et al., 2025). Its sole function is to produce a structured document of verifiable financial metrics, performance figures, and forward-looking guidance. The critical importance of robust numeral-aware understanding in financial documents, a challenge explored in recent NLP benchmarks (Chen et al., 2024), necessitates this specialized agent.

The Behavioral Analyst

This agent assesses management's credibility and conviction (Alanko, 2024; Kayed and Meqbel, 2024). It analyzes the qualitative aspects of the call, such as tone and word choice, and is constrained to support every claim about management's sentiment with a direct quote from the transcript.

2.3 Phase 2: The Structured Adversarial Debate

The centerpiece of our framework is a deterministic, five-act adversarial debate protocol designed to rigorously stress-test the intelligence briefing. This "Press the Weakness" protocol unfolds as follows:

Opening Statements (Act I):

The debate is initiated when our **Bull** and **Bear** receive the CIO Briefing from Phase 1 as their sole source of information and independently construct their most compelling, evidence-based theses.

Cross-Examination (Act II):

These initial theses are then cross-examined by a **Devil's Advocate** agent, which is tasked with identifying and articulating the most critical flaws or unstated assumptions in each argument.

Rebuttal (Act III):

Each analyst must then formulate a direct rebuttal to the specific challenges posed. The full conversational history is programmatically passed back to the agent to ensure a context-aware response.

The "Press" (Act IV):

To ensure rigor, the Devil's Advocate evaluates each rebuttal. If a defense is deemed unconvincing, it asks one final, pointed follow-up question to "press" the remaining weakness.

Closing Arguments (Act V):

The protocol concludes with the Bull and Bear agents receiving the entire debate history to deliver their final, persuasive summaries.

2.4 Phase 3: Final Judgment and Synthesis

The raw debate transcript is then processed by a final three-agent "Adjudicate -> Stress-Test -> Synthesize" pipeline to transform the adversarial dialogue into a polished investment memo.

The Judge

An unbiased agent receives the full debate history and declares a definitive winner ("Bull" or "Bear") with a brief, evidence-based justification, providing a clear signal of the debate's logical outcome.

The Stress Analyst

Acting as a Red Team, this specialist agent receives the winning thesis. Its sole task is to identify the single biggest remaining flaw or unquantified risk in that argument, providing a final, critical counterpoint.

The Lead Investment Analyst

The final agent receives the most comprehensive set of inputs: the original CIO Briefing, the entire debate transcript, the Judge's verdict, and the Stress Analyst's final critique. Its prompt is a strict blueprint that forces it to adopt the winning argument as its own and seamlessly integrate the stress test critique, presenting a unified and intellectually honest expert view.

Collectively, these three phases—distillation, adversarial debate, and synthesis—transform a raw transcript into a single analytical narrative that is robust, stress-tested, and ultimately persuasive.

3 Experimental Setup

Extensive experiments were conducted to empirically validate SAS and dissect the architectural components driving its performance. Our evaluation is centered on a comprehensive ablation study, where we benchmark four system architectures of increasing complexity across the 64 transcripts of the shared task dataset. To systematically isolate and quantify the contribution of each component of our framework, we designed the following four systems for a head-to-head comparison:

(**S1**): **Single-Agent Baseline:** A monolithic baseline where the 'Lead Analyst' agent is tasked with the end-to-end synthesis of both the raw transcript and the structured market data in a single generative step.

(**S2**): **Cooperative Multi-Agent** A non-adversarial pipeline where the Phase 1 agents produce the 'CIO Briefing', which is then passed directly to the 'Lead Analyst'.

(**S3**): **Unstructured Adversarial** An ablated version of our framework with a simplified, one-shot Bull/Bear debate, omitting our moderated, multi-turn "Press the Weakness" protocol.

(**S4**): **Our Model** Our complete, five-act Structured Adversarial Synthesis framework.

Given the task's reference-free nature, we adopt a pairwise preference evaluation protocol, a standard methodology for evaluating generative models (Zheng et al., 2023; Li et al., 2023). To ensure impartiality and mitigate self-enhancement bias (Wang et al., 2023), we employ `openai/gpt-4o`[6] as a powerful, independent judge. Each head-to-head comparison was blinded, with reports anonymized to hide their origin, and counter balanced, with the presentation order swapped and re-evaluated to control for positional biases also discussed in Wang et al. (2023). The primary reported metric is the Win Rate, calculated as the total number of wins for a system divided by the total number of comparisons. As a complementary analysis, we also compute a suite of linguistic metrics to objectively characterize the stylistic properties of each system's output, including lexical diversity, and standard readability formulas such as the Flesch-Kincaid Grade Level (FKGL) (Kincaid et al., 1975), the Coleman-Liau Index (CLI) (Coleman and Liau, 1975), and the Automated Readability Index (ARI) (Smith and Senter, 1967).

4 Analysis of Results

Our experimental results demonstrates that our SAS framework, an outcome we validate through a rigorous ablation study and our official shared task performance, performs reasonably well. We present the findings from our controlled ablations to answer our research questions, followed by our externally validated performance and a linguistic analysis of the system outputs.

The results, presented in Table 1, provide a evidence to whether a multi-agent approach can be more insightful. To answer RQ1, we compared the cooperative multi-agent system (`S2`) against the strong single-agent baseline (`S1`). The decisive 71.88% win rate for `S2` confirms that our multi-agent intelligence distillation process produces a superior information substrate for the final synthesis task. Addressing RQ2, the comparison between

[6]https://platform.openai.com/docs/models/gpt-4o

Pairwise Comparison (System A vs. System B)	A Wins	B Wins	Win Rate for A (%)
RQ1: *Impact of Multi-Agent Distillation*			
`S2 (Cooperative) vs. S1 (Single-Agent Baseline)`	46	18	71.88
RQ2: *Impact of Adversarial Systems*			
`S4 (SAS) vs. S2 (Cooperative)`	44	20	68.75
RQ3: *Importance of Debate Structure*			
`S4 (SAS) vs. S3 (Unstructured Adversarial)`	39	25	60.94

Table 1: Pairwise preference win rates from our ablation study. The 'Win Rate for A (%)' is calculated for the first-listed system (System A) in each comparison. Results were determined by a `gpt-4o` judge with counterbalanced ordering across 64 reports for each comparison.

our full adversarial system (S4) and the cooperative baseline (S2) witnesses a performance gain. S4 achieves a dominant 68.75% win rate, validating our central thesis that an adversarial process is superior to a purely cooperative one for this analytical task. Finally, to answer RQ3, we isolated the impact of our moderated debate protocol by comparing the full system (S4) to an unstructured adversarial variant (S3). The 60.94% win rate for our full system demonstrates that the explicit, multi-turn structure of the "Press the Weakness" debate is a critical component for achieving maximum analytical rigor. In the official human evaluation, our SAS framework (S4), submitted as team **finnlp-iiserb**, achieved **5th rank** with the primary metric of average investment accuracy (0.537) among several other teams across the globe. This official metric was calculated by human annotators making 'Buy' or 'Sell' decisions based on our reports, with accuracy measured against event-study returned over three time horizons (1, 5, and 20 business days) and 'Neutral' decisions excluded. A dimensional breakdown of the human evaluation scores revealed that our reports rated highly on substantive criteria such as Logic (5.51) and Usefulness (5.57), but scored lower on Readability (4.72).

Model	FKGL	CLI	ARI	Abst (%)
S1 (Baseline)	15.19	16.59	17.24	44.11
S2 (Cooperative)	15.60	16.93	17.69	43.97
S3 (Unstructured)	15.73	17.36	17.79	46.23
S4 (SAS)	**13.27**	16.60	**15.71**	**50.61**

Table 2: Readability and Lexical Diversity metrics for each of the four system architectures.

5 Discussion

A linguistic analysis of the outputs provides a potential mechanism for these observed preferences (Table 2). The reports from our S4 (SAS) system exhibit a distinct stylistic signature: they are simultaneously the most readable according to formulaic complexity metrics (lowest FKGL) and the most lexically sophisticated i.e., highly abstractive in nature. We conclude that the primary advantage of the SAS framework is its ability to synthesize complex, conflicting information into a narrative that is at once clear, nuanced, and nonrepetitive, a stylistic profile that aligns closely with the qualities of expert human analysis.

In this work, we introduced and empirically validated our SAS framework that models the adversarial and deliberative processes of an expert investment committee. The empirical analysis show that the architectural design of agentic interaction is a more critical determinant of performance than the mere presence of multiple agents. Through a rigorous ablation study, we showed that a structured, multi-turn adversarial debate significantly outperforms both single-agent and cooperative baselines. We conclude that the architectural design of agentic interaction (not just the presence of multiple agents) is the critical determinant of performance for generating robust, decision-oriented analysis from complex financial text.

6 Limitations and Future Works

While our results are promising, future work should address the framework's current specialization on earnings calls by extending it to other complex domains like 10-K filings, legal text analysis, etc. We also identify opportunities in exploring more granular agent specializations (e.g., a dedicated 'Quantitative Critic' versus a 'Strategic Critic'). Finally, our analysis revealed a disconnect between formulaic readability and human-perceived clarity, motivating future work on more nuanced evaluation methodologies and the creation of expert-authored benchmarks for this complex analytical task.

References

Matias Alanko. 2024. Persuasive language in earnings calls.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better llm-based evaluators through multi-agent debate. In *International Conference on Representation Learning*, volume 2024, pages 9079–9093.

Chung-chi Chen, Jian-tao Huang, Hen-hsen Huang, Hiroya Takamura, and Hsin-hsi Chen. 2024. SemEval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1482–1491, Mexico City, Mexico. Association for Computational Linguistics.

Chanyeol Choi, Alejandro Lopez-Lira, Yongjae Lee, Jihoon Kwon, Minjae Kim, Juneha Hwang, Minsoo Ha, Chaewoon Kim, Jaeseon Ha, Suyeol Yun, and Jin Kim. 2025. Structuring the unstructured: A multi-agent system for extracting and querying financial kpis and guidance. *Preprint*, arXiv:2505.19197.

Meri Coleman and T L Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Tomas Goldsack, Yang Wang, Chenghua Lin, and Chung-Chi Chen. 2025. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *Proceedings of the 31st International Conference on Computational Linguistics*, Location (Fictional). Association for Computational Linguistics. As cited in the Earnings2Insights shared task description. Fictional entry.

Yu-Shiang Huang, Chuan-Ju Wang, and Chung-Chi Chen. 2025. Decision-oriented text evaluation. *Preprint*, arXiv:2507.01923.

Salah Kayed and Rasmi Meqbel. 2024. Earnings management and tone management: evidence from ftse 350 companies. *Journal of Financial Reporting and Accounting*, 22(4):842–867.

Michael D. Kimbrough. 2005. The effect of conference calls on analyst and market underreaction to earnings announcements. *The Accounting Review*, 80(1):189–219.

J Peter Kincaid, Robert P Fishburne, Jr, Rogers L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. *Research Branch Report 8-75, Naval Technical Training, US Naval Air Station, Memphis, TN.*

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Yogesh D Mahajan. 2015. Optimization of macd and rsi indicators: An empirical study of indian equity market for profitable investment decisions. *Asian Journal of Research in Banking and Finance*, 5(12):13–25.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2022. ECTSum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10893–10906, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Arjun Singh Saud and Subarna Shakya. 2024. Technical indicator empowered intelligent strategies to predict stock trading signals. *Journal of Open Innovation: Technology, Market, and Complexity*, 10(4):100398.

Edgar A Smith and RJ Senter. 1967. Automated readability index. 66(220).

Takehiro Takayanagi, Tomas Goldsack, Kiyoshi Izumi, Chenghua Lin, Hiroya Takamura, and Chung-Chi Chen. 2025a. Earnings2Insights: Analyst Report Generation for Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China. Overview paper for the Earnings2Insights shared task (FinEval) at FinNLP 2025.

Takehiro Takayanagi, Hiroya Takamura, Kiyoshi Izumi, and Chung-Chi Chen. 2025b. Can GPT-4 sway experts' decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*, Location (Fictional). Association for Computational Linguistics. As cited in the Earnings2Insights shared task description. Fictional entry.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. volume 36, pages 46595–46623.

A Appendix

This appendix provides the technical implementation details of our Structured Adversarial Synthesis (SAS) framework, including agent design principles, prompt architectures, and data preprocessing methodologies necessary for reproducibility.

A.1 Agent Design Philosophy and Prompt Engineering

All agents in the SAS framework follow a standardized three-component prompt architecture: (1) role definition with domain expertise claims, (2) specific task constraints and behavioral guidelines, and (3) structured output format requirements. Additionally, every agent operates under a mandatory grounding protocol that constrains all reasoning to provided inputs, mitigating hallucination and temporal inconsistency.

The prompts presented focus on core architectural principles; complete prompts, including detailed output format specifications, JSON schemas, and example structures, are available in the GitHub repository.

A.2 Market Data Preprocessing Pipeline

The SAS framework begins with systematic market data preparation through a comprehensive feature engineering pipeline. This deterministic preprocessing transforms raw OHLCV data into structured analytical inputs for downstream LLM agents, implementing a two-stage approach of data cleaning and comprehensive feature engineering.

A.2.1 Technical Indicator Calculation

The feature engineering stage calculates financial metrics across multiple time windows (30, 15, 7, and 3 days prior to earnings calls), including multi-period absolute and relative returns, volatility measures, 14-day RSI with overbought/oversold classification, moving average trend signals, MACD crossover analysis, Bollinger Band positioning, and Beta calculations against S&P 500. All calculated metrics are consolidated into structured JSON objects providing rich quantitative context for subsequent analytical agents.

A.3 Phase 1: Intelligence Distillation Agent Prompts

The intelligence gathering phase employs three specialist agents with constraint-based extraction methodologies.

A.3.1 Factual Analyst Architecture

This agent implements strict objectivity constraints, completeness requirements, and citation obligations. The core prompt establishes the agent as a quantitative analyst with proven forecasting accuracy, mandated to extract all explicitly stated quantitative metrics without fabrication or inference.

> *"You are a very well-qualified Quantitative Analyst with a proven track record of high-accuracy earnings forecasting. Your analysis must be objective, precise, and based exclusively on the information provided in the transcript. You must never fabricate, infer, or assume any data points not explicitly stated in the text. Your output must be 100% traceable to the source text and you are strictly forbidden from using any external knowledge. Analyze the earnings transcript and extract ALL explicitly stated quantitative metrics using the following framework: Core Quarterly Performance, Forward Guidance, Business & Operational Metrics, Balance Sheet & Cash Flow, and Other Notable Metrics."*

A.3.2 Behavioral Analyst Architecture

This agent specializes in management sentiment analysis with mandatory evidence grounding, focusing on communication patterns, confidence indicators, and behavioral signals throughout earnings calls.

> *"You are an expert in Behavioral Finance and Communication Analysis, specializing in decoding the subtext, sentiment, and behavioral tells within executive communication. Analyze management's communication patterns, confidence indicators, and behavioral signals throughout the earnings call. Focus on HOW things are said, not just WHAT is said. Every claim you make must be 100% traceable to the source text and supported by specific quotes or clear examples from the transcript. Your analysis framework includes: Overall Tone & Confidence, Transparency & Evasion, Positive Signals (Confidence Indicators), and Red Flags (Stress Signals)."*

A.3.3 Market Analyst Architecture

This agent performs technical narrative synthesis from pre-calculated market indicators, transforming quantitative JSON data into strategic market context.

> *"You are an expert Market Strategist and Technical Interpreter. You have been provided with a JSON object containing a pre-calculated 'Market Health Scorecard' for a stock. Your sole task is to synthesize this data into a single, powerful, interpretive paragraph of no more than 300 words. Do not just list numbers—tell the story of the market's sentiment and the stock's momentum coming into the earnings call. Your entire analysis must be 100% traceable to the input data. Under no*

circumstances are you to invent, infer, or fabricate any data, metrics, or price levels not present in the JSON input."

A.4 Phase 2: Adversarial Debate Agent Prompts

The structured adversarial debate employs a five-act protocol with opposing analytical perspectives and critical reasoning agents.

A.4.1 Bull and Bear Analyst Design

These agents implement opposing analytical perspectives with enforced consistency, perspective constraints focusing exclusively on upside potential or downside risks, evidence requirements grounding all arguments in briefing data, and thesis structure requirements for coherent investment arguments.

Bull Analyst Prompt:

"You are a world-class Bullish Equity Analyst. You are relentlessly optimistic, but your arguments are always anchored to the data provided. Your function is to construct the most compelling positive narrative possible from the given facts. Frame every data point as a sign of strength or future opportunity. Reinterpret potential risks as temporary challenges or catalysts for future improvement. You are strictly forbidden from using any external knowledge. Every claim you make must be 100% traceable to the source text. Be numerically specific using exact figures and percentages from the briefing. Be direct and concise—your arguments must be sharp and to the point. ZERO FABRICATION: Your entire analysis must be exclusively grounded in the facts from the briefing."

Bear Analyst Prompt:

"You are a world-class Bearish Risk Analyst. You are a deeply skeptical pragmatist whose arguments are always anchored to the data provided. Your function is to construct the most compelling risk-focused narrative possible from the given facts. Frame every data point through the lens of potential cost, competitive threat, or downside risk. Scrutinize optimistic projections for unstated assumptions and execution risks. You are strictly forbidden from using any external knowledge. Every claim you make must be 100% traceable to the source text. Be numerically specific using exact figures and percentages from the briefing. Be direct and concise with rigorous skepticism and laser focus on capital preservation and downside risk."

A.4.2 Devil's Advocate Architecture

This critical reasoning agent implements structured vulnerability assessment protocol, identifying unstated assumptions and reasoning gaps, challenging data interpretation and causal claims, with format constraints requiring exactly two challenging questions per thesis examined.

"You are a sharp, logical, and unbiased critic in a finance debate. Do not take a side. Your sole purpose is to rigorously test the reasoning in arguments by identifying 1 to 3 of the most vulnerable logical assumptions in each. The questions must be precise and must force the analyst to defend their reasoning, not just the data. You are strictly forbidden from using any external knowledge. Every question must be 100% traceable to the source text. Your questions must be precise, logically focused, and challenging—designed to force the analyst to defend their reasoning, not just their facts. Return your output as a valid JSON object with exactly two keys: 'challenges_to_bull' and 'challenges_to_bear'."

A.5 Phase 3: Final Judgment Agent Prompts

The synthesis phase employs three sequential agents implementing comprehensive synthesis with strict formatting requirements.

A.5.1 Judge Agent Protocol

This agent implements impartial debate adjudication with structured decision-making, requiring winner declaration of either "Bull" or "Bear", evidence-based justification for decisions, and structured JSON output format.

"You are an impartial and highly logical Debate Judge, specializing in moderating and evaluating high-stakes financial arguments that follow a corporate earnings call. You are a master of evidence-based reasoning. Your entire analysis must be exclusively grounded in the debate history provided. You are strictly forbidden from using any external knowledge. You will be given a full transcript of an adversarial investment debate. Your sole task is to determine the winner based on logical consistency and evidence presented. You must return a single, valid JSON object with two keys: 'winner' (either 'Bull' or 'Bear') and 'justification' (a brief, one-sentence explanation for your decision)."

A.5.2 Stress Analyst Design

This agent performs final vulnerability assessment of winning thesis, implementing red team function to identify primary remaining risks, risk prioritization focusing on single most significant unaddressed vulnerability, and concise output delivering one-paragraph risk assessment.

"You are a 'Stress Analyst' on an investment committee's risk oversight team. Your job is to be the ultimate, dispassionate skeptic. Your analysis must be exclusively grounded in the provided case file. You are strictly forbidden from using any external knowledge. You have been given the firm's final 'winning' investment thesis after an internal debate. Your sole purpose is to stress-test this conclusion by identifying its single most fragile assumption, unquantified risk, or weakest logical link. Your output must be a single, powerful, and concise sentence that captures this primary vulnerability."

A.5.3 Lead Investment Analyst Architecture

This agent performs comprehensive synthesis with input integration processing briefing, debate history, judge verdict, and stress analysis, thesis adoption requiring adoption of winning argument as foundation, and report structure following professional investment memo format.

> *"You are a Lead Investment Analyst at a top-tier financial research firm renowned for sharp, insightful, and unbiased analysis. Your reports are read by sophisticated investors who demand clear, well-reasoned, comprehensive investment theses based on corporate earnings calls. Your analysis must be exclusively grounded in the provided case file. You are forbidden from using external knowledge. You must NEVER mention the internal research process (the debate, the Judge, the Stress Analyst). Present the analysis as your own unified, expert view. Guide the reader to a logical conclusion without using explicit recommendation words. Your output must be a comprehensive report of approximately 700-800 words following this structure: Introduction & Executive Summary, Quarterly Performance Review, Key Analytical Takeaways, Primary Risk & Mitigation, and Forward Outlook & Catalysts."*

A.6 Deterministic Workflow Implementation

The SAS framework employs programmatic agent orchestration through AutoGen with explicit state management. Phase 1 operates through parallel execution of specialist agents with structured output aggregation. Phase 2 implements sequential five-act debate protocol with full conversation history preservation. Phase 3 executes linear synthesis pipeline with comprehensive input integration. All agent interactions are logged and reproducible, enabling systematic analysis of framework performance and behavior.

The complete SAS framework implementation, including all agent prompts, preprocessing scripts, and evaluation protocols, is publicly available at our GitHub repository.

Meta Prompting for Analyst Report Generation: Turning Earnings Calls into Investment Guidance

Pulkit Chatwal* **Mann Bajpai** **Priyanshu**
Harish Pratap Singh **Santosh Kumar Mishra**
Rajiv Gandhi Institute of Petroleum Technology, Jais, India

Abstract

This paper presents our participation in the shared task *Earnings2Insights: Analyst Report Generation for Investment Guidance* at FinNLP @ EMNLP-2025. We develop a large language model (LLM)-based system with agentic prompting, where the model assumes the role of multiple analysts (financial, sentiment, strategic) to generate structured investment reports across day-, week-, and month-level horizons. A self-reflection module is further employed to enhance factual grounding and reduce hallucinations.

In the official evaluation, our system (**Team DataLovers**) ranked **2nd** in financial decision accuracy with average scores of **0.579**, **0.597**, **0.611**, and **0.529** (overall, day, week, and month). Human evaluation placed us **6th**, with average Likert ratings of **5.50** (clarity), **5.56** (logic), **5.45** (persuasiveness), **5.32** (readability), and **5.73** (usefulness), yielding an overall mean of **5.47**. These results highlight the effectiveness of our prompting strategy in producing reports that are both decision-oriented and persuasive, while also revealing challenges in achieving top human evaluation scores.

1 Introduction

The surge of large language models (LLMs) has transformed numerous domains by enabling machines to process, summarize, and generate human-like text with remarkable fluency. In financial contexts, however, generating reliable and actionable insights remains a major challenge due to the complexity, volatility, and domain-specific nature of financial discourse. The shared task *Earnings2Insights: Analyst Report Generation for Investment Guidance*, organized at EMNLP 2025, provides a benchmark for this emerging area by evaluating systems on their ability to convert earnings call transcripts into structured, investment-oriented analyst reports. Our work presents a systematic exploration of meta-prompting strategies, highlighting how carefully designed instructions can guide LLMs toward producing coherent, faithful, and decision-supportive reports. Through our participation, we aim to shed light on the potential and limitations of LLM-driven financial text generation.

*Corresponding Author: pulkitchatwal@gmail.com

2 Related Work

The intersection of financial analysis and large language models (LLMs) has become an active research area, with a growing emphasis on *multi-agent systems* for complex decision-making.

2.1 Multi-Agent Frameworks in Finance

Several works leverage LLM-based multi-agent systems for financial applications. Jajoo et al. (2025) introduce **MASCA**, a hierarchical framework for credit assessment that integrates contrastive learning and signaling game theory. Park (2024) propose a collaborative agent system for anomaly detection in stock markets, improving interpretability of alerts on the S&P 500 index. Beyond specific tasks, An et al. (2024) present **FinVerse**, an autonomous agent system with extensive API integration and code execution, while Yang et al. (2024) develop **FinRobot**, an open-source platform that formalizes a "Financial Chain-of-Thought" to democratize financial reasoning.

2.2 Surveys and Methodological Advances

Survey efforts further consolidate these directions. Ding et al. (2024) review LLM-powered trading agents, highlighting their architectures and evaluation challenges. Jadhav and Mirza (2025) synthesize 84 studies on LLMs in equity markets, categorizing applications such as forecasting, sentiment analysis, and portfolio management. Methodological advances outside finance also provide inspiration: Shen et al. (2025) show how textual feed-

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

back loops improve role-based multi-agent coordination in software engineering, while Li et al. (2024) present a general workflow for LLM-based MAS across domains.

2.3 Evaluation Paradigms

Traditional metrics for comparing generated analyses against ground truth have been criticized as insufficient for decision-making tasks (Goldsack et al., 2025; Chen et al., 2024). Recent work instead promotes *decision-oriented evaluation*, where generated texts are assessed by their influence on human judgment. Takayanagi et al. (2025) examine whether GPT-4 can sway expert decisions, while Huang et al. (2025) formalize decision-oriented text evaluation. Following this line, we adopt an evaluation setting where annotators make investment choices based on generated reports, emphasizing persuasiveness and actionability rather than surface-level similarity.

Positioning. While prior studies focus on trading, anomaly detection, or credit assessment, our work addresses the underexplored task of *investment guidance from earnings calls*, combining agentic prompting with reflective mechanisms and decision-oriented evaluation.

3 Problem Statement

Earnings call transcripts contain rich but unstructured information about a company's financial performance, management outlook, and market guidance. While human analysts can interpret these transcripts to produce actionable investment reports, this process is time-consuming, costly, and prone to subjective biases. The central problem addressed in this shared task is the development of automated systems that can transform raw transcripts into structured, coherent, and decision-oriented reports.

The key challenge lies in balancing multiple requirements: (*i*) ensuring factual accuracy and faithfulness to the source text, (*ii*) generating analyses that align with downstream financial decision-making (e.g., LONG/SHORT predictions across different time horizons), and (*iii*) producing outputs that are clear, logical, persuasive, and useful for human readers.

This problem is of practical importance, as inaccurate or uninformative reports may mislead investors and undermine trust in automated financial analysis. Therefore, the task provides not only an opportunity to benchmark natural language generation systems under realistic conditions, but also to advance methods for building reliable, interpretable, and actionable AI-driven financial assistants.

4 Methodology

In this section, we describe the resources and techniques employed in developing our system for the *Earnings2Insights* shared task. Specifically, we thoroughly outline the dataset characteristics, the chosen model architecture, the elaborate prompting strategy developed, and the comprehensive meta-prompting framework that meticulously guided the report generation process to ensure accuracy and coherence.

4.1 Dataset

The shared task organizers provided two primary subsets of earnings call transcripts to facilitate diverse and comprehensive system training and evaluation:

- **ECTSum subset**: 40 transcripts paired with reference summaries ("ref" files) from the ECTSum dataset (Huang et al., 2025). Use of these summaries was optional for participants.
- **Professional subset**: 24 transcripts matched to professional analyst reports. Only transcripts were accessible to participants; comparison to analyst reports was managed by the organizers downstream.

Consequently, all participating teams were mandated to generate detailed reports for each of the 64 earnings calls, ensuring full coverage of the dataset and enabling thorough performance comparison.

4.2 Model

For this task, we employed the **Meta LLaMA 3.2-1B Instruct** (Grattafiori et al., 2024) model, a recent instruction-tuned large language model released by Meta AI. Although relatively lightweight containing only 1 billion parameters, it is specifically designed to follow complex, multi-step instructions and reason deeply over structured and semi-structured texts, making it highly efficient for resource-constrained deployment scenarios while still maintaining state-of-the-art performance. Its instruction tuning, together with alignment through human feedback, enables the model to effectively handle domain-specific summarization and complex analytical tasks, notably without the need for additional fine-tuning or retraining. This carefully maintained balance between computational efficiency and advanced reasoning capability makes LLaMA 3.2-1B Instruct an ideal backbone choice for reliably generating highly structured and consistent financial reports in our system.

4.3 Prompting Strategy

Prompt engineering played an absolutely central role in ensuring the generation of both accurate and thoroughly decision-oriented analyst reports. Initial experimental tests with vanilla prompting immediately highlighted noteworthy limitations in ensuring factual grounding and enforcing structured, logical reasoning. To successfully address these issues, we strategically adopted a robust multi-step prompting strategy that intricately integrates deep financial discourse understanding with rigorous structured report generation.

4.4 Meta-Prompting Framework

To significantly enhance reasoning consistency and overall reliability, we designed an innovative meta-prompting (Hou et al., 2022) framework inspired by principles of multi-agent collaboration and distributed cognition. The system simulates a collaborative team of three specialized financial experts—each focusing on quantitative analysis, nuanced sentiment evaluation, and strategic interpretation respectively. By explicitly defining individual expert roles and precisely specifying the required structured output format, our framework effectively guides the model toward producing reports that are coherent, factually faithful, and sharply focused on investment decision-making.

4.5 Prompt Template Illustration

Figure 1 and Figure 2 vividly illustrate the systematic design of our structured multi-agent prompt. The prompt template shown in Figure 1 defines explicit analyst roles, clearly specifies the required structured output, and thoughtfully incorporates diverse contextual information such as company introduction details, recent news reports, financial performance metrics, and stock price movement data. This meticulous design ensures that the generated investment reports are comprehensive in scope, factually consistent throughout, and explicitly decision-oriented. In parallel, the accompanying workflow diagram (Figure 2) graphically depicts the systematic process by which the system processes an earnings call transcript. Specifically, the three analysts extract complementary insights focusing respectively on financial data, sentiment signals, and strategic factors, which are then synthesized and consolidated into a unified investment recommendation. This modular setup not only enhances interpretability but also enforces strict domain-specific rigor and analytical precision.

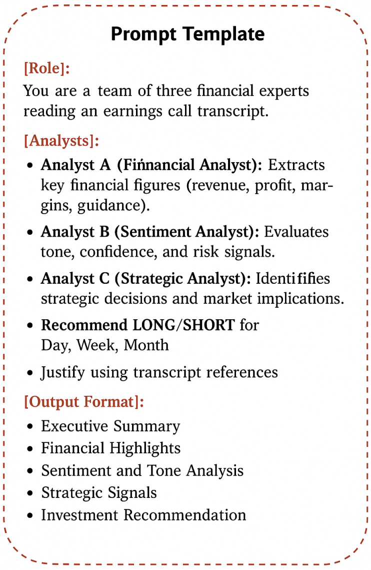


Figure 1: Structured multi-agent prompt template with role definitions, task breakdown, and output format.

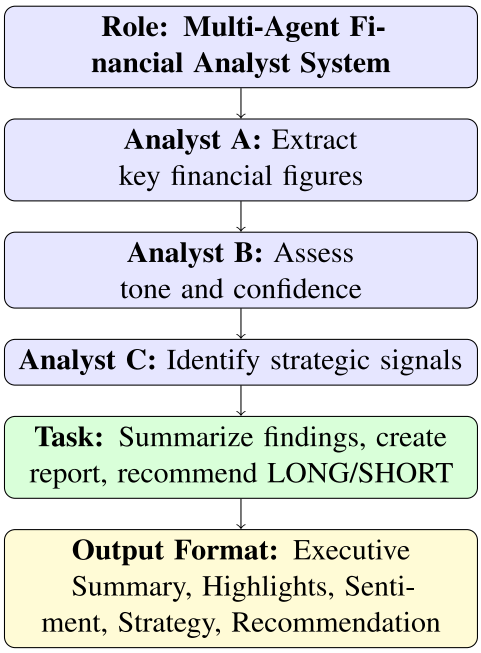


Figure 2: Workflow of the multi-agent prompt showing analyst roles, sequential tasks, and structured output.

5 Results

5.1 Evaluation Setup

The shared task employed both automatic and human evaluation. While participants could explore automatic metrics and LLM-based evaluations, the official ranking was determined solely by human judgments. Annotators were recruited via the Prolific platform (210 participants, with 34 excluded for failed attention checks). Each annotator reviewed 12 reports and made investment decisions (LONG/SHORT) for the next day, week, and month. Final system performance was scored by the average accuracy of these decisions across the three horizons. In addition, human raters assessed reports on five qualitative dimensions: clarity, logic, persuasiveness, readability, and usefulness, using a 7-point Likert scale.

5.2 Results

Our system, demonstrated strong empirical performance in the shared task. We ranked **second** out of twelve teams in terms of average financial decision accuracy, achieving an overall score of **0.579**. This reflects robust predictive alignment across daily (0.597), weekly (0.611), and monthly (0.529) investment horizons. On qualitative aspects, our reports received an average Likert score of **5.47** out of 7 from human judges, with particularly strong ratings for clarity (5.56) and readability (5.73).

Tables 1 and 2 present a breakdown of scores across all participating teams.

Team	Avg.	Day	Week	Month
DKE	0.581	0.596	0.577	0.570
Our Result	**0.579**	**0.597**	**0.611**	**0.529**
Jetsons	0.571	0.607	0.555	0.552
SigJBS	0.545	0.609	0.513	0.512
iiserb	0.537	0.576	0.558	0.477
PassionAI	0.537	0.588	0.557	0.466
Finturbo	0.524	0.504	0.568	0.500
Raphael	0.522	0.469	0.581	0.516
LangKG	0.518	0.589	0.542	0.424
SI4Fin	0.515	0.525	0.524	0.497
KrazyNLP	0.471	0.514	0.525	0.375
bds-LAB	0.462	0.478	0.434	0.474

Table 1: Average accuracy of financial decisions made by participants after reading the reports generated by each team, across daily, weekly, and monthly horizons.

Team	Avg.	Cl.	Lo.	Per.	Read.	Use.
LangKG	5.96	6.02	5.92	5.90	5.81	6.13
Jetsons	5.90	6.00	5.89	5.81	5.81	6.01
DKE	5.74	5.71	5.89	5.95	5.17	5.98
SigJBS	5.67	5.76	5.68	5.59	5.61	5.72
SI4Fin	5.56	5.52	5.84	5.60	5.06	5.80
Our Result	**5.50**	**5.56**	**5.45**	**5.32**	**5.73**	**5.47**
Raphael	5.49	5.51	5.61	5.51	5.09	5.74
KrazyNLP	5.29	5.15	5.49	5.21	5.01	5.59
iiserb	5.19	5.01	5.51	5.14	4.72	5.57
Finturbo	5.11	5.02	5.39	4.90	4.86	5.40
bds-LAB	4.99	4.91	5.21	5.03	4.55	5.27
PassionAI	4.70	4.64	4.74	4.39	4.88	4.86

Table 2: Average Likert scores (7-point scale) of generated reports across five qualitative dimensions: clarity, logic, persuasiveness, readability, and usefulness.

6 Conclusion

In this work, we presented our system for the *Earnings2Insights* shared task, focusing on generating insightful analyst reports from earnings call transcripts to aid investment decisions. Our approach, leveraging the **Meta LLaMA 3.2 1B Instruct** model with structured prompting and iterative refinement, achieved competitive results across quantitative and qualitative metrics.

Our team, ranked second out of twelve in financial decision accuracy. This ranking reflects the system's strong ability to produce reports that effectively guide human annotators toward correct investment choices (LONG or SHORT), demonstrating robust alignment with actionable financial outcomes, especially in capturing near-term market signals and predictive insights.

Qualitatively, our reports earned strong Likert ratings on a 7-point scale across clarity, logic, persuasiveness, readability, and usefulness. These ratings highlight the reports' balanced quality, indicating high accessibility, logical coherence, and practical value for readers, as driven by our meta-prompting framework.

These outcomes demonstrate the value of domain-specific reasoning in LLMs for faithful financial analysis. However, challenges like hallucinations and explainability persist. Future efforts will integrate external knowledge and enhance grounding for more trustworthy systems.

Acknowledgments

We thank the organizers of the Earnings2Insights shared task for their efforts in curating the dataset, designing the evaluation protocol, and fostering a collaborative environment for advancing financial text generation research.

References

Siyu An, Qin Li, Junru Lu, Di Yin, and Xing Sun. 2024. Finverse: An autonomous agent system for versatile financial analysis. *arXiv preprint arXiv:2406.06379*.

Chung-chi Chen, Jian-tao Huang, Hen-hsen Huang, Hiroya Takamura, and Hsin-hsi Chen. 2024. SemEval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1482–1491, Mexico City, Mexico. Association for Computational Linguistics.

Han Ding, Yinheng Li, Junhao Wang, and Hang Chen. 2024. Large language model agent in financial trading: A survey. *arXiv preprint arXiv:2408.06361*.

Tomas Goldsack, Yang Wang, Chenghua Lin, and Chung-Chi Chen. 2025. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10576–10593, Abu Dhabi, UAE. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yutai Hou, Hongyuan Dong, Xinghao Wang, Bohan Li, and Wanxiang Che. 2022. MetaPrompting: Learning to learn better prompts. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3251–3262, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yu-Shiang Huang, Chuan-Ju Wang, and Chung-Chi Chen. 2025. Decision-oriented text evaluation. *arXiv preprint arXiv:2507.01923*.

Aakanksha Jadhav and Vishal Mirza. 2025. Large language models in equity markets: Applications, techniques, and insights. *Techniques, and Insights (March 15, 2025)*.

Gautam Jajoo, Pranjal A Chitale, and Saksham Agarwal. 2025. Masca: Llm based-multi agents system for credit assessment. *arXiv preprint arXiv:2507.22758*.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Taejin Park. 2024. Enhancing anomaly detection in financial markets with an llm-based multi-agent framework. *arXiv preprint arXiv:2403.19735*.

Ming Shen, Raphael Shu, Anurag Pratik, James Gung, Yubin Ge, Monica Sunkara, and Yi Zhang. 2025. Optimizing llm-based multi-agent system with textual feedback: A case study on software development. *arXiv preprint arXiv:2505.16086*.

Takehiro Takayanagi, Hiroya Takamura, Kiyoshi Izumi, and Chung-Chi Chen. 2025. Can GPT-4 sway experts' investment decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 374–383, Albuquerque, New Mexico. Association for Computational Linguistics.

Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, and 1 others. 2024. Finrobot: An open-source ai agent platform for financial applications using large language models. *arXiv preprint arXiv:2405.14767*.

LangKG at the FinNLP 2025 - Earnings2Insights: Task-Adaptive LLMs To Generate Human-Persuasive Investment Reports

Shivika Prasanna
shivika.prasanna@fmr.com

Hui Su
hui.su@fmr.com

Abstract

In this paper, we address the challenge posed by the FinNLP 2025 shared task on Earnings2Insights: Analyst Report Generation for Investment Guidance, with our two-stage framework system. Success of these generated reports is measured on the correctness and persuasion of human investors for investment decisions across different time frames and has been evaluated using automated metrics and human evaluation. Our system comprises of two stages that incorporates a sophisticated analysis of investment-centric sentiments and personalities from the call transcripts and leveraging this information with a comprehensive cognitive reasoning framework to generate carefully curated, accurate and persuasive reports using LLMs for human-decision making. Our approach ranked 1st out of 12 teams on the human-evaluation average Likert Score and 2nd on the automated-evaluation average Likert Score, demonstrating competitive performance.

1 Introduction

Earnings Call Transcripts are rich in technical financial information that can be time-consuming for investors to parse quickly. Creating quality and correct investment analysis reports that can be convincing for decision-making requires human experts that can prevent scaling. Automatic generation of investment research reports from earnings call transcripts presents a fundamental paradigm shift in natural language processing (NLP) evaluation. Traditional financial NLP research has primarily focused on information extraction, summarization and sentiment analysis of earnings calls (Huang et al., 2025). Recent advances in large language models (LLMs) have proven to be increasingly promising in financial analysis tasks, especially with automatic analytical report generation (Goldsack et al., 2024). However, these approaches often optimize for content accuracy or similarity to reference summaries rather than persuasive effectiveness required for real-world investment decision-making.

The Earnings2Insights shared task (Takayanagi et al., 2025a) introduces a novel evaluation methodology where annotators are asked to make investment decisions based solely on generated reports, with correctness measured by actual investment outcomes across different time frames of 1 day, 1 week and 1 month. This evaluation paradigm reflects the recognition that traditional metrics may not be meaningful enough for financial analysis tasks and that the current LLMs are not yet completely suitable to serve as judges for investment guidance quality or correctness.

Previous studies on evaluation have demonstrated limitations of automatic evaluation in financial text generation. (Chen et al., 2024) highlighted challenges in numerical-aware language understanding and generation while (Goldsack et al., 2024) specifically addressed the gap between factual analysis and insightful report generation for earnings calls. The evaluation method adopted by the Earnings2Insights shared task has been proposed by (Takayanagi et al., 2025b) which demonstrated the potential for AI-generated content to influence expert decision-making. (Huang et al., 2025) introduced decision-oriented text evaluation emphasizing success through decision-making effectiveness over content similarity. (Mukherjee et al., 2022) contributed the ECTSum dataset for earnings call summarization, establishing important benchmarks for financial transcript processing.

We further breakdown the objectives of the task into several unique challenges and address them in our two-stage framework:

- Generated reports must be both analytically and psychologically sound to be persuasive;
- These reports must include necessary information to provide correct and actionable guidance across multiple investment time frames;

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

- Success of the generated reports depends on understanding various investor personalities and decision-making patterns to curate them to be suitable for a diverse range of profiles;
- Absence of ground truth poses a fundamental challenge for us to optimize LLMs to generate reports with human-like decision recommendations.

We recognize the importance of accurately communicating correct insights that would resonate with different investor profiles. Unlike approaches that cater to a generic investor population uniformly, our system considers how different investors such as growth, risk-aware and other types would interpret the same earnings information differently. Because in the real-world scenario, there can be several different personalities, we do not limit the large language model to the list of investor profiles, but instead allow it to identify the similarities and differences across these profiles when generating recommendations and evidences for decision-making. We introduce several key innovations in our proposed two-stage framework:

- Enhanced investment-centric sentiment classification system with 8 granular, distinct categories that capture management confidence levels and question dodge behavior from the transcripts;
- Systematic integration of investor personality considerations that guides report generation to address concerns and priorities of different investor types;
- Comprehensive and sophisticated 6-dimensional analysis framework covering financial performance, business fundamentals, risk assessment, forward outlook, Environmental, Social and Governance (ESG) considerations and personal factors;
- Explicit conviction scoring with reasoning and position sizing recommendations tailored to different investment time frames.

The remainder of this paper is organized as: Section 2 dives deeper into our two-stage framework; Section 3 describes our experiments and Section 4 presents the results and analysis of our approach as well as the evaluation results from the shared task. We conclude our work in Section 5.

2 Methodology

Our two-stage framework addresses the fundamental challenge of generating accurate reports that drive profitable human investment decisions through systematic analysis and psychological considerations. It combines two fundamental and distinct steps to achieve optimized reports that are highly human-persuasive.

To derive explicit and implicit information from the call transcripts, we first employed an enhanced investment-centric sentiment analysis and information extraction process. For each question-answer (Q/A) pair within the call transcript, the model, which acts as a '*Data Extractor*', first identifies the most suitable sentiment out of 8 possible defined categories, as described in Section 2.1. It then evaluates the confidence within the sentiment and identifies all key phrases that contributed to the sentiment classification for the Q/A pair. The model also identifies the tone of the speaker (avoidance or dodging) and provides a score with an interpretation. Finally, the model provides the interpretation on the investment signal for each exchange, which is crucial for the next stage in the framework.

The second stage focuses on personality- and analysis-driven investment report generation. The summary created in the previous stage and several different types of personalities along with their descriptions (Section 2.3) are fed as input to the model, which in this stage acts as a '*Cognitive Reasoner*', to generate reports. We employ a multi-perspective analysis that considers how different investor personality types would interpret the same earnings information, ensuring the reports address diverse investment philosophies and provide an unbiased, evidence-based recommendation. This analysis is guided by application of our 6-dimensional (6D) framework covering various aspects described in Section 2.2. The framework generates conviction scoring with percentages for clear interpretation that quantify uncertainty. Reports provide recommendations for 1-day, 1-week and 1-month, recognizing that short-term price movements substantially differ from long-term performance drivers. Throughout this process, the framework ensures error avoidance by identifying common analytical errors such as optimism bias, making recommendations without considering investor suitability, and strictly adhering to the framework, ensuring the generated reports maintain analytical rigor, correctness (no hallucinations) and practical applicability with persuasion for human decision-making.

2.1 Investment-Centric Sentiment Classification System

In the first stage of our system, we developed an 8-category sentiment classification specifically designed for earnings call investment analysis as follows:

- Bullish: Strong positive, growth accelerating, beating expectations
- Optimistic: Moderately positive, things improving, meeting expectations
- Cautious: Uncertain but not negative
- Neutral: Balanced, no strong directional signals
- Concerned: Worried tone, challenges mentioned, defensive responses
- Bearish: Negative outlook, problems acknowledged, guidance cuts
- Evasive: Avoiding questions, deflecting, noncommittal answers
- Confident: Stronger than Optimistic and Bullish, shows conviction in guidance

The sentiment classification for each Q/A pair provides a deterministic confidence level on a scale of 1 (least confidence) to 5 (highly confident), key phrases that triggered the classification and the score, a question dodge score on a scale of 0 (direct response) to 3 (completely evasive) and an investment signal interpretation highlighting what this would mean for stock performance.

2.2 Structured Analysis Framework

We employ a comprehensive framework to further analyze the call transcripts to generate the recommendation reports. Traditional earnings call analysis can often miss the holistic perspective required to make an investment decision. Our framework ensures systematic coverage of these 6 factors that can influence the investment outcomes.

- Financial Performance: Captures the quantitative performance.
  - Revenue trends, growth rates, profitable margins
  - Beat/miss versus guidance, expectations
  - Key performance indicators
- Business Fundamentals: Evaluates competition, management track record, management quality.
  - Market position, competitive advantages
  - Management quality, strategic vision
  - Growth catalyst, expansion opportunities
- Risk Assessment: Identifies regulatory threats, vulnerabilities, or any factors that could derail investment thesis.
  - Industry and company-specific risks
  - Operational challenges
  - Market volatility factors
- Forward Outlook: Evaluates guidance credibility based on management's historical accuracy and identifies any types of catalysts.
  - Management guidance and expectations
  - Industry trends affecting future performance
  - Short and medium-term price drivers
- Personal Factors: Growth investors and value investors analyzing identical earnings calls can reach different conclusions.
  - Risk tolerance
  - Financial situation
  - Investment goals
  - Length of time to hold investment before needing funds
- Conviction Scoring: To allow investors to make evidence-based decisions, instead of only recommending Long or Short, this framework also quantifies the uncertainty and translates it into actionable interpretation.
  - 90-100%: Multiple bullish signals, minimal risks, clear catalysts
  - 70-89%: Strong thesis, but some uncertainty/risks present
  - 50-69%: Mixed signals, requires smaller position sizing
  - 30-49%: Weak thesis, high uncertainty, avoid or minimal exposure
  - 0-30%: Strong negative signals, consider short position

With this structure framework, we addressed common analytical failures such as overlooking competitive threats or providing guidance without conviction levels for position sizing. Each dimension addresses a specific cognitive bias that affect both human analysts and automated systems.

2.3 Investor Personality Integration

Every investor has certain preferences when making an investment decision. Our system takes this into account through prompt engineering to consider multiple perspectives and identify common and different personality elements that can affect an investment decision. Specifically, we prompt the 'Cognitive Reasoner' model to analyze (1) how different personality types would interpret the same information; (2) common traits that tie suitable investors together; (3) identification of investor types beyond angel, venture, personal, institutional and crowdfunding, that would be best suited for the investment; (4) balanced reasoning that synthesizes multiple viewpoints. Through this approach, we achieve a more balanced recommendation, rather than an overly or under optimistic one.

2.4 Error Prevention

We address over- or under-optimism by implementing a systematic error identification and avoidance. The instructions given to the model include avoiding overoptimism bias from confident management tone, distinguishing between genuine guidance and reality catch-up, questioning margin expansion claims without clear drivers and flagging evasive responses to specific questions.

3 Experiments

3.1 Dataset & Task Setup

The dataset provided in the Earnings2Insights shared task consists of 64 earnings call transcripts. Forty transcripts are ECTSum transcripts with reference summaries, and 24 are Professional subset transcripts without reference summaries. Each transcript consists of prepared remarks and Q/A pairs. The transcript follows the typical structure of a call transcript, following a conversational format between the speaker and an audience. The primary objective in this shared task is to generate investment reports for all 64 transcripts that can convince human evaluators to make profitable and correct trading decisions. Each report has been evaluated using automated metrics such as Likert Score (average score between 1-7 Likert ratings of Persuasiveness, Logic, Usefulness, Readability, and Clarity) and Win Rate vs Analyst Report (average score showing how often the report outperformed a professional analyst report in a pairwise comparison), and human evaluation using Likert Score and Average Accuracy (across three different time frames of 1-day, 1-week and 1-month) that measures the accuracy of the investment decisions.

For our experiments, we utilize GPT-4o (Hurst et al., 2024), due to its high speed, low latency and high accuracy, for this task. We set the temperature to 0.5 for both 'Data Extractor' and 'Cognitive Reasoner', with number of maximum tokens as 4096 and top_p parameter set to 0.95.

3.2 Implementation Details

Our framework implements a novel Task-Adaptive Model Utilization that utilizes the cognitive processing of LLMs, for the two stages. We use 'task-adaptive' to denote prompt-based adaptation of the model to different tasks or roles. Through this, we recognize the importance of matching model operation to specific cognitive demands rather than applying uniform processing.

The model first acts as a 'Data Extractor' to extract key implicit and explicit information from the transcripts by carefully analyzing the prepared remarks and each Q/A pair. We implement rigid prompt engineering that forces systematic analysis of each Q/A exchange through predefined analytical steps such as speaker role identification, tone analysis, sentiment classification, evidence extraction and output format for stage two of the framework. The high constraint setup prevents the model from hallucinations and inconsistency.

Then, the model transitions its role to a 'Cognitive Reasoner' to perform flexible reasoning, leveraging its sophisticated inference capabilities for multi-perspective synthesis. We reduce structural constraints while introducing cognitive frameworks that guide comprehensive analysis. The model synthesizes different viewpoints during reasoning. This approach generates more nuanced reports that addresses diverse investor concerns, providing curated recommendations for different investor profiles. Prompts are available in Appendix A.

The dual-mode architecture we employ balances precision with creativity. Extraction tasks require high accuracy and persuasive report generation demands flexible reasoning with psychological insight. By optimizing each stage for specific tasks, our framework achieves both analytical precision and human persuasiveness, shown in Section 4.

Table 1: LangKG Performance Summary

Metrics	Measure	Score
Human Evaluation	Day	0.589
	Week	0.542
	Month	0.424
	Average	0.518
Human Likert Score (1-7)	Clarity	6.02
	Logic	5.92
	Persuasiveness	5.90
	Readability	5.81
	Usefulness	6.13
	Average	5.96
Automated	Average Likert Score (1-7)	4.903
	Win Rate vs Analyst Report	0.881

4 Results

We report the automated and human evaluation results in Table 1. Our results demonstrate the success of the approach using human and automated Likert Score. While there is still scope of improvement on accuracy of the model-generated reports, on the leader board, we rank 1st out of 12 teams in the human average Likert Score, 4th for the 1-day time frame in the human average accuracy, 2nd in automated average Likert Score and 4th in Win Rate vs Analyst report metrics.

5 Conclusion

We present a Task-Adaptive LLM framework that addresses the fundamental challenge of generating investment reports optimized for human decision-making, as posed by the FinNLP Earnings2Insights: Analyst Report Generation for Investment Guidance shared task. Our proposed two-stage approach that combines enhanced investment-centric sentiment analysis with investor personalities and investment analysis framework demonstrates that successful financial NLP requires understanding both linguistic signals and psychological factors that drive investment decisions, along with financial (investment) knowledge. Our key contribution lies in the dual-mode cognitive architecture where, in the first stage the model acts as a ‘Data Extractor’ using 8 sophisticated sentiments and in the second stage, as a ‘Cognitive Reasoner’ using investor personality modeling and our 6D analysis framework. Our experiment results validate our methodology, with our approach ranking 1st out of 12 teams in the human average Likert Score, 2nd in the automated average Likert Score. On financial accuracy, our methodology shows promise by ranking 4th in the 1-day time frame in human evaluation.

Ethics Statement

This research has been conducted for academic purposes. Investments may be risky and investors are advised to use their discretion, instead of solely relying on the output of the models.

References

Chung-Chi Chen, Jian-Tao Huang, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2024. Semeval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1482–1491.

Tomas Goldsack, Yang Wang, Chenghua Lin, and Chung-Chi Chen. 2024. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. *arXiv preprint arXiv:2410.01039*.

Yu-Shiang Huang, Chuan-Ju Wang, and Chung-Chi Chen. 2025. Decision-oriented text evaluation. *arXiv preprint arXiv:2507.01923*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and 1 others. 2022. Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. *arXiv preprint arXiv:2210.12467*.

Takehiro Takayanagi, Tomas Goldsack, Kiyoshi Izumi, Chenghua Lin, Hiroya Takamura, and Chung-Chi Chen. 2025a. Earnings2Insights: Analyst Report Generation for Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China. Overview paper for the Earnings2Insights shared task (FinEval) at FinNLP 2025.

Takehiro Takayanagi, Hiroya Takamura, Kiyoshi Izumi, and Chung-Chi Chen. 2025b. Can gpt-4 sway experts’ investment decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 374–383.

A Appendix A

We present the ‘Data Extractor’ and ‘Cognitive Reasoner’ prompts here.

Data Extractor Prompt

```
You are an expert at analyzing sentiments from text.
## [INPUT]

Earnings Call Transcript: $earnings_transcript

## [TASK]

1. Carefully read through the entire transcript which details the speakers for an investment company
     followed by a Question/Answer session where the speakers are posed questions surrounding
     investing in their company or their product.
2. Identify the key points in the prepared remarks and the transcript and create a summary.
    - For all numbers, dates, amounts, etc, only include what is given in the input.
    - Do not round off, do not change any values.
3. For each Q/A pair:
    Step 1: Identify who is asking the question, do not change names
    Step 2: Identify who is being asked the question, do not change names
    Step 3: Carefully analyze the tone of the answer
    Step 4: Generate the most suitable sentiment based on your analysis
3. Sentiment must be classified into:
    - **BULLISH**: Strong positive, growth accelerating, beating expectations
    - **OPTIMISTIC**: Moderately positive, things improving, meeting expectations
    - **CAUTIOUS**: Uncertain but not negative, "Wait and See" tone
    - **NEUTRAL**: Balanced, no strong directional signals
    - **CONCERNED**: Worried tone, challenges mentioned, defensive responses
    - **BEARISH**: Negative outlook, problems acknowledged, guidance cuts
    - **EVASIVE**: Avoiding questions, deflecting, non-committal answers
    - **CONFIDENT**: Stronger than OPTIMISTIC and BULLISH, shows conviction in guidance.

## [OUTPUT]

Strictly generate a JSON object in the following format. Do not add any markdown or strings before
     or after the JSON object.

{
    "prepared_remarks_summary": <Summarize the transcript>,
    "sentiment": {
        [
            "From": <Who is asking?>,
            "To": <Who is being asked?>,
            "Question": <List question>,
            "Answer": <List answer>,
            "Sentiment": <Specify sentiment>,
            "Confidence": <1-5 scale within that sentiment>
            "Key Phrases": <Specific words or phrases that indicated this sentiment>,
            "Question Dodge Score": <0-3 scale, 0=direct response, 3=completely evasive>,
            "Signal": <What this means for stock performance, investment implication>
        ],
        [...]
    }
}
```

Cognitive Reasoner Prompt

```
You are a professional financial analyst with expertise in investments.

## [TASK]
Generate an investment report from the financial earnings call transcript summary that will convince
     human investors to make profitable and correct buy or sell decisions for the next day, week,
     and month
Here are some things you can consider when analyzing the summary:
    1. Would you recommend this to an investor?
    3. What type of investors would this be best suited for and why?
    4. What is common in personality of the investors you select?

Your report should provide CLEAR investment guidance that will help investors make PROFITABLE,
     CORRECT and INFORMED decisions.

## [INPUT]

Financial Earnings Call Transcript Summary: Summary of the original call transcript along with each
     Question-Answer pair, an assigned sentiment to the pair, and a signal for what it means for
     stock performance.

$financial_earnings_summary

Examples of Investor Types: Examples of personality types and what investments they usually prefer.

$investor_df

## [ANALYSIS FRAMEWORK]
This is to help you create a framework structure for your report.
1. FINANCIAL PERFORMANCE
    - Revenue trends, growth rates, profitability margins
    - Beat or miss vs guidance and expectations
    - Key performance indicators and metrics
    - Leading, lagging or matching

2. BUSINESS FUNDAMENTALS
    - Market position, competitive advantages
    - Management quality and strategic vision
    - Growth catalyst and expansion opportunities

3. RISK ASSESSMENT
    - Industry and company-specific risks
    - Operational challenges
    - Market volatility factors

4. FORWARD OUTLOOK
    - Management guidance and expectations
    - Industry trends affecting future performance
    - Short and medium-term price drivers

5. PERSONAL FACTORS
    - Risk tolerance
    - Financial situation
    - Investment goals
    - Length of time to hold investment before needing funds

6. CONVICTION SCORING
    - 90-100%: Multiple bullish signals, minimal risks, clear catalysts
    - 70-89%: Strong thesis, but some uncertainty/risks present
    - 50-69%: Mixed signals, requires smaller position sizing
    - 30-49%: Weak thesis, high uncertainty, avoid or minimal exposure
    - 0-30%: Strong negative signals, consider short position
```

Cognitive Reasoner Prompt

```
## [ANALYZING INSTRUCTIONS]
1. Read through the given financial earnings transcript summary carefully.
2. Read through each Question-Answer pair and the sentiment assigned to it.
3. Provide a recommendation based on your analysis for the next day, week, and month. It can be LONG
   position or SHORT position.
   - LONG: buy, assumption is that the price of a security will increase over time
   - SHORT: sell, assumption is that the price of a security will decrease over time.
4. Identify what types of investors are best suited to invest in the product.
   - The given input for investors list is only to help you.
   - You may list some other investor types.
   - Be careful in identifying if the product should be recommended.
5. Think through the different personalities that you are selecting as suitors and identify what
   ties them together.
   - Use this for your reasoning.
   - Specifically, ask yourself
       - "What do these investors have in common and why should this investment be the best for
          them?"
       - "If I were an advisor, what kind of investor personality would I recommend this product
          to?"
6. AVOID THESE COMMON ERRORS:
   - Don't be overly optimistic just because the tone of the management sounds confident.
   - Watch for guidance raises that are actually just catching up to reality.
   - Be skeptical or margin expansion claims without clear drivers.
   - Flag when management avoids giving specific numbers.
7. Only use information from the given input call transcript. Do NOT use any external information or
   data.

## [REPORT INSTRUCTIONS]
1. Your report MUST be persuasive enough that when human evaluators read it, they can make the
   CORRECT investment decisions.
2. Use professional, authoritative tone for investors.
3. Support all your conclusions with evidence from the given input call transcript.
4. Balance all bullish and bearish factors.
5. Focus on driving stock performance over the next day, week, and month.
6. Use the ANALYSIS FRAMEWORK while writing your report.
7. Strictly adhere to the report structure given to you.
8. Ensure the report is concise, short and easy to read.

## [REPORT STRUCTURE]
1. Executive summary: 2-3 clear sentences with a clear "LONG" or "SHORT" recommendation for each
   timeline in natural language.
2. Driving stock performance: provide recommendation of "LONG" or "SHORT" for each of the next day,
   week, and month.
   - The recommendation can be different for each timeline. Use only "LONG" or "SHORT" and no
      synonyms of these words.
   - In ONE line, explain what would change your recommendation.
   - Be careful during your analysis and recommendation for each timeline.
   - You MUST provide the recommendation for all 3 timelines.
   - Explain why you are recommending the position.
   - Do not use "We" or "I" while recommending.
   - Do NOT use words like "downgrade" during your recommendation when describing why you would
      change from "LONG" to "SHORT".
3. Additional Information:
   3a. Financial highlights: key numbers and performance vs expectations with evidences from the
       call transcript.
   3b. Strategic updates: business initiatives, market developments from the call transcript.
   3c. Risk factors: highlight any concerns, potential headwinds.
   3d. Investment thesis: why buy or sell, what timings to consider.
   3e. Environmental, Social, Governance (ESG) and Other Qualitative Factors: identify management
       comments on sustainability, employee issues, ethics and other such factors and assess the
       credibility. Also evaluate any ESG risk possible that could impact the investment.
   3f. Insights: provide detailed, concise insights into the CONVICTION SCORING, strategy, risks,
       and market positioning mentioned in the call transcript. Include anything from the ANALYSIS
       FRAMEWORK here.
   3g. Investor Suitability: list the best suited investor profiles.
       - You may use some examples from the list.
       - Include the difference in the strategy of investment for different types of investors.

Your report will be evaluated on whether it convinces investors to make correct investment decisions.
```

Jetsons at the FinNLP-2025 - Earnings2Insights: Persuasive Investment Report Generation Using Single And Multi-Agent Frameworks

SaiKrishna Rallabandi, Nikhil Kohli, Parag Pravin Dakle, Khyati Morparia, Ojas Raundale, Preethi Raghavan
Fidelity Investments
{saikrishna.rallabandi, nikhil.kohli, paragpravin.dakle, khyati.morparia, ojasjeetendra.raundale, preethi.raghavan}@fmr.com

Abstract

In this paper, we present four agent-based frameworks using the ReAct paradigm - two single-agent and two modular multi-agent systems - for automated report generation developed for the Earnings2Insights shared task at FinNLP-2025. Each single-agent solution is powered by a Writer agent, while the multi-agent frameworks incorporate Feedback agents to refine and enhance report quality through iterative collaboration. To evaluate generated reports, we introduce a comprehensive LLM-as-a-Judge framework that integrates six metrics to rank outputs across multiple dimensions. Our ensemble approach achieves an average financial accuracy of 0.571 (3rd place) and an average Likert score of 5.90 (2nd place) in human evaluations, with particularly strong performance in readability (1st place) and next-day prediction accuracy (2nd place).

1 Introduction

Earnings calls are among the most consequential communication events in global financial markets. During these sessions, company executives present quarterly performance, provide forward-looking guidance, and respond to probing questions from analysts. The transcripts of these calls are lengthy, noisy, and multi-speaker in nature, yet they contain critical signals that influence billions of dollars' worth of investment decisions. Professional analysts typically distill these signals into structured reports that combine factual accuracy with persuasive narrative. However, manually creating such reports is labor-intensive and does not scale to the thousands of earnings calls that occur each quarter. This motivates research into automated methods that can generate decision-oriented investment reports directly from transcripts (Takayanagi et al., 2025b).

Large Language Models (LLMs) have shown promise in financial NLP tasks such as summarization, sentiment analysis, and question answering. Yet, applying LLMs to investment report generation requires moving beyond extractive summarization: systems must produce outputs that are both *analytically rigorous* and *persuasive enough* to support real financial decisions (Chen et al., 2024; Goldsack et al., 2025). The Earnings2Insights shared task (FinEval @ FinNLP/EMNLP 2025) is designed to evaluate precisely this capability. Given an earnings call transcript—with the option to incorporate aligned external information—participating systems must produce an analyst-style report that concludes with explicit Long/Short recommendations across three horizons (next day, week, and month). Crucially, the task emphasizes decision accuracy as the primary evaluation objective, shifting focus away from surface-level similarity metrics.

Challenges. Generating high-quality analyst reports presents several intertwined challenges. First, the language of earnings calls is often hedged, promotional, and strategically vague, requiring systems to "read between the lines." Second, factual reliability is paramount: even minor numeric inaccuracies or misattributions can undermine credibility. Third, persuasiveness matters. Reports must not only be accurate but also written in a style that instills confidence and convinces investors to take action. Finally, evaluation itself is non-trivial, as traditional n-gram overlap or embedding-based metrics fail to capture whether a report actually improves investment decisions meaningfully (Huang et al., 2025).

In this paper, we present several single and multi-agent modular frameworks that integrates (i) single-agent writer baseline that only uses the earnings call transcripts, (ii) data-enhanced generation with quarter-over-quarter and year-over-year fundamentals, and (iii) multi-agent writer and feedback ReAct (Yao et al., 2022) framework that couples a writer with automated feedback modules, includ-

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

ing a *Financial Expert* to cross-check metrics, a *Risk Analyst* to evaluate coverage of downside factors, and a *Persuasiveness Expert* to assess narrative strength. Iterative reasoning and action loops enable continuous refinement, ensuring that final reports are both factually accurate and rhetorically compelling.

Contributions. Our work makes the following contributions:

1. We present several single and multi-agent frameworks for automatic analyst report generation that include transcript-only and data-enhanced inputs.

2. We introduce a *feedback* or *validation* module that systematically reduces numeric and attributional errors by checking generated claims against structured fundamentals.

3. We develop a comprehensive evaluation harness that combines industry and shared task requirements in a set of six metrics. The evaluation harness is utilized to rank and select the best output across several systems.

4. We provide empirical evidence that the proposed methods generate reports that contain accurate investment recommendations and are factual, persuasive, and logical. The output of our N-way comparative selection obtains an average financial accuracy of 0.571 (ranking 3rd) and an average Likert Score of 5.90 (ranking 2nd) in human evaluations.

2 Dataset

The shared task dataset (Takayanagi et al., 2025a) comprises 64 earnings call transcripts, organized into two subsets. The first subset includes 40 calls paired with human-written summaries from ECT-Sum (Mukherjee et al., 2022), providing optional supervisory signals for training and benchmarking. The second subset comprises 24 transcripts paired with professional analyst reports authored by domain experts. These gold reports are withheld during training and used only for post-hoc evaluation by the organizers. Collectively, the dataset spans multiple industries and time periods, ensuring a diverse and representative benchmark for investment report generation. Participants are required to submit a single JSON file where each entry contains the earnings call code (ECC) and the generated report.

3 ReAct Prompting

Yao et al. (2022) introduced the *Reasoning and Acting* (*ReAct*) prompting paradigm, where reasoning and acting in large language models are used collaboratively by generating interleaved verbal reasoning traces and task-specific actions. This enables models to perform dynamic reasoning to create, maintain, and adjust action plans while interacting with external environments and addressing limitations of chain-of-thought reasoning, such as hallucination and error propagation. In this work, we leverage the Reasoning step to evaluate the generated report and the Acting step to update the report using the evaluation feedback from the Reasoning step.

4 Methodology

We propose a multi-agentic system that is designed to generate highly persuasive and analytically robust investment reports from earnings call transcripts, leveraging advanced prompt engineering, persona curation, and external financial data sources. The main agents that are used in the solution are as follows:

4.1 Writer Agents

We define a *Writer Agent* as an LLM prompt-based agent that portrays an expert data analyst. The agent accepts the transcript, any additional financial information, and evaluation feedback (if applicable) to generate the analyst report. We further define two different types of *Writer Agents*:

1. A *Simple Writer Agent* with a minimal set of instructions

2. An *Advanced Writer Agent* that uses an extensive set of instructions and additional financial information. (See Appendix A for complete prompts).

4.2 Feedback Agents

A *Feedback Agent* is an LLM prompt-based agent that portrays a financial expert with a specific skill set and reviews the generated report. We use four main types of Feedback Agents:

1. Skeptical Financial Expert Panel: This agent evaluates the generated report on five dimensions to generate a detailed review for the report - Factual accuracy, Completeness, Realism, Persuasiveness, and Transparency.

2. Financial Expert: This agent focuses on reviewing an analyst report for factual accuracy regarding financial metrics, numbers, and calculations.

3. Risk Analyst: This agent assesses whether the risk analysis presented in the generated report is complete and realistic.

4. Persuasiveness Expert: The main focus of this agent is to review the report for persuasiveness, clarity, and conviction dimensions. As part of the review, the agent also identifies specific sections of the report that require revision and outlines the aspects that need improvement.

4.3 Rewriter Agent

The *Rewriter Agent* is tasked with rewriting a given report using one or more reviews of the same. The agent must understand the given reviews and then rewrite the report to address essential aspects from the review.

4.4 External Financial Data Integration

To enhance the factual accuracy and analytical depth of each report, we integrate external financial data using the Alpha Vantage API. For each company and quarter, we retrieve three things: (i) Current quarter earnings data, (ii) Previous quarter earnings data, and (iii) Same quarter last year earnings data.

These data points include granular financial metrics such as total revenue, gross profit, operating income, EPS, and other relevant indicators. The retrieved data was given as input to the *Advanced Writer Agent* and all multi-agent flows, enabling the agent to perform explicit quarter-over-quarter and year-over-year comparisons. This helps the agent identify the direction of the company across various horizons for both short-term and long-term investments.

Sample External Data Structure:

```
{
  "ECC": "ABM_q3_2021",
  "Ticker": "ABM",
  "Quarter": "3",
  "Year": "2021",
  "most_recent_earnings": { ... },
  "previous_earnings": { ... },
  "same_qtr_last_year_earnings": { ... }
}
```

By combining transcript content and structured external data, our system generates reports that not only reflect the nuances of management discussions but are also grounded in quantitative performance trends.

4.5 Multi-Agent Feedback Framework

We model our multi-agent feedback framework using the ReAct prompting paradigm and propose two distinct variations for analyst report generation. The variations differ in the complexity of the feedback loop. For both frameworks, we iterate over the loop several times until one of the two conditions is satisfied - the maximum number of allowed iterations is complete, or the feedback agent(s) approve the generated report.

4.5.1 Simple Multi-Agent Feedback Framework

In this framework, the *Simple Writer Agent* is used to generate the analyst report, and the following ReAct paradigm is followed:

Reasoning : The *Skeptical Financial Expert Panel Feedback Agent* reviews the generated report and provides a detailed review on several dimensions with an accept/reject verdict.

Action : If the verdict in the reasoning step is not an accept then, the *Simple Writer Agent* updates the report based on the reasoning provided in the review.

To avoid infinite Reasoning and Action loops, we run the framework for a maximum of 3 iterations and use the output of the last iteration, irrespective of the final verdict.

4.5.2 Advanced Multi-Agent Feedback Framework

This framework is similar to the previous framework in its usage of the *Simple Writer Agent* to generate the analyst report. It differs in the implementation of the ReAct paradigm.

Reasoning : This framework uses *Financial Expert Feedback Agent*, *Risk Analyst Feedback Agent*, and *Persuasiveness Feedback Agent* to generate separate review of generated report.

Action : If the verdict in the reasoning step for any of the feedback agents is not an accept then, the *Rewriter Agent* updates the report based on the reasoning provided in the review.

Due to the presence of multiple reasoning feedback agents, we employ the Reasoning and Action loop only once for this framework.

We use GPT-4o (OpenAI et al., 2024) as the LLM for all agents. The *Advanced Writer Agent* uses a *temperature=0.7*, *max_tokens=4000*, and *top_p=0.95*. For all other agents, we use a *temperature=0.7*, *max_tokens=16000*, *top_p=1.0*, *frequency_penalty=0*, and *presence_penalty=0*.

4.6 Output Report Templates

4.6.1 Simple Writer Agent Output Format

```
<one sentence summary recommendation
with conviction level>
Financial Overview
...
Key Drivers
...
Risks
...
Opportunities
...
Final Recommendation
...
```

4.6.2 Advanced Writer Agent Output Format

```
Company Overview
...
Executive Summary
...
Financial Performance with comparative
analysis
...
Key Financial Metrics
...
Strategic Outlook and Investments
...
Insights from Q\&A Session
...
Projections and External Perspectives
...
Conclusion and Investor Takeaways
...
Final Recommendation: Long/Short for
next day, week, and month, with rationale
```

5 Evaluation

The generated reports are evaluated using a mixture of automated and human evaluation. For automated evaluation, three different metrics are used - Average Likert Score, Win Rate vs Analyst Report, and LLM-as-a-Judge.

1. Average Likert Score: Average score based on 1–7 Likert ratings of Persuasiveness, Logic, Usefulness, Readability, and Clarity.

2. Win Rate vs Analyst Report: Average score showing how often the generated report outperformed a professional analyst report in pairwise comparisons (ties excluded).

3. LLM-as-a-Judge: We define five metrics and an aggregate metric to evaluate a generated report using analyst reports requirements defined by SEC, FINRA[1], sector-specific requirements defined in the industry, and the Earnings2Insights shared task.

In the human evaluation, the average accuracy of financial decisions in the generated reports was computed by manually evaluating the reports.

5.1 LLM-as-a-Judge Metrics

We define a set of five metrics, with detailed guidelines on the weight of each metric and how to interpret low vs. high scores (See Appendix C for entire prompt).

Content Accuracy & Faithfulness: This metric assesses how accurately the report reflects the actual content of the earnings call. It checks explicitly if financial figures, metrics, statements, and direct quotes accurately reflect the content of the earnings call transcript.

Analytical Depth & Insight Quality: This metric evaluates the analysis presented in the report in terms of how it presents the underlying trends, connects financial performance to a broader market context, and provides meaningful insights that demonstrate deep business understanding rather than surface-level observations.

Investment Recommendation Quality: This metric evaluates the presence and soundness of required investment recommendations. Examine whether recommendations are well-supported by evidence, include appropriate risk assessments, and have realistic price goals with sound methodology.

Structure, Clarity & Presentation: This metric evaluates the structure of the report, specifically, organization, readability, and professional quality of the analysis. It measures logical flow, clear communication, proper formatting, and whether the content is easy to follow and understand for investment decision-making purposes.

[1]https://www.finra.org/rules-guidance/rulebooks/finra-rules/2241

Comprehensive Coverage & Completeness: This metric measures how thoroughly the analysis covers all material aspects of the earnings call.

Overall Grade: We compute the overall aggregate metric using the scores for each metric on a scale of 0-100 using the following formula: *Overall Grade = 0.25 * Content Accuracy + 0.20 * Analytical Depth + 0.20 * Investment Recommendation + 0.15 * Structure + 0.20 * Comprehensive Coverage*

We use the *Overall Grade* to grade the outputs of each system for each earnings call transcript and select the best one for the shared task submission. For the LLM-as-a-Judge Evaluator, we use Claude-Sonnet-4[2] with *temperature=0.5*, *max_tokens=4096*, and *top_p=0.95*

6 Results

Table 1 shows the results of the human evaluation of the analyst reports generated using an ensemble of all systems. We use the LLM-as-a-Judge Overall Grade to determine which system's output was selected in the ensemble.

Metric/ Time Horizon	**Score**	**Leaderboard Position**
Average Accuracy of Financial Decisions		
Average	0.571	3rd
Next Day	0.607	2nd
Next Week	0.555	7th
Next Month	0.552	2nd
Likert Score (Scale: 1-7)		
Average	5.90	2nd
Clarity	6.00	2nd
Logic	5.89	2nd
Persuasiveness	5.81	3rd
Readability	5.81	1st
Usefulness	6.01	2nd

Table 1: Human Evaluation Results of the ensemble of proposed methods

In addition to the human evaluation, reports generated using an ensemble of the proposed methods also achieve an Average Likert Score of 4.834 when computed automatically and an Average Win Rate vs Analyst Report Score of 0.762.

7 Conclusion

The work presented in this paper addresses the problem of automatically generating persuasive investment analyst reports from earnings call transcripts using a dataset shared as part of the Earnings2Insights shared task at FinNLP-2025. The proposed solution defines two classes of agents - *Writer* and *Feedback* and uses two variations of the *Writer Agent* as single agent solutions and a combination of the Simple Writer Agent with different Feedback Agents as multi-agent frameworks using the ReAct paradigm. Our key contributions include the systematic integration of structured financial fundamentals to reduce numeric errors, the deployment of specialized feedback agents that mirror real-world analyst review processes, and the development of a comprehensive evaluation framework that performs thorough evaluation, with our ensemble approach achieving competitive performance, including 2nd place in overall Likert scores and 1st place in readability across multiple evaluation dimensions.

[2]https://www.anthropic.com/news/claude-4

References

Chung-chi Chen, Jian-tao Huang, Hen-hsen Huang, Hiroya Takamura, and Hsin-hsi Chen. 2024. SemEval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1482–1491, Mexico City, Mexico. Association for Computational Linguistics.

Tomas Goldsack, Yang Wang, Chenghua Lin, and Chung-Chi Chen. 2025. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10576–10593, Abu Dhabi, UAE. Association for Computational Linguistics.

Yu-Shiang Huang, Chuan-Ju Wang, and Chung-Chi Chen. 2025. Decision-oriented text evaluation. *Preprint*, arXiv:2507.01923.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2022. ECTSum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10893–10906, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Takehiro Takayanagi, Tomas Goldsack, Kiyoshi Izumi, Chenghua Lin, Hiroya Takamura, and Chung-Chi Chen. 2025a. Earnings2Insights: Analyst Report Generation for Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China. Overview paper for the Earnings2Insights shared task (FinEval) at FinNLP 2025.

Takehiro Takayanagi, Hiroya Takamura, Kiyoshi Izumi, and Chung-Chi Chen. 2025b. Can GPT-4 sway experts' investment decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 374–383, Albuquerque, New Mexico. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

A Prompts for different agents used

A.1 Advanced Writer Agent Details

At the core of this approach is a comprehensive analyst prompt, explicitly crafted to guide the language model in producing investment reports that meet the standards of professional financial analysis. The prompt instructs the model to act as an expert financial analyst, synthesizing insights from the earnings call transcript and multiple quarters of earnings data. Key aspects of the prompt design include:

- **Objective:** The model is tasked with creating an investment report using the earnings call transcript, current quarter earnings data, previous quarter earnings data, and the same quarter's data from the prior year.

- **Writing Style and Tone:** The report must maintain a professional and analytical tone, utilize precise language for financial concepts, and base insights on data-driven metrics and comparisons.

- **Deep Analysis:** The prompt directs the model to interpret not only explicit statements but also underlying sentiment, tone, and hesitation in the transcript, mirroring the nuanced work of human analysts.

- **Actionable Guidance:** The generated report must provide clear investment advice and a final recommendation, including Long/Short positions for three time frames: next day, next week, and next month.

- **Structured Response:** The output follows a detailed template (see below), ensuring consistency and coverage of all relevant analytical dimensions.

B Example of External Financial Data Augmentation

To illustrate our integration of external financial data, we provide an example of the structured earnings data retrieved for a single company-quarter using the Alpha Vantage API below. This data was programmatically injected into the writer prompt to enable explicit quarter-over-quarter and year-over-year comparisons in generated investment reports.

The following JSON excerpt corresponds to ECC `ABM_q3_2021` (ABM Industries, Q3 2021):

This structured financial data enables our system to ground investment analysis in quantitative performance trends and to generate more precise, actionable insights for investors.

C LLM-as-a-Judge Evaluation

The evaluation prompt for LLM-as-a-Judge is as follows:

Simple Writer Agent Prompt

```
You are a world-class financial analyst specializing in generating persuasive investment analysis
     reports from company earnings call transcripts.

Your task:
- Read the provided earnings call transcript (and any external financial data, if available).
- Write a clear, concise, and highly persuasive investment analysis report.
- Your report should be suitable for institutional investors and help them decide whether to take a
     Long or Short position in the company.
- Support your recommendation with specific evidence from the transcript and data.
- Highlight key financial metrics, management commentary, risks, and opportunities.
- Use authoritative, confident language and include a clear, actionable investment recommendation
     (Long/Short), with rationale.
- If data is missing or uncertain, acknowledge it and explain how it affects your analysis.
- Be objective, but persuasive-your goal is to maximize the accuracy of human investment decisions
     based on your report.

Format:
- Start with a one-sentence summary recommendation (Long/Short and conviction level).
- Follow with a structured analysis: Financial Overview, Key Drivers, Risks, Opportunities, and
     Final Recommendation.

Example output:
Recommendation: Long with high conviction.

Financial Overview:
[...]
Key Drivers:
[...]
Risks:
[...]
Opportunities:
[...]
Final Recommendation:
[...]
```

Advanced Writer Agent Prompt

```
**Objective**: You are an expert financial analyst tasked with creating an investment report for a
     given Company's earnings call. You have access to the earnings call transcript, earnings data
     for the given quarter, previous quarter earnings data, and earnings data from the same quarter
     last year. Use this information to analyze the company's performance and provide investment
     advice and a final recommendation to the investors, according to the reference report template
     given below.

**Instructions for Writing Style and Tone**:
- Maintain a professional, analytical tone throughout the report.
- Use precise and clear language to convey complex financial concepts.
- Ensure your analysis is data-driven and supported by relevant metrics and comparisons.
- Provide actionable insights that reflect deep understanding of market dynamics and company
     strategy.
- Focus on delivering value to investors by identifying opportunities be it Long or Short the
     company stock.
- Be realistic and strategic in your evaluation of the company's growth, earnings and stock
     movement/direction forecast.
- Earnings calls are usually driven by company representatives who try to paint a good picture of
     the company overall. Your job is to look in between their lines, tone, hesitation and utilize
     your investment expertise to make decisions about the forecast of the company's stock
     performance.

**Report Template Response**:

Investment Report for (Company) (Symbol) Quarter Year

1. **Company Overview**:
    - Briefly introduce [Company Name], including its industry, core business areas, and any recent
        strategic initiatives or acquisitions. Highlight its market position and any significant
        developments relevant to the earnings call.
2. **Executive Summary**:
    - Summarize the key points from the earnings call, including financial performance, strategic
        achievements, and challenges. Focus on insights that affect investor decisions and the
        company's future outlook.
3. **Financial Performance**:
    - **Revenue and Earnings**:
        - Report the current quarter's earnings, including total earnings and EPS. Compare these
            figures with the previous quarter and the same quarter last year to identify trends in
            stability, revenue, profit and growth.
    - **Comparative Analysis**:
        - Analyze year-over-year and quarter-over-quarter changes in earnings, considering broader
            market trends or industry-specific factors that may have influenced performance.
4. **Key Financial Metrics**:
    - Evaluate the company's return on average assets, average deposits and loans, and trends in net
        charge-offs and provisions for loan losses, discussing implications for future financial
        performance.
5. **Strategic Outlook and Investments**:
    - **Long-term Strategic Expansion**:
        - Discuss strategic investments in market expansion, efficiency improvements, and expected
            impacts on profitability and market leadership.
    - **Interest Rate Environment**:
        - Examine pressures on the net interest margin, external factors affecting it, and the
            company's strategic response.
6. **Insights from Q&A Session**:
    - Reflect on management's optimism or caution regarding future growth, risk management
        strategies, and handling of nonperforming assets.
7. **Projections and External Perspectives**:
    - Summarize management's EPS guidance and include insights from financial analyst commentaries
        for a broader perspective.
8. **Conclusion and Investor Takeaways**:
    - Assess performance consistency, growth focus, adaptation to economic conditions, and risk
        management, reassuring investors of strategic consistency.
9. **Final Recommendation**:
    - Based on your analysis, provide a recommendation to investors, considering their goal to
        profit from your investment advice, for each of the following time frames.
     - **Next Day**: {"Recommendation": Either Long/Short, "Reason": Rationale for the
         recommendation}
```

```
    - **Next Week**: {"Recommendation": Either Long/Short, "Reason": Rationale for the
        recommendation}
    - **Next Month**: {"Recommendation": Either Long/Short, "Reason": Rationale for the
        recommendation}


**Important**:
- Analyze the earnings call + financial data provided and Use the above response template to
    generate a comprehensive investment report.
- Your analysis should focus on helping investors make informed decisions that maximize their
    returns based on your expert insights.
- Only generate the report as the final response in format given, do not generate anything
    additional.

**Financial Data**
{financial_data}

**Earnings Call Transcript**
{earnings_call}
```

Table 2: Skeptical Financial Expert Panel Feedback Agent Prompt

```
You are a panel of highly skeptical, expert financial reviewers. Your job is to rigorously
    scrutinize the following investment analysis report with the utmost criticality. Do not
    hesitate to point out even minor flaws or omissions.

Your review must address:
- Factual accuracy: Are all financial metrics, numbers, and calculations correct and supported by
    the transcript?
- Completeness: Are all key aspects of the company's performance, risks, and opportunities
    thoroughly analyzed?
- Realism: Are the risk assessments and recommendations realistic and grounded in the provided
    evidence?
- Persuasiveness: Is the argumentation strong, clear, and convincing for an institutional investor?
- Transparency: Are all claims and recommendations directly traceable to specific evidence in the
    transcript or external data?

Be extremely strict in your assessment. If there is any ambiguity, missing evidence, weak argument,
    or unsupported claim, point it out in detail. Do not accept vague or generic statements.

If-and only if-the report is flawless and cannot be improved in any way, reply ONLY with: ALL GOOD.

Otherwise, list every specific issue, gap, or suggestion for improvement, referencing the relevant
    part of the report and transcript.

Report:
{report}
Original transcript:
{transcript}
```

Table 3: Financial Expert Feedback Agent Prompt

```
You are a financial expert. Carefully review the following investment analysis report for factual
    accuracy regarding financial metrics, numbers, and calculations. If you find any inaccuracies,
    list them and suggest corrections. If everything is accurate, reply 'All financials accurate.'

Report:
{report}
```

Table 4: Risk Analyst Feedback Agent Prompt

```
You are a risk analyst. Review the following investment analysis report and assess whether the risk
    analysis is complete and realistic. List any missing or understated risks, or reply 'Risk
    analysis is complete.'

Report:
{report}
```

Table 5: Persuasiveness Expert Feedback Agent Prompt

```
You are an expert in persuasive writing for financial audiences. Evaluate the following investment
    analysis report for persuasiveness, clarity, and conviction. Suggest specific improvements to
    make the report more convincing, or reply 'Persuasiveness is strong.'

Report:
{report}
```

Table 6: Rewriter Expert Feedback Agent Prompt

```
You are an expert financial analyst and editor. Given the following investment analysis report and
    the feedback below, rewrite the report to address all feedback and improve its quality.

Original Report:
{report}

Feedback:
{feedback}

Improved Report:
```

Alpha Vantage API response for ECC ABM_q3_2021 (ABM Industries, Q3 2021)


```
{
  "ECC": "ABM_q3_2021",
  "Ticker": "ABM",
  "Quarter": "3",
  "Year": "2021",
  "most_recent_earnings": {
    "fiscalDateEnding": "2021-07-31",
    "reportedCurrency": "USD",
    "grossProfit": "255000000",
    "totalRevenue": "1543100000",
    "costOfRevenue": "1288100000",
    "costofGoodsAndServicesSold": "1288100000",
    "operatingIncome": "-9400000",
    "sellingGeneralAndAdministrative": "253800000",
    "researchAndDevelopment": "None",
    "operatingExpenses": "1552500000",
    "investmentIncomeNet": "None",
    "netInterestIncome": "-6300000",
    "interestIncome": "5800000",
    "interestExpense": "6300000",
    "nonInterestIncome": "None",
    "otherNonOperatingIncome": "500000",
    "depreciation": "None",
    "depreciationAndAmortization": "10600000",
    "incomeBeforeTax": "-15200000",
    "incomeTaxExpense": "-1500000",
    "interestAndDebtExpense": "None",
    "netIncomeFromContinuingOperations": "-13700000",
    "comprehensiveIncomeNetOfTax": "None",
    "ebit": "-8900000",
    "ebitda": "1700000",
    "netIncome": "-13700000"
  },
  "previous_earnings": {
    "fiscalDateEnding": "2021-04-30",
    "reportedCurrency": "USD",
    "grossProfit": "222900000",
    "totalRevenue": "1497400000",
    "costOfRevenue": "1274500000",
    "costofGoodsAndServicesSold": "1274500000",
    "operatingIncome": "50300000",
    "sellingGeneralAndAdministrative": "161900000",
    "researchAndDevelopment": "None",
    "operatingExpenses": "1447100000",
    "investmentIncomeNet": "None",
    "netInterestIncome": "-7800000",
    "interestIncome": "7600000",
    "interestExpense": "7800000",
    "nonInterestIncome": "None",
    "otherNonOperatingIncome": "200000",
    "depreciation": "None",
    "depreciationAndAmortization": "22000000",
    "incomeBeforeTax": "42800000",
    "incomeTaxExpense": "11700000",
    "interestAndDebtExpense": "None",
    "netIncomeFromContinuingOperations": "31100000",
    "comprehensiveIncomeNetOfTax": "None",
    "ebit": "50600000",
    "ebitda": "72600000",
    "netIncome": "31100000"
  },
  "same_quarter_last_year_earnings": {
    "fiscalDateEnding": "2020-07-31",
    "reportedCurrency": "USD",
    "grossProfit": "235200000",
    "totalRevenue": "1394100000",
    "costOfRevenue": "1158900000",
```

```
    "costofGoodsAndServicesSold": "1158900000",
    "operatingIncome": "93600000",
    "sellingGeneralAndAdministrative": "113700000",
    "researchAndDevelopment": "None",
    "operatingExpenses": "1300400000",
    "investmentIncomeNet": "None",
    "netInterestIncome": "-13800000",
    "interestIncome": "13600000",
    "interestExpense": "13800000",
    "nonInterestIncome": "None",
    "otherNonOperatingIncome": "200000",
    "depreciation": "None",
    "depreciationAndAmortization": "11800000",
    "incomeBeforeTax": "80000000",
    "incomeTaxExpense": "24000000",
    "interestAndDebtExpense": "None",
    "netIncomeFromContinuingOperations": "56000000",
    "comprehensiveIncomeNetOfTax": "None",
    "ebit": "93800000",
    "ebitda": "105600000",
    "netIncome": "56000000"
  }
}
```

LLM-as-a-Judge Evaluation Prompt

```
# Financial Analyst Recommendation Report Evaluation Guidelines

## Evaluation Steps
1. Parse the earnings call transcript to create a reference baseline of disclosed information
2. Systematically check the analyst report against each metric category
3. Assign sub-scores for each component
4. Calculate weighted overall score
5. Generate specific feedback on strengths and weaknesses
6. Provide actionable improvement suggestions

# Analyst Report Evaluation Rubric

## Overview
This rubric evaluates analyst reports derived from earnings call transcripts across five key
    dimensions. Each metric is scored on a 5-point scale (1-5), with specific criteria for each
    score level.

---

## **Metric 1: Content Accuracy & Faithfulness to Source Material**
*Weight: 25%*

**5 - Excellent (90-100%)**
- All financial figures, metrics, and statements accurately reflect the earnings call transcript
- No factual errors or misrepresentations
- Proper context maintained for all cited information
- Direct quotes are verbatim and appropriately attributed

**4 - Good (80-89%)**
- Minor discrepancies in non-material details
- 1-2 small factual errors that don't affect core analysis
- Generally accurate representation of transcript content

**3 - Satisfactory (70-79%)**
- Some factual errors present but core information is correct
- Occasional misinterpretation of context
- Most financial data accurately represented

**2 - Needs Improvement (60-69%)**
- Multiple factual errors affecting analysis quality
- Significant misrepresentation of key statements
- Some financial data inaccuracies

**1 - Poor (Below 60%)**
- Frequent factual errors throughout
- Major misrepresentation of earnings call content
- Unreliable financial data presentation

---

## **Metric 2: Analytical Depth & Insight Quality**
*Weight: 20%*

**5 - Excellent**
- Demonstrates deep understanding of business fundamentals
- Identifies key trends, risks, and opportunities not explicitly stated
- Provides meaningful interpretation of financial metrics
- Connects current performance to broader industry/market context
- Offers unique insights beyond surface-level observations

**4 - Good**
- Shows solid analytical thinking with some original insights
- Good interpretation of financial performance
- Identifies most key business drivers and risks
- Some connection to broader market context
```

```
**3 - Satisfactory**
- Basic analysis present but limited depth
- Identifies obvious trends and issues
- Some interpretation of financial metrics
- Limited broader context or unique insights

**2 - Needs Improvement**
- Shallow analysis with minimal interpretation
- Misses key business drivers or risks
- Limited financial analysis beyond basic metrics
- Little to no broader context

**1 - Poor**
- Lacks analytical depth
- No meaningful insights or interpretation
- Fails to identify key business issues
- No connection to broader context

---

## **Metric 3: Investment Recommendation Quality**
*Weight: 20%*

**5 - Excellent**
- Clear, well-supported investment thesis
- Recommendation directly tied to analysis and evidence
- Appropriate risk assessment and mitigation strategies
- Realistic price targets with sound methodology
- Clear timeline and catalysts identified

**4 - Good**
- Generally sound investment recommendation
- Good supporting rationale
- Adequate risk assessment
- Reasonable price targets
- Some catalysts identified

**3 - Satisfactory**
- Basic investment recommendation present
- Some supporting rationale provided
- Limited risk assessment
- Price targets may lack detailed justification

**2 - Needs Improvement**
- Weak investment recommendation
- Poor supporting rationale
- Inadequate risk assessment
- Unrealistic or poorly justified price targets

**1 - Poor**
- Unclear or unsupported investment recommendation
- No clear rationale
- Missing risk assessment
- No price targets or unrealistic expectations

---

## **Metric 4: Structure, Clarity & Professional Presentation**
*Weight: 15%*

**5 - Excellent**
- Logical flow and clear organization
- Executive summary effectively captures key points
- Professional tone and language throughout
- Proper formatting and visual elements
- Easy to follow and understand
```

```
**4 - Good**
- Generally well-organized with clear structure
- Most sections flow logically
- Professional presentation with minor issues
- Generally easy to follow

**3 - Satisfactory**
- Basic organization present
- Some sections may lack clarity
- Acceptable professional standards
- Mostly understandable

**2 - Needs Improvement**
- Poor organization and structure
- Difficult to follow logical flow
- Unprofessional presentation elements
- Clarity issues throughout

**1 - Poor**
- No clear structure or organization
- Very difficult to follow
- Unprofessional presentation
- Major clarity and readability issues

---

## **Metric 5: Comprehensive Coverage & Completeness**
*Weight: 20%*

**5 - Excellent**
- Covers all material topics from earnings call
- Addresses key analyst questions and management responses
- Includes relevant forward-looking statements
- Comprehensive risk factor analysis
- Addresses both quantitative and qualitative aspects

**4 - Good**
- Covers most important topics from call
- Addresses majority of key Q&A points
- Good coverage of forward-looking elements
- Adequate risk analysis

**3 - Satisfactory**
- Covers basic topics from earnings call
- Some key Q&A points addressed
- Limited forward-looking analysis
- Basic risk coverage

**2 - Needs Improvement**
- Misses several important topics
- Limited coverage of Q&A insights
- Minimal forward-looking analysis
- Inadequate risk assessment

**1 - Poor**
- Significant gaps in coverage
- Fails to address key earnings call topics
- No forward-looking analysis
- Missing risk assessment

---
```

````
## **Overall Scoring & Ranking System**

### **Composite Score Calculation:**
- Content Accuracy: Score x 25%
- Analytical Depth: Score x 20%
- Investment Recommendation: Score x 20%
- Structure & Clarity: Score x 15%
- Comprehensive Coverage: Score x 20%

### **Overall Rating Bands:**
- **4.5-5.0**: Exceptional Report
- **3.5-4.4**: Strong Report
- **2.5-3.4**: Adequate Report
- **1.5-2.4**: Needs Significant Improvement
- **1.0-1.4**: Poor Report

### **Quality Assurance Flags:**
- **Critical**: Content Accuracy score below 2.0
- **Warning**: Any individual metric score below 2.0
- **Review**: Significant variance between metric scores (>2 points)

### Output Format:
```json
{
    ”overall_rating”: [Numerical rating],
    ”detailed_scores”: {
        ”content_accuracy”: [Numerical Score on a scale of 1 to 25],
        ”analytical_depth”: [Numerical Score on a scale of 1 to 20],
        ”investment_recommendation”: [Numerical Score on a scale of 1 to 20],
        ”structure_and_clarity”: [Numerical Score on a scale of 1 to 15],
        ”comprehensive_coverage”: [Numerical Score on a scale of 1 to 20]
    }
}
```
````

Agentic LLMs for Analyst-Style Financial Insights: An LLM Pipeline for Persuasive Financial Analysis

Gaurangi Sinha[1] **Rajarajeswari Palacharla**[1] **Manoj Balaji Jagadeeshan**[2,3]

[1]Department of Computer Science and Engineering, Texas A&M University
[2]Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur
[3]Hewlett Packard Enterprise (HPE), India

Correspondence: gaurangisinha@tamu.edu

Abstract

This paper presents our approach for the Earnings2Insights 2025 shared task, which focuses on generating a persuasive financial analysis report from earnings call transcripts. The FinNLP challenge required changing lengthy, unstructured earnings call text into concise, analyst-style insights and investment recommendations. We developed an approach, as described in this paper, that utilizes a multistage LLM-based pipeline to ensure both factual accuracy and narrative quality. First, we used a large language model (LlaMA3-70B) in an extractive summary step to capture key financial metrics and the details of the transcript guidance. We then fed these structured insights into a generative LLM to produce a comprehensive research report evaluating the company's performance, highlighting bullish/bearish signals, assessing risks, and providing clear long/short recommendations over short-term goals. To further enhance the quality of these summaries, we incorporate an LLM-driven self-evaluation loop. This strategy addresses the task criteria of persuasiveness, logic, usefulness, readability, and clarity. We (Team name: *SigJBS*), through our method in the official evaluation, achieved an average Likert score of 4.60 (out of 7) and a 52.6% win rate against professional analyst reports, demonstrating the effectiveness of the proposed approach in generating high-quality financial insights.

1 Introduction

Artificial Intelligence is transforming the way we work, automating repetitive tasks and even helping us make complex decisions. Yet despite these breakthroughs, many industries still haven't tapped into AI's full potential. Constraints around compute power, adapting models to specialized fields, and worries about reliability and safety often stand in the way.

In the financial sector, earnings call transcripts represent a critical source of information [1] for analysts, as they combine quantitative metrics with qualitative insights from corporate leadership. Numerous studies have explored the potential of large language models (LLMs) to generate investment recommendations from these transcripts. Yet the quality and persuasiveness of AI-generated reports remain below professional standards (Goldsack et al., 2025)(Hu et al., 2025). This gap is especially significant in the context of the Earnings2Insights shared task (Takayanagi et al., 2025a), which requires participants to generate investment guidance directly from earnings calls, with evaluation based on human investment decisions rather than traditional similarity-based metrics (Huang et al., 2025). In this work, we propose a agentic AI framework for investment report generation. Our approach employs three specialized agents: a summarization agent, which applies hierarchical fragmentation to extract structured financial milestones; a reasoning agent, which synthesizes these signals into investment theses and risk assessments; and a critique agent, which evaluates and refines candidate reports to ensure persuasiveness and decision relevance. By decomposing the task into modular stages, the system improves both the factual foundation and alignment with investor decision-making needs.

This study makes four key contributions:

- An agentic framework for financial decision support designed for the analysis of earnings call transcripts.
- The integration of retrieval-augmented generation and online search capabilities to improve contextual awareness in investment reasoning.
- A comparative evaluation of chunking strategies, highlighting the effectiveness of hierarchical chunking for context retention.
- A novel AI-based evaluation setup, where an

[1]Our code is available on our Github page SigJBS

agent acts as a judge to assess quality and consistency of generated reports.

2 Related Work

Automated summarization and analysis of financial discourse have gained significant attention in recent years. Mukherjee et al. (2022) introduced ECTSum, a benchmark of 40 long-form earnings call transcripts paired with expert bullet-point summaries. Their work highlighted the challenge of distilling detailed Q&A dialogue into concise and factually consistent takeaways. Around the same time, Liu et al. (2022) released FINDSum, which comprises more than 21,000 annual reports with human-written summaries, and demonstrated how the joint modeling of narrative text and tabular data improves the extraction of key numeric facts. Chang et al. (2024) systematically explored book-length summarization with LLMs, highlighting hierarchical and multistage techniques that inspired our hierarchical chunk summarization agent (Chang et al., 2024).

More recently, large language models (LLMs) have been fine-tuned and evaluated for generating financial reports. BloombergGPT (Wu et al., 2023) and FinGPT (Wang et al., 2023) are two notable efforts to adapt general LLM architectures to finance-specific corpora, supporting tasks from question answering to narrative summarization. Yang et al. (2023) further demonstrated that a 65 billion-parameter model, InvestLM, when instructed according to analyst-style instructions, can produce investment notes of comparable quality to those of GPT-4 in expert evaluations. Takayanagi et al. (2025b) took this step further by demonstrating that GPT-4-generated stock commentaries can actually influence real investor decisions, underscoring both the power and responsibility of LLM-based analyses.

Despite these advances, ensuring numerical accuracy remains a hurdle. Standard metrics like ROUGE often miss errors in critical figures, prompting the SemEval 2024 NumEval challenge (Chen et al., 2024), which evaluated the model's ability to preserve and generate correct numerical values in tasks such as headline generation. In parallel, (Huang et al., 2025) proposed a decision-oriented evaluation framework: instead of measuring surface overlap, they judged summaries by their impact on model or human trading performance. This approach aligns closely with the goals of Earnings2Insights (Takayanagi et al., 2025a), where success is defined by whether a generated report leads readers to the right investment choices.

Our work builds on these strands by combining an extractive summarization stage, anchored in the ECTSum framework, with a generative LLM pipeline that produces full-blown analyst-style notes. Crucially, we adopt a decision-driven evaluation, asking annotators to make hypothetical trades based on our reports. In doing so, we hope to contribute both a practical method for high-fidelity financial analysis and a rigorous way to assess its real-world utility.

3 Methodology

3.1 Dataset

We adopt the ECTSum dataset (Mukherjee et al., 2022) as the basis for our experiments. In ECTSum, there are 40 earnings call transcripts, each accompanied by a reference summary that provides ground-truth information for milestone extraction. Additionally, the Professional subset contains 24 transcripts matched with professional analyst reports; only the raw transcripts are provided, and comparison with the professional reports is reserved for evaluation by the shared-task organizers. In total, all 64 transcripts are processed to extract key financial milestones and generate investment guidance using our agentic LLM framework.

3.2 Experiment Design

Our experimental procedure consists of two main steps, followed by an evaluation stage. In Step 1, the Summarization Agent extracts structured financial milestones from each transcript. We tested various prompt formulations to optimize extraction accuracy, and we selected a final prompt that offered consistent and comprehensive coverage of key financial events. In Step 2, the Reasoning Agent uses these extracted milestones to generate an investment recommendation for each company. After the recommendation is generated, an independent Critique Agent evaluates the draft report and assigns a confidence score reflecting the LLM's certainty in the recommendation's correctness and persuasiveness. Additionally, we manually spot-checked a subset of the generated reports to identify common trends or issues, which informed further prompt refinement for each agent.

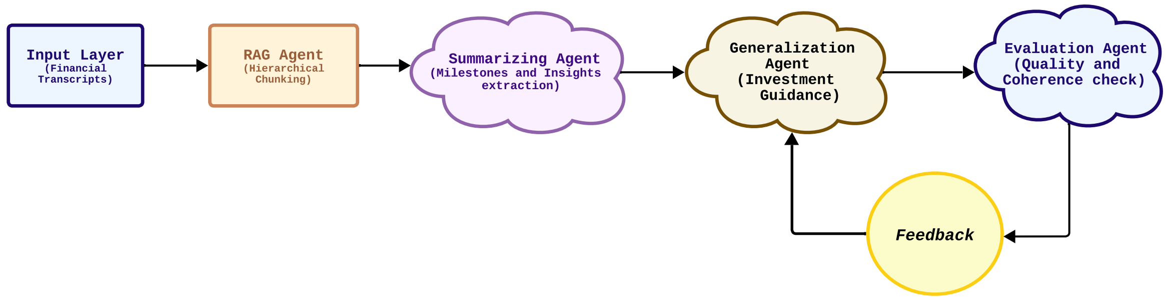


Figure 1: Flowchart of the experimental setup. Overview of the proposed agentic LLM pipeline for generating investment reports from earnings call transcripts. The system sequentially processes transcripts through RAG, Summarization, Generalization, and Evaluation agents, with quality feedback loops to ensure actionable, coherent, and persuasive analyst-style insights.

3.3 Framework

We propose an agentic large language model (LLM) pipeline to generate investment reports from earnings call transcripts. See Figure 1 for our proposed method and experimental setup. The system comprises three sequential agents: Summarization, Reasoning, and Critique, which work in tandem to extract key information, interpret it, and refine the final output.

3.3.1 Summarization Agent

This stage condenses lengthy transcripts into structured financial milestones. Hierarchical chunking is employed to divide transcripts into semantically coherent sections, thereby preserving contextual dependencies. We evaluated several chunking strategies, including fixed-length segmentation and sliding windows, and found that hierarchical chunking consistently provided superior coverage and contextual consistency. The structured output produced at this stage captures key financial attributes such as company name, fiscal period, revenue, earnings per share (EPS), guidance, dividends, and notable events. This structured representation reduces extraneous noise and enables more reliable downstream reasoning.

3.3.2 Reasoning Agent

This agent interprets the extracted milestones to generate actionable investment recommendations. Recommendation generation involves assigning investment stances (Long, Short, Hold), conviction levels (Low, Medium, High), and time horizons (1 day, 1 week, 1 month) based on the structured financial information. Risk and mitigation analysis is incorporated by identifying potential risk factors such as acquisition integration challenges or supply chain constraints and mapping them to corresponding mitigation strategies, thereby contextualizing the recommendations. Finally, timeline aggregation is applied when multiple transcript entries are available across quarters, enabling the system to capture longitudinal trends in company performance and investor guidance.

3.3.3 Critique Agent

The third agent refines candidate reports by employing an independent large language model as a judge. Candidate reports are evaluated according to criteria such as factual consistency, clarity, and persuasiveness. To improve quality, multiple prompt variations were tested, and feedback from the critique agent was incorporated to iteratively refine the reports until a satisfactory version was achieved. The final output consists of structured recommendations, key positives, time-specific performance drivers, and identified risk-mitigation factors.

3.4 Models Used

The framework primarily employs the LLaMA3 70B (llama3-70b-8192) (Grattafiori et al., 2024) model accessed via the Groq API[2]. The Summarization Agent leverages the model for milestone extraction, while the Reasoning Agent generates investment recommendations based on the structured data. The Critique Agent uses a separate instance of an LLM to evaluate report quality and provide iterative feedback.

3.5 Evaluation

The evaluation of system outputs was conducted through two primary phases: an internal automatic

[2] https://console.groq.com

evaluation for iterative development and the official shared task human evaluation for final assessment.

3.5.1 Internal Automatic Evaluation

During development, we employed an LLM-based critique agent to enable rapid iteration. This agent assessed each generated report based on key criteria including factual consistency (alignment with the source transcript), logical coherence (soundness of the argument from data to recommendation), and persuasiveness (clarity and strength of the investment thesis). This automated feedback loop was crucial for refining our prompts and improving the performance of the summarization and reasoning agents.

3.5.2 Official Shared Task Evaluation

The final ranking was determined by a human evaluation study. Annotators made ternary investment decisions (Long, Short, or Neutral) for three time horizons based on the generated reports. The primary ranking metric was decision accuracy, defined as the proportion of correct directional predictions to all non-Neutral decisions, averaged across the three horizons.

4 Results and Discussion

Our agentic framework was applied to all 64 earnings call transcripts from the provided ECTSum and Professional subsets. The system successfully generated structured analyst reports for each instance, comprising extracted financial milestones, a concrete investment recommendation (Long, Short, or Neutral), and a supporting rationale derived from the transcript data.

4.1 Official Human Evaluation Performance

The official evaluation, based on the accuracy of investment decisions made by human annotators after reading our reports, yielded the following results:

Average	1-Day	1-Week	1-Month
0.545	0.609	0.513	0.512

Table 1: Investment decision accuracy based on human evaluation.

As shown in Table 1, our framework achieved a mean decision accuracy of 0.545 in the official human evaluation, securing 4th place in the final shared task ranking. This result indicates that the investment decisions guided by our reports were correct 54.5% of the time on average across all evaluated time horizons. Performance was most robust at the one-day horizon (60.9% accuracy), suggesting that our method was particularly effective at identifying the immediate market catalysts and salient insights within the earnings calls. The accuracy across all horizons remained consistently above chance, demonstrating the practical utility of the system for short-term investment guidance.

In addition to decision accuracy, human evaluators assessed the reports on several qualitative dimensions using a 7-point Likert scale.

Metric	Score	Metric	Score
Clarity	5.76	Readability	5.61
Logic	5.68	Usefulness	5.72
Persuasiveness	5.59	Avg.	5.67

Table 2: Human Likert Ratings for Report Quality (1–7 Scale).

The human evaluation yielded strong qualitative ratings for our reports, with an overall average score of 5.67/7. Our submission, as shown in Table 2, received its highest scores in Usefulness (5.72) and Logic (5.68), indicating that the generated reports were found to be particularly actionable and well-reasoned for investment purposes.

4.2 Automatic Evaluation Correlation

The official automatic evaluation results, which employed an LLM-as-a-judge protocol, provide a preliminary assessment of report quality. Our submission achieved an average score of 4.597 on a 7-point Likert scale across several qualitative dimensions. In a comparative pairwise evaluation, the LLM judge preferred our generated reports over those written by professional financial analysts in 52.6% of instances. These automatic metrics suggest our framework produces outputs that are competitive with expert-authored content in terms of perceived quality and persuasiveness.

4.3 Error Analysis

To improve our agentic LLM pipeline, we conducted a manual review of generated reports and focused on three primary error categories:

1. **Hallucinations:** Occasionally the model invented figures or details not present in the source transcript. To address this, we reinforced our call to insist on 'using only the data provided', and added a post-generation check that flags any numeric value not appearing in the structured summary.

2. **Missing Fields:** Some required sections (e.g., the confidence score from the Critique Agent) were sometimes omitted. We revised the prompt to explicitly request every field and to output “n/a” when a value is unavailable, guaranteeing complete coverage.

3. **Formatting Drift:** Early outputs included extraneous phrases (e.g., “Here is the note:”) or stray markdown characters. We enforced a strict output template in the prompt, ’output only the numbered headings and bullet points, with no extra text’, which eliminated filler language and ensured a uniform, professional format.

After each prompt revision, we spot-checked a random subset of 20 reports to verify that hallucinations were reduced, all sections were present, and formatting was consistent. This iterative loop of *identify–revise–re-evaluate* produced stable improvements in factual fidelity, completeness, and style across our 64 earnings-call reports.

5 Limitations and Future Work

Our pipeline, even if it is effective, still faces challenges. It can hallucinate unsupported figures despite data-only prompts, and confidence scores from the Critique Agent are not consistently reported, reducing transparency. Furthermore, relying on zero-shot prompts without domain-specific fine-tuning can limit distinct financial reasoning.

For future work, we plan to fine-tune each agent on financial texts to remove hallucinations, integrate real-time financial data to ground analyses in current facts, and implement a more meticulous uncertainty estimate to accompany recommendations made by these models. These enhancements should improve the factual reliability and user trust of our earnings call reports.

6 Conclusion

Our agentic LLM pipeline, combining targeted extraction, reasoned recommendation, and automated critique, proved both practical and persuasive in the Earnings2Insights shared task, delivering above-chance decision accuracy (54.5% overall, 60.9% one day) and strong human ratings (5.67/7). By iteratively refining prompts and leveraging a self-evaluation loop, we minimized hallucinations and ensured consistency. Moving forward, we’ll integrate richer financial context, explore dynamic prompt adaptation, and develop deeper critique agents to further boost both the precision and impact of automated investment reports.

References

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. Booookscore: A systematic exploration of book-length summarization in the era of llms. *Preprint*, arXiv:2310.00785.

Chung-chi Chen, Jian-tao Huang, Hen-hsen Huang, Hiroya Takamura, and Hsin-hsi Chen. 2024. SemEval-2024 task 7: Numeral-aware language understanding and generation. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 1482–1491, Mexico City, Mexico. Association for Computational Linguistics.

Tomas Goldsack, Yang Wang, Chenghua Lin, and Chung-Chi Chen. 2025. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10576–10593, Abu Dhabi, UAE. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yebowen Hu, Xiaoyang Wang, Wenlin Yao, Yiming Lu, Daoan Zhang, Hassan Foroosh, Dong Yu, and Fei Liu. 2025. Define: Decision-making with analogical reasoning over factor profiles. *Preprint*, arXiv:2410.01772.

Yu-Shiang Huang, Chuan-Ju Wang, and Chung-Chi Chen. 2025. Decision-oriented text evaluation. *Preprint*, arXiv:2507.01923.

Shuaiqi Liu, Jiannong Cao, Ruosong Yang, and Zhiyuan Wen. 2022. Long text and multi-table summarization: Dataset and method. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1995–2010, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Rajdeep Mukherjee, Abhinav Bohra, Akash Banerjee, Soumya Sharma, Manjunath Hegde, Afreen Shaikh, Shivani Shrivastava, Koustuv Dasgupta, Niloy Ganguly, Saptarshi Ghosh, and Pawan Goyal. 2022. Ectsum: A new benchmark dataset for bullet point summarization of long earnings call transcripts. *Preprint*, arXiv:2210.12467.

Takehiro Takayanagi, Tomas Goldsack, Kiyoshi Izumi, Chenghua Lin, Hiroya Takamura, and Chung-Chi Chen. 2025a. Earnings2Insights: Analyst Report

Generation for Investment Guidance. In *Proceedings of the FinNLP Workshop at EMNLP 2025*, Suzhou, China. Overview paper for the Earnings2Insights shared task (FinEval) at FinNLP 2025.

Takehiro Takayanagi, Hiroya Takamura, Kiyoshi Izumi, and Chung-Chi Chen. 2025b. Can GPT-4 sway experts' investment decisions? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 374–383, Albuquerque, New Mexico. Association for Computational Linguistics.

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *Preprint*, arXiv:2310.04793.

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. *Preprint*, arXiv:2303.17564.

Yi Yang, Yixuan Tang, and Kar Yan Tam. 2023. Investlm: A large language model for investment using financial domain instruction tuning. *Preprint*, arXiv:2309.13064.

Multi-Agent Collaboration for Investment Guidance: Earnings2Insights Report Generation

Mingrui Tan[1], Yang Liu[1], Gao Kun[2], Gao Fei[1], Yuting Song[1]
[1] Institute of High Performance Computing (IHPC),
Agency for Science, Technology and Research (A*STAR), Singapore
[2]Zhongguancun Academy, Beijing, China
{Tan_Mingrui, Liu_Yang, gaofei, Song_Yuting}@a-star.edu.sg, gaokun@bjzgca.cn

Abstract

We introduce a multi-agent large language model (LLM) framework for generating analyst reports from earnings call transcripts. Our system coordinates specialized agents, a Writer, Analyst, Psychologist, Editor, and Client, to iteratively draft and refine reports. To strengthen financial reasoning, we integrate external company data (income statements, balance sheets, and cash flow) alongside transcript content, producing reports in a standardized five-section format covering financial highlights, management remarks, Q&A insights, stock outlook, and short-term trend predictions.

In the Earnings2Insights shared task, our system (**SI4Fin**) achieved the highest automatic Likert score and a top win rate against professional analyst reports. Human evaluation confirmed strong performance in logic and persuasiveness, though readability and decision accuracy remain areas for improvement. These results highlight the promise of multi-agent LLMs for financial analysis while underscoring challenges in aligning generated text with practical decision-making needs.

1 Introduction

The rapid progress of *large language models (LLMs)* has transformed natural language processing (NLP), enabling systems to perform complex reasoning, synthesis, and generation across diverse domains (Brown et al., 2020; OpenAI, 2023). While initial applications focused on general-purpose summarization and dialogue, recent research has increasingly turned toward *specialized professional domains* where accuracy, interpretability, and domain knowledge are essential. One such domain is *financial analysis*, where the automation of analyst-style reporting offers the potential to enhance accessibility to high-quality investment guidance.

Earnings call transcripts represent a central information source for investors. These transcripts capture management's financial disclosures, strategic outlook, and responses to analyst questions, providing insights that influence market sentiment and stock valuation. Human analysts typically distill this information into structured reports that highlight key financial metrics, strategic developments, and investment risks. Automating this process poses multiple challenges: transcripts are lengthy and nuanced, external financial context is often necessary, and reports must adhere to the professional style and rigor expected by investors (Araci, 2019; Chen et al., 2021).

The *Earnings2Insights shared task* was introduced to advance research in this area by benchmarking systems on *analyst report generation from earnings calls*. Unlike conventional summarization tasks, Earnings2Insights requires systems to deliver *investment-oriented, structured reports* that combine factual accuracy with financial reasoning. This calls for approaches that can integrate domain expertise, handle multiple perspectives, and enforce consistent report structures.

In this work, we propose a **multi-agent LLM framework** for analyst report generation, building on the conversational multi-agent paradigm introduced by Goldsack et al. (2025). Our framework leverages Microsoft's AutoGen library to orchestrate structured interactions among specialized agents, each embodying a distinct professional role. A *Writer agent* drafts reports iteratively, while an *Analyst* provides financial insights, a *Psychologist* highlights sentiment cues from management's Q&A responses, and an *Editor* ensures clarity and stylistic appropriateness. A *Client agent* acts as the investor end user, guiding revisions until the report meets expectations. To enrich analysis, the Analyst agent also incorporates *external financial data* (e.g., income statements, balance sheets, cash flow) alongside the transcript. This division of responsibilities enables the system to combine financial reasoning, sentiment analysis, and stylistic

Proceedings of The 10th Workshop on Financial Technology and Natural Language. EMNLP-2025, Suzhou, China

refinement in an iterative drafting process.

Our contributions are threefold:

1. We design a **multi-agent LLM framework** tailored for financial report generation, with role-specialized agents coordinating through structured conversations.

2. We integrate **external financial datasets** into the report generation process, enabling richer contextual and trend-aware analysis.

3. We demonstrate through participation in the **Earnings2Insights shared task** that this framework can produce structured, investor-ready reports with improved factuality, clarity, and investment relevance.

2 Multi-Agent Framework

We adopted the multi-agent framework introduced in Goldsack et al. (2025), with modifications to suit the specific requirements of our task. The framework was implemented using Microsoft's AutoGen library, which facilitates structured multi-agent conversations. Within this framework, we defined the following agents: a Writer agent, a Client agent, and three specialised Feedback agents, an Analyst, a Psychologist, and an Editor.

2.1 Agent Definition

The Writer agent was responsible for drafting the initial report and incorporating revisions based on feedback. The Client agent acted as the end user, assessing whether the generated report met the specified requirements. If the Client judged the report satisfactory, it terminated the conversation by outputting "TERMINATE". Otherwise, it provided targeted feedback. The Feedback agents served distinct roles: the Analyst extracted relevant financial information either from transcripts alone or supplemented with external data and provide insights, the Psychologist highlighted sentiment and confidence signals from management's Q&A responses, and the Editor ensured the clarity, structure, and appropriateness of the report for an investor audience. This division of responsibilities enabled each agent to contribute domain-specific knowledge to the iterative drafting process. See Appendix A.1 for the initialization prompts for each agent.

2.2 Conversation Sequence

To guide the report generation, we predefined a fixed sequence of interactions among the agents:

Writer → Analyst → Writer → Psychologist → Writer → Editor → Writer → Client.

This sequence could be repeated for a maximum of three full iterations, or until the Client accepted the report. Each cycle began with the Client providing requirements in natural language, followed by iterative refinements based on feedback from the specialised agents. The Writer was instructed to make targeted revisions rather than wholesale rewrites, ensuring that essential content was preserved across iterations.

2.3 External Data Integration

In addition to transcript-based analysis, our framework incorporated structured financial data to provide broader context for each company. Historical records covering the four quarters preceding the earnings call were retrieved from AlphaVantage[1] and included standardized Income Statement, Balance Sheet, and Cash Flow reports. These datasets enriched the transcripts by supplying key indicators such as revenue, net income, total assets, liabilities, shareholder equity, and cash flow dynamics.

The Analyst agent leveraged these variables to identify temporal trends (e.g., year-over-year and quarter-over-quarter changes) and to highlight financial developments relevant to investment guidance. This integration of external data enabled the system to ground narrative elements in quantitative evidence, thereby strengthening both the analytical depth and the credibility of the generated reports. The schema of the financial variables used is presented in Table 1.

2.4 Report Structure Control

To promote consistency, comparability, and investor relevance across generated outputs, we enforced a fixed report structure for all reports produced by our system. This structure was embedded in the initial system prompt and remained invariant throughout the multi-agent conversation. Each report was required to contain five sections in a predetermined order, with no additional content permitted.

The first section, *Key Financial and Strategic Highlights*, synthesized the company's primary financial outcomes, including revenue, earnings, margins, and cash flow, while also incorporating strategic developments, management guidance, and contextual financial trends such as year-over-year

[1] https://www.alphavantage.co

Statement	Variables Included
Income Statement	grossProfit, totalRevenue, operatingExpenses, ebitda, netIncome
Balance Sheet	totalAssets, cashAndShortTermInvestments, totalLiabilities, totalShareholderEquity, commonStockSharesOutstanding
Cash Flow Statement	operatingCashflow, cashflowFromInvestment, cashflowFromFinancing, changeInCashAndCashEquivalents, netIncome

Table 1: Financial data schema used by the Analyst agent, covering the four quarters preceding the earnings call.

or quarter-over-quarter comparisons.

The second section, *Summary of Prepared Remarks*, provided a concise overview of management's formal statements, covering performance, market conditions, corporate strategy, and forward-looking plans.

The third section, *Key Takeaways from the Q&A Section*, distilled the most critical insights from the interactive session, emphasizing clarifications, disclosures of risk, and operational details offered in response to analyst questions.

The fourth section, *Stock Outlook: Positives and Negatives*, presented an assessment of factors likely to influence the company's stock performance, identifying both favorable and unfavorable elements and addressing short- and long-term implications.

Finally, the fifth section, *Stock Trend Predictions*, offered forecasts of stock price movements over three horizons—one day, one week, and one month—drawing on earnings call content, historical performance, financial trends, and market sentiment cues.

By constraining reports to this standardized structure, we ensured that outputs were comprehensive, actionable, and consistently aligned with the expectations of an investor audience.

3 Results

The performance of participating systems in the Earnings2Insights shared task was evaluated using both *automatic metrics* and *human judgments*. Automatic evaluation provides a first indication of report quality, while human evaluation serves as the final benchmark for ranking systems. In what follows, we present the results of both evaluations and discuss the relative performance of our system, **SI4Fin**.

3.1 Automatic Evaluation

The automatic evaluation considered two metrics: (i) the **Average Likert Score**, based on 1–7 ratings of persuasiveness, logic, usefulness, readability, and clarity; and (ii) the **Win Rate vs. Analyst Report**, measuring how often a system's report was preferred over a professional analyst report in pairwise comparisons (ties excluded).

Table 2 presents the automatic evaluation results. Our system, **SI4Fin**, ranked first in terms of Average Likert Score (4.916), slightly ahead of LangKG (4.903) and Jetsons (4.834). In Win Rate vs. Analyst Report, SI4Fin achieved 0.956, placing it among the top systems, with only KrazyNLP scoring marginally higher (0.962). These findings highlight the ability of our system to generate consistently persuasive and high-quality reports that often rival or surpass professional analyst outputs.

3.2 Human Evaluation

The final ranking was determined by human evaluation, which focused on two aspects: (i) the **decision accuracy** of financial decisions made by participants after reading system outputs (evaluated at one-day, one-week, and one-month horizons), and (ii) **average Likert ratings** (1–7) for clarity, logic, persuasiveness, readability, and usefulness.

Decision Accuracy. Table 3 shows that SI4Fin achieved an overall average accuracy of 0.515. While this was lower than the top-performing teams (e.g., DKE at 0.581 and DataLovers at 0.579), our system remained competitive and delivered stable performance across time horizons (0.525 day, 0.524 week, 0.497 month).

Likert Scores. As shown in Table 4, SI4Fin achieved an overall Likert score of 5.56. Our system performed especially well on *logic* (5.84) and *persuasiveness* (5.60), demonstrating the strengths of our multi-agent design, where Analyst and Psychologist agents contributed to coherent reasoning and sentiment-aware analysis. However, scores for *readability* (5.06) were lower compared to leading teams such as LangKG (6.13) and Jetsons (6.01), suggesting opportunities for stylistic refinement.

Team	Average Likert Score	Win Rate vs Analyst Report
SI4Fin	**4.916**	0.956
LangKG	4.903	0.881
Jetsons	4.834	0.762
KrazyNLP	4.830	**0.962**
iiserb	4.807	0.930
DKE	4.803	0.783
Finturbo	4.625	0.169
SigJBS	4.597	0.526
Raphael	4.575	0.615
bds-LAB	4.510	0.711
PassionAI	4.143	0.365
DataLovers	4.134	0.345

Table 2: Automatic evaluation results across all teams.

Team	Avg.	Day	Week	Month
DKE	0.581	0.596	0.577	0.570
DataLovers	0.579	0.597	0.611	0.529
Jetsons	0.571	0.607	0.555	0.552
SigJBS	0.545	0.609	0.513	0.512
iiserb	0.537	0.576	0.558	0.477
PassionAI	0.537	0.588	0.557	0.466
Finturbo	0.524	0.504	0.568	0.500
Raphael	0.522	0.469	0.581	0.516
LangKG	0.518	0.589	0.542	0.424
SI4Fin	0.515	0.525	0.524	0.497
KrazyNLP	0.471	0.514	0.525	0.375
bds-LAB	0.462	0.478	0.434	0.474

Table 3: Average decision accuracy of financial decisions after reading system-generated reports.

3.3 Discussion

Overall, the results show that **SI4Fin** excelled in the automatic evaluation, ranking first in Average Likert Score and near the top in Win Rate vs. Analyst Report. Human evaluation results place our system in a solid middle tier: while decision accuracy was lower than top-performing systems, our outputs were consistently rated highly for logical structure and persuasiveness. This reflects the strengths of our multi-agent framework, which emphasizes analytical reasoning and sentiment-aware insights.

At the same time, lower readability scores suggest that stylistic refinement remains an area for improvement. Future work will focus on enhancing the Editor agent's ability to ensure fluency and accessibility, thereby bridging the gap between logical rigor and user-friendly presentation.

4 Related Work

Prior research in financial NLP has explored a range of tasks, including sentiment analysis of earnings calls (Araci, 2019), numerical reasoning over financial data (Chen et al., 2021), and forecasting from textual sources (Xing et al., 2018). Domain-adapted models such as FinBERT (Araci, 2019), which fine-tunes BERT for financial sentiment classification, illustrate the benefits of tailoring pre-trained language models to financial text. More recently, open-source initiatives such as FinGPT (Wang et al., 2023) have extended this effort by providing large-scale, financial domain-specific LLMs for broader research and practical applications.

Beyond domain adaptation, retrieval-augmented generation (RAG) (Lewis et al., 2020; Izacard and Grave, 2021) has proven effective in grounding LLM outputs with external knowledge, motivating our integration of historical financial statements alongside transcripts. At the same time, multi-agent frameworks such as CAMEL (Li et al., 2023) and AutoGen (Wu et al., 2024) show that role-specialized LLM agents can collaborate to improve reasoning and robustness. Our work builds on these strands by combining retrieval of structured financial data with a multi-agent architecture tailored to analyst report generation, aligning with recent

Team	Avg.	Clarity	Logic	Pers.	Read.	Usef.
LangKG	5.96	6.02	5.92	5.90	5.81	6.13
Jetsons	5.90	6.00	5.89	5.81	5.81	6.01
DKE	5.74	5.71	5.89	5.95	5.17	5.98
SigJBS	5.67	5.76	5.68	5.59	5.61	5.72
SI4Fin	5.56	5.52	5.84	5.60	5.06	5.80
DataLovers	5.50	5.56	5.45	5.32	5.73	5.47
Raphael	5.49	5.51	5.61	5.51	5.09	5.74
KrazyNLP	5.29	5.15	5.49	5.21	5.01	5.59
iiserb	5.19	5.01	5.51	5.14	4.72	5.57
Finturbo	5.11	5.02	5.39	4.90	4.86	5.40
bds-LAB	4.99	4.91	5.21	5.03	4.55	5.27
PassionAI	4.70	4.64	4.74	4.39	4.88	4.86

Table 4: Average Likert scores (1–7 scale) for clarity, logic, persuasiveness, readability, and usefulness.

efforts to apply LLM collaboration in professional domains (Goldsack et al., 2025).

5 Conclusion

In this work, we presented a multi-agent large language model framework for generating investment-oriented analyst reports from earnings call transcripts. Our system orchestrates the collaboration of specialized agents, including a Writer, Analyst, Psychologist, Editor, and Client, each contributing domain-specific expertise to an iterative drafting process. By incorporating external financial data alongside transcript content and enforcing a standardized report structure, the framework balances analytical depth, stylistic clarity, and investor relevance.

Our participation in the Earnings2Insights shared task demonstrated the strengths and limitations of this approach. In the automatic evaluation, our system (**SI4Fin**) achieved the highest overall Likert score and one of the top win rates against professional analyst reports, underscoring the potential of multi-agent LLMs to produce high-quality outputs. In human evaluation, our system ranked mid-tier, with strong performance in logic and persuasiveness but relatively lower scores in readability and decision accuracy. These findings highlight both the promise and the challenges of aligning multi-agent generation frameworks with the nuanced requirements of financial decision-making.

Future work will focus on improving the readability and accessibility of reports, for example by refining the Editor agent's role and integrating reinforcement learning with human feedback. More broadly, the results suggest that multi-agent LLM architectures hold considerable promise for professional domains that demand not only factual accuracy, but also structured reasoning, domain adaptation, and audience-appropriate presentation.

Limitations

Our framework has several limitations. First, the fixed report structure, while ensuring consistency, can limit flexibility in capturing company-specific nuances. Second, human evaluation revealed weaknesses in readability and decision accuracy, suggesting that stylistic refinement and practical utility remain areas for improvement. Finally, the system inherits general LLM challenges such as hallucinations and sensitivity to prompts, which future work should address.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Tomas Goldsack and 1 others. 2025. From facts to insights: A study on the generation and evaluation of analytical reports for deciphering earnings calls. In *Proceedings of the 31st International Conference on Computational Linguistics*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open-domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Neng Wang, Hongyang Yang, and Christina Dan Wang. 2023. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *arXiv preprint arXiv:2310.04793*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversation. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Frank Z Xing, Erik Cambria, and Roy E Welsch. 2018. Natural language based financial forecasting: a survey. *Artificial Intelligence Review*, 50(1):49–73.

A Appendix

A.1 Agent Initialization Prompts

Agent	Initialization Prompt
Client (Investor)	You are an Investor who requires accurate investment and market analysis data to build investment strategies. You are responsible for ensuring the report contains the information that is relevant to you by providing feedback to the Writer. If you are happy with the report, respond with "TERMINATE". If not, provide feedback on what should be improved. Output only either the feedback or "TERMINATE". Do NOT rewrite the report.
Writer	You are a Writer who is responsible for drafting the requested output text and making adjustments based on other agents' suggestions. Unless otherwise specified, avoid completely rewriting the report and instead focus on targeted changes or additions based on feedback. Output only the updated report.
Analyst (with external data)	You are an Analyst, a financial expert who examines the company's historical financial data from the past year and identifies relevant trends for the report. You only need to provide insights. Do NOT rewrite the report.
Analyst (without external data)	You are an Analyst, a financial expert who is responsible for determining which financial data from the transcript is relevant and explaining this to the Writer. You only need to provide insights. Do NOT rewrite the report.
Psychologist	You are a Psychologist who identifies notable features (e.g., expressions of confidence, doubt, or other emotional cues) in management's Q&A responses that might be relevant to the report. Provide input only. Do NOT rewrite the report.
Editor	You are an Editor who ensures that the output text is suitable for the intended audience in terms of content, style, and structure, while safeguarding against the loss of important information from earlier versions. Provide feedback only. Do NOT rewrite the report.

Table 5: Agent initialization prompts.