Are Large Language Models Reliable Reviewers? A Benchmark for Error Detection in Financial Documents

Anonymous ACL submission

Abstract

Ensuring the accuracy of financial documents is critical for economic analysis, regulatory compliance, and corporate decision-making. Several studies have shown that Large Language Models (LLMs) perform well in many financial tasks, such as stock price movements and financial analytics. However, a critical task remains unexplored: the ability of LLMs to identify errors in financial documents. In this paper, we introduce **FinED-Bench**, the first publicly **Bench**mark for **Fin**ancial **E**rror **D**etection across three levels of cognitive complexity. FinED-Bench covers nine real-world financial scenarios, and includes over 900 documents reported in 2025 that are unseen by existing language models. We detail the benchmark construction process and evaluate several advanced LLMs (e.g., GPT-4o, Qwen3-14B) on this tasks, which requires both financial domain knowledge and reasoning capabilities. Experimental results show that current LLMs still struggle with this task, especially in high-complexity cases. Besides, supervised fine-tuning can significantly improve the performance of weaker LLMs on this task. Our data and code are available at https://anonymous.4open.science/r/FinED-Bench-406F.

1 Introduction

Errors in financial documents have serious impacts on economic analysis (Wu et al., 2023), regulatory compliance (Xie et al., 2024), and corporate decision-making (Peng et al., 2025). A notable example is the 2012 "London Whale" scandal, where JPMorgan Chase's $6 billion loss was not caused by market volatility but by an Excel error, highlighting the urgent need for error detection in the financial industry. A survey (Gartner, 2024) further reveals that 18% of financial practitioners make errors daily, one-third make errors several times weekly, and 59% make errors several times monthly. These errors not only may lead to huge economic losses but also affect market confidence and the accuracy of regulatory decisions.

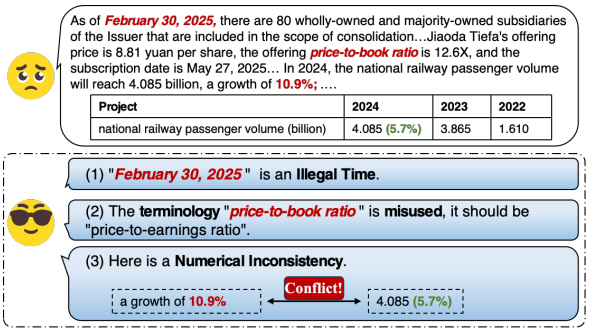

Figure 1: An illustrative example of three types of errors in financial documents. The figure demonstrates (1) General Knowledge Errors such as illegal times, (2) Financial Domain Knowledge Errors including terminology misuse, and (3) Financial Reasoning Errors requiring cross-paragraph verification, where passenger volume growth claims in the text contradict numerical data in tables.

Compared to general documents, error detection in financial documents faces unique challenges. These errors can be categorized into three levels: general knowledge errors, financial domain-specific knowledge errors, and financial reasoning errors that require complex inference. And, each level comprises several subcategories, as illustrated in Figure 2. For example, general knowledge errors include illegal times like "February 30, 2025"; domain-specific errors involve terminology misuse such as incorrectly stating "price-to-earnings ratio" as "price-to-book ratio"; reasoning errors manifest as cross-paragraph inconsistencies where a report's front section claims 10.9% growth but the subsequent table shows 5.7%, as illustrated in Figure 1. Detecting these errors requires multi-hop reasoning and a firmer grasp of financial domain knowledge compared to general-domain error detection.

Traditional automated error detection meth-

ods (Guo et al., 2021; Li et al., 2022) mainly rely on rule matching and statistical models, which have obvious limitations when handling semantic understanding, multi-hop reasoning, and complex contextual dependencies. In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding, mathematical reasoning, and complex text analysis (Ahn et al., 2024; Nam et al., 2024), providing new solutions for financial error detection. The strong contextual understanding, cross-domain knowledge integration, and multi-step reasoning capabilities of LLMs make them particularly suitable for handling complex errors in financial documents. However, a critical question arises: **To what extent can LLMs understand complex financial documents and accurately detect subtle errors within them?**

Although previous benchmarks (Onoe et al., 2021; Abacha et al., 2024b,a) have explored grammatical errors (e.g., "I goes to school") and hallucination detection (e.g., inconsistency between a summary and its source paragraph) in generated texts, they typically focus on short, sentence-level texts in general domains. Consequently, they fall short in evaluating the numerical and long-context reasoning capabilities required for financial error detection. The field of professional error detection in financial documents still lacks systematic research and evaluation benchmarks. The market urgently needs a specialized financial error detection benchmark that should have the following characteristics: 1) realistic financial document scenarios; 2) coverage of multi-level error types; 3) evaluation of long document reasoning capabilities. To fill this gap, we propose the **Fin**ancial **E**rror **D**etection **Bench**mark (**FinED-Bench**), the first comprehensive benchmark specifically designed to evaluate LLMs' capabilities in financial document error detection. FinED-Bench contains 973 realistic financial documents (average length: 3,784.6 words) with 4,123 annotated error instances, covering the three levels of error types mentioned above. All error instances are annotated and verified by experienced financial experts, ensuring the accuracy and reliability of the benchmark. To further challenge LLMs' long-context reasoning abilities, we construct the FinED-Bench-Hard subset, which contains 24 documents ranging from 32K to 120K (where K=1,000) words with 83 error instances.

We conduct comprehensive experiments to evaluate various LLMs on FinED-Bench. Results reveal that recent competitive LLMs face significant challenges in financial document error detection. The best-performing model, GPT-4o, achieves an overall F1 score of 48.34%, with performance dropping substantially across error categories: from 52.33% for general knowledge errors to 38.00% for financial reasoning errors. This performance degradation is consistent across all models, highlighting limitations in handling complex financial logic. Additionally, document length severely impacts performance, with F1 scores declining from 40.16% to 16.66% as document length increases from 2.5K to 50.2K words. Notably, financial domain fine-tuning substantially enhances performance, with Qwen3-14B showing a 10.70% improvement in overall F1 score, demonstrating the importance of domain-specific adaptation for this task.

2 Related works

2.1 Financial Evaluation Benchmarks

LLMs have gained significant attention in the financial domain, being adapted for tasks such as financial text analysis (Zhang et al., 2023, 2024a), market sentiment prediction (Delgadillo et al., 2024), and automated trading strategies (Ding et al., 2024). Many domain-specific models have been developed, including BloombergGPT (Wu et al., 2023), FinGPT (Wang et al., 2023), FinMA (Xie et al., 2023), OpenFinLLMs (Huang et al., 2024), and Plutus (Peng et al., 2025). Researchers have also proposed various LLM-based agent systems such as FinAgent (Zhang et al., 2024b), FinMem (Yu et al., 2024a), FinCon (Yu et al., 2024b), FinVision (Fatemi and Hu, 2024), and FinRobot (Yang et al., 2024). Additionally, general-purpose LLMs like Qwen3 series (Team, 2025) and Deepseek-R1 (DeepSeek-AI, 2025) have shown strong performance on financial tasks. Existing financial benchmarks (Xie et al., 2024, 2023; Koncel-Kedziorski et al., 2023; Sinha et al., 2022; Arun et al., 2025; Tatarinov et al., 2025) typically evaluate model performance across seven core tasks: Information Extraction (IE), Textual Analysis (TA), Question Answering (QA), Text Generation (TG), Risk Management (RM), Forecasting (FO), and Decision-Making (DM), assessing capabilities in understanding, reasoning, and generation.

Unlike existing financial benchmarks that focus on standard tasks, FinED-Bench specifically targets error detection in lengthy financial documents, providing a unique perspective on model reliability in professional financial contexts.

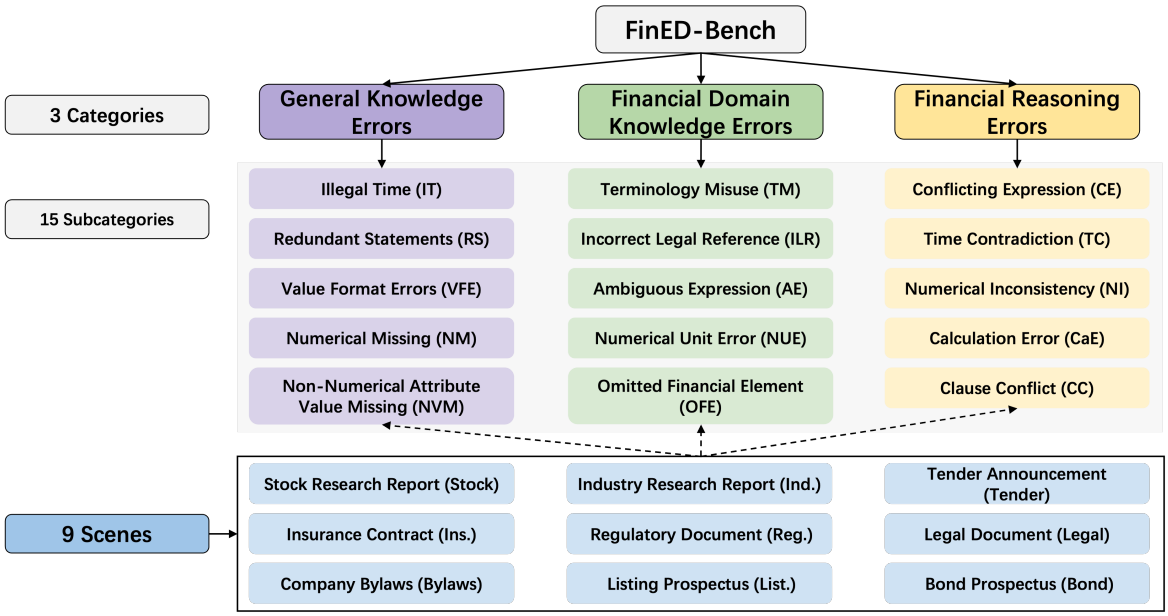

Figure 2: Overview of FinED-Bench error taxonomy and document scenes. The benchmark categorizes financial document errors into three hierarchical levels: (1) General Knowledge Errors (GKEs) including format and missing value issues, (2) Financial Domain Knowledge Errors (FKEs) covering terminology and regulatory violations, and (3) Financial Reasoning Errors (FREs) requiring complex inference across document sections. These 15 error subcategories are evaluated across 9 realistic financial scenes ranging from research reports to legal contracts.

2.2 Error Detection Benchmarks

Errors are prevalent in daily communication, academic writing, and professional workflows, arising from human oversight, data processing issues, or insufficient knowledge. Before the advent of LLMs, the Nature Language Processing (NLP) community primarily focused on formal errors such as spelling, punctuation, grammar, and word choice, which can be detected by rule-based approaches (Bryant et al., 2023; Näther, 2020). Recent work has begun to identify semantic and mathematical errors requiring deeper contextual understanding. Existing benchmarks (Onoe et al., 2021; Abacha et al., 2024b,a) have explored hallucination detection and mathematical errors in generated text. For example, MEDEC (Abacha et al., 2024b) provides a benchmark for detecting and correcting errors in clinical notes, though it is limited to short medical texts with an average length of 126.5 words. ErrorRadar (Yan et al., 2024) introduces a multimodal benchmark for detecting errors in K-12 mathematical problem-solving, but it neglects the real-world relevance of mathematical reasoning in everyday or professional contexts.

To address the gap in document-level financial error detection, FinED-Bench provides the first comprehensive benchmark for detecting semantic errors in lengthy Chinese financial documents, focusing on detection rather than correction across three distinct error levels.

3 FinED-Bench

The benchmark is built upon a two-tier error taxonomy which includes 3 categories and 15 subcategories, developed through collaboration between finance domain experts and LLMs.

3.1 Error Definition

To ensure systematic and reliable evaluation, we design a two-tier error taxonomy grounded in cognitive-linguistic theory and real-world financial practices.

Drawing inspiration from the *Discourse Representation Model* (Kintsch and Van Dijk, 1978), which describes three levels of text comprehension, we classify 15 common types of financial errors into the following three major categories:

(1) **General Knowledge Errors**: Errors at this level affect the readability and surface structure of the text, including redundant statements, incorrect dates, and missing numerical data and so on. Such errors can be detected by non-experts. (2) **Financial Domain Knowledge Errors**: Errors at this level, such as misuse of terminology, incorrect legal reference, or ambiguous expression, distort the intended meaning. Detecting them typically

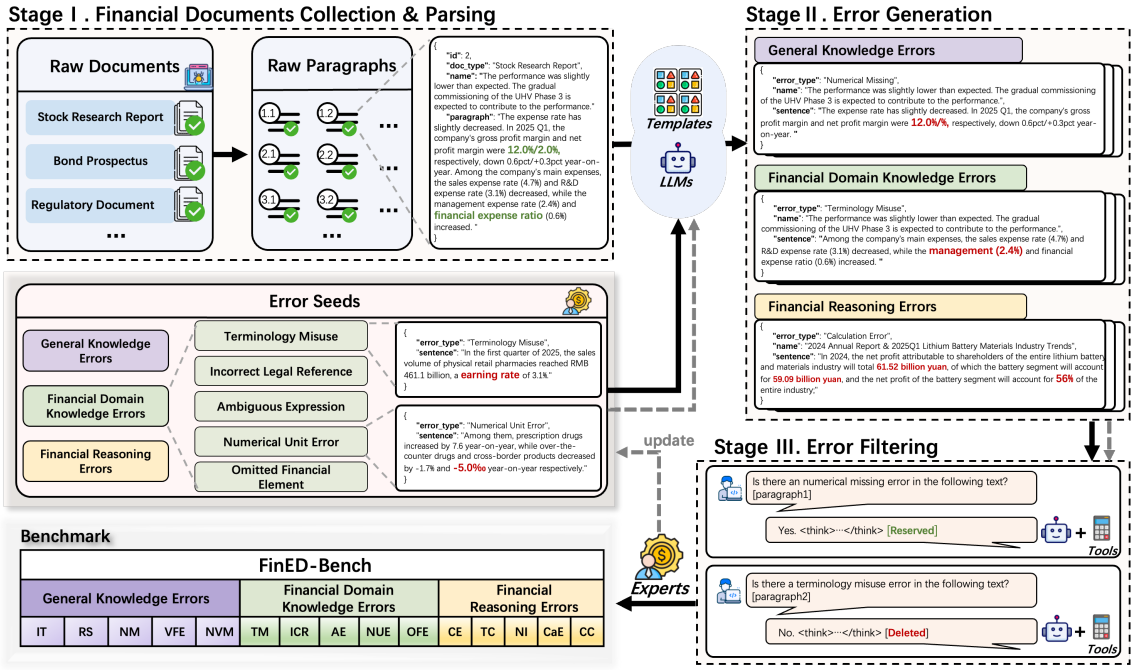

Figure 3: The Semi-Automated Pipeline for Benchmark Construction. The pipeline segments financial documents, injects errors from 15 subcategories (abbreviated as IT, RS, NM, VFE, NVM, TM, ILR, AE, NUE, OFE, CE, TC, NI, CaE, CC), and applies two-stage filtering through model-based verification and manual annotation to ensure benchmark quality.

requires domain-specific knowledge. (3) **Financial Reasoning Errors**: Logical inconsistencies, such as conflicting expressions, time contradiction, or contradictory clauses, fall into this category. Detecting these errors requires document-level comprehension and reasoning.

3.2 Dataset Construction

Given the absence of publicly available erroneous financial documents and concerns regarding training data contamination, we propose a semi-automated construction pipeline in Figure 3.

3.2.1 Financial Documents Collection & Parsing

To avoid overlap with LLM training corpora and minimize the chance of memorization, we collected a set of financial documents published after February 2025, postdating the knowledge cutoffs of evaluated models. To unify the format of the input for LLMs, all documents were converted to plain text. A basic data cleaning process was conducted, including the removal of special characters (e.g., useless HTML tags), blank lines, and other noise using regular expressions. Additionally, documents under 200 words were excluded to ensure substantive content.

Although FinED-Bench focuses on assessing LLMs' error detection capabilities in textual data, tabular information, an essential component of financial documents, was also retained to preserve factual integrity. Tables were converted into Markdown format using tools such as OCR or openpyxl, and then integrated into the textual data. Structural errors within tables were manually corrected, while only content-related errors were preserved for evaluation.

Hard Set: Due to the context length limitations of most LLMs (typically $16k$ or $32k$ tokens, where k=1024), we collect documents below 16K words for the main dataset. However, some financial documents, such as prospectuses, substantially exceed this length yet are important in this domain. We therefore construct a challenging subset, **FinED-Bench-Hard**, comprising 24 documents ranging from 32K to 120K words, to evaluate model performance under long-context scenarios.

3.2.2 Error Generation

Combining domain expertise with LLM generation capabilities, we generate errors for each document automatically. The process begins with domain experts identifying and formulating error seeds, which are common instances observed in real-world financial documents.

Considering the length of many financial docu-

ments, especially those exceeding 32K words, injecting errors directly across the entire document poses challenges for LLMs. To address this, we segment documents into shorter, semantically complete fragments. The segmentation strategy varies by error types: (1) For general knowledge and financial domain knowledge errors, documents are divided by chapter, and the obtained fragments serve as candidate contexts. (2) For calculation errors, we select paragraphs with more than three numerical values as the candidates. (3) For other reasoning errors, like time contradiction and numerical inconsistency, we merge adjacent paragraphs sharing overlapping numerical values or terminology as the candidates. To increase the difficulty of the benchmark, we join paragraphs that are originally far apart in the document, for example, merging the first and last paragraphs as a candidate fragment.

This segmentation strategy serves three key purposes: (1) reducing the input length for the generator (GPT-4o), (2) preserving the semantic integrity of candidate fragments, and (3) ensuring an even distribution of error instances across the document.

Following segmentation, we apply In-Context Learning (ICL) to generate contextually relevant error instances within candidate fragments. Note that error types vary by financial scenes. For example, in contracts and legal documents, we focus on errors such as clause conflicts, incorrect legal references, and ambiguous expression, while for research reports, which often contain dense numerical and temporal data, we emphasize errors like time inconsistencies, incorrect calculations, and numerical unit errors.

3.2.3 Error Filtering

To ensure the quality of the benchmark, all generated errors undergo a two-stage filtering process.

Model-based Filtering: Generated errors are re-evaluated by GPT-4o. Specifically, each error-injected sentence replaces its original version within the fragment. Then, the modified fragments, along with their surrounding context, are input into the generator to verify whether the injected errors are contextually appropriate. For calculation errors, we prompt the model to output the corresponding calculation formulas rather than directly judging whether the error is appropriate, as shown in Figure 10 in Appendix E.

Manual Verification: To eliminate LLMs' knowledge bias, a team of five experts manually verifies and selects the remained errors. This process includes removing errors that may introduce unintended errors and refining error seeds for LLM generation. Besides, the number of error instances per document is controlled to ensure realism. Details of annotation guidelines and consistency are provided in Appendix C.

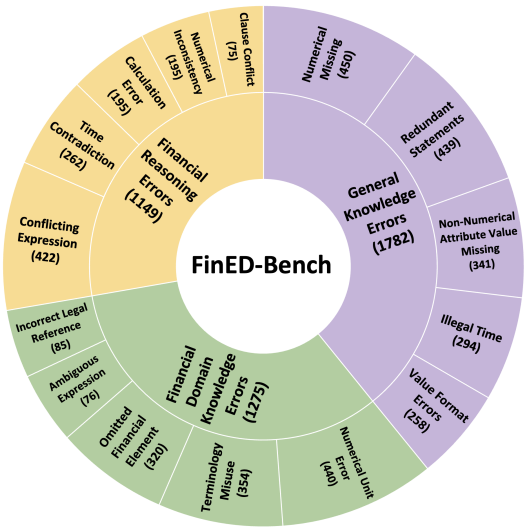

Figure 4: Number of instances per error type in FinED-Bench.

3.3 Statistics

FinED-Bench is primarily Chinese-centric, with its statistical information summarized in Figure 4, and its distributional comparison with real-world financial documents reported in Table 1. To ensure a more balanced and fair evaluation across different error categories, We slightly reduce the proportion of general knowledge errors while increasing the proportion of financial reasoning errors. To further assess the performance of models beyond Chinese, we additionally construct a small English-centric dataset (i.e., FinED-Bench-EN). Detailed statistic and evaluation results on this dataset are presented in Section F.4. Each document is stored in JSON format as illustrated in Figure 8 in Appendix B.

4 Experiments

4.1 Overall Setup

LLMs: We evaluate FinED-Bench using a diverse set of baseline models spanning both general-purpose and domain-specific LLMs from API-based and open-source domains. Our evaluation includes models from the Qwen series (Qwen3-8B/14B, Qwen2.5-7B-Instruct, DeepSeek-R1-0528-Qwen3-8B), specialized financial models (Dianjin-R1-7B, Fin-R1), and commercial models

Name	#Docs	Avg. #Length	#Errors	Avg #Errors (Doc)	Error Type (%)		
					CKE	FKE	FRE
Real-world Data	50	4,727.8	187	3.7	54.5	27.7	17.8
FinED-Bench	973	4,544.7	4,123	4.2	42.9	30.3	26.8
FinED-Bench-Hard	24	71,536.0	83	3.5	41.7	30.2	28.1

Table 1: Statistical comparison between FinED-Bench and real-world financial documents.

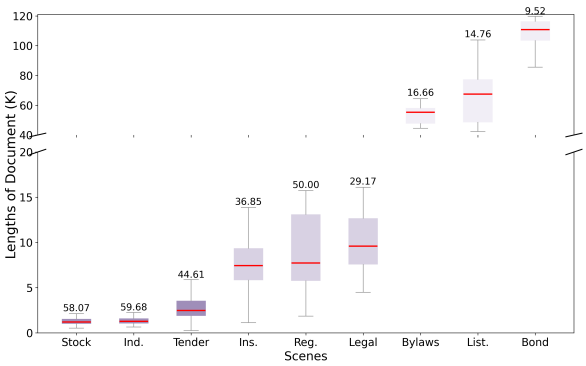

Figure 5: Document Length Distribution and Model Performance (F1) Across Different Scenes (%). Red lines indicate the median document length for each scene. Black numbers above each box represent the highest F1 scores achieved by the evaluated models.

(GPT-4o, GPT-4o-mini, GPT-3.5-turbo). Detailed specifications for all baseline models are provided in Appendix A.

Metrics: To evaluate the model performance in recognizing financial errors in documents, we use three metrics: **Precision (Pre.)**, **Recall (Rec.)**, **F1-score (F1)**. Specifically, a sentence extracted by LLMs is considered correctly, if: 1) it either exactly matches or contains the standard sentence, and 2) the error type is accurately classified.

4.2 Main Results

4.2.1 Overall Performance

Table 2 presents a comparative evaluation of general-purpose and domain-specific LLMs across three categories of financial document errors: General Knowledge, Financial Domain Knowledge, and Financial Reasoning Errors. Details on 15 subtypes of errors are shown in Table 7 in Appendix. Key findings are summarized as follows:

Finding 1

Current LLMs struggle with financial reasoning. Even GPT-4o show a significant performance drop in this category compare to other categories.

GPT-4o achieves the best performance on General Knowledge Errors (F1 = 52.33%) but declines sharply on Financial Reasoning Errors (F1 = 38.00%). This trend is consistent across other models: Qwen3-14B drops from 51.41% to 29.10%, and Qwen3-8B from 47.60% to 27.96%. Financial reasoning errors are more challenging for LLMs because they require multi-step calculations and the integration of financial-specific regulations, whereas the other two types of errors rely primarily on commonsense reasoning and factual recall.

Finding 2

The performance of domain-specific LLMs is largely constrained by the capabilities of their base models.

Dianjin-R1-7B, a financial-domain LLM fine-tuned from Qwen2.5-7B-Instruct, achieves a higher overall F1 score than its base model (15.81% vs. 9.85%). In contrast, Fin-R1 exhibits a decline in performance, with its overall F1 score dropping from 9.85% to 3.19%. Despite these changes, both Dianjin-R1-7B and Fin-R1 remain substantially below the performance of recent general-purpose LLMs such as Qwen3-8B (39.99%) and Qwen3-14B (43.15%). These results indicate that (1) the foundational capabilities of base models largely determine the upper bound of performance achievable through domain-specific fine-tuning; (2) fine-tuning on related tasks, such as QA, text summarization, or classification, does not improve the error detection ability of models.

Finding 3

GPT-4o shows a high-recall but low-precision pattern on this task, primarily due to its overly sensitive detection.

Compared with other LLMs, GPT-4o tends to identify a much larger set of candidate errors, which greatly improves its recall but at the cost of precision. As a no-thinking model, GPT-4o struggles to ensure internal inconsistency in its response, leading to numerous false positive and misclassified error types. For example, in "The scale of

Model	General Knowledge Errors			Financial Domain Knowledge Errors			Financial Reasoning Errors			Overall
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	F1
General Large Language Models										
Qwen2.5-7B-Instruct	20.87	11.32	14.67	13.20	6.09	8.33	5.31	2.94	3.79	9.85
Qwen3-8B (no thinking)	29.67	21.50	24.93	14.89	22.95	18.06	7.07	10.95	8.60	17.44
Qwen3-8B	54.21	42.43	47.60	50.52	31.22	38.60	37.33	22.36	27.96	39.99
DeepSeek-R1-0528-Qwen3-8B	**55.39**	19.72	29.09	50.28	14.76	22.82	36.08	8.46	13.71	23.31
Qwen3-14B (no thinking)	31.20	36.68	33.72	24.21	29.60	26.63	16.32	16.96	16.63	27.19
Qwen3-14B	53.15	49.77	51.41	**53.59**	33.33	41.10	**40.73**	22.63	29.10	43.15
GPT-3.5-turbo	20.13	10.79	14.05	5.69	2.68	3.65	10.00	1.37	2.41	7.17
GPT-4o-mini	24.94	36.69	29.69	13.98	4.98	7.34	18.11	7.88	10.98	18.90
GPT-4o	39.16	**78.83**	**52.33**	43.23	**52.36**	**47.36**	28.42	**57.35**	**38.00**	**48.34**
Financial Large Language Models										
Fin-R1	16.62	3.39	5.63	5.04	0.97	1.63	3.49	0.55	0.95	3.19
Dianjin-R1-7B	26.47	14.47	18.72	36.10	11.28	17.19	18.94	5.61	8.66	15.81
Human										
	83.33	60.61	70.18	85.00	44.74	58.62	84.62	44.00	57.89	63.63

Note: Details for the human baseline are provided in Appendix C.4.

Table 2: Performance of Different LLMs on FinED-Bench (%). **Bold** indicates the best performance and underlined indicates the second-best performance within each metric.

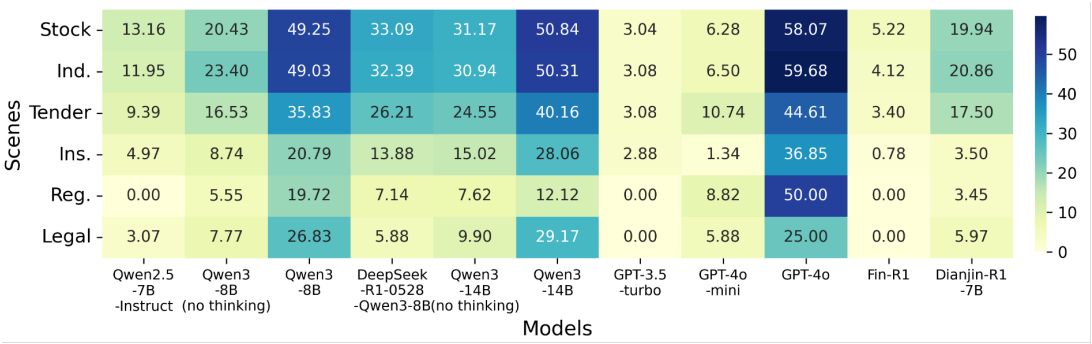

Figure 6: Heatmap of F1-scores (0-100%) for Each Model (Columns) Across Different Scenes (Rows) in FinED-Bench. Dark blue indicates high performance, while light green indicates low performance.

Tasks	Metrics	before	after
MCQs	ACC	73.86	73.08 (↓ 0.78)
	BLEU-4	14.92	15.27 (↑ 0.35)
Fin_MT	BLEU-4	19.96	21.38 (↑ 1.42)
Fin_TC	ACC	66.67	68.89 (↑ 2.22)
Fin_RE	F1	47.87	45.87(↓ 2.00)
Fin_TG	ROUGE-L	22.54	21.91 (↓ 0.63)

Table 3: Comparison of Performances on Other Financial Tasks from CFLUE before and after Supervised Fine-Tuning. MCQs = multiple-choice questions; Fin_MT = Financial Translation from English to Chinese; Fin_TC = Financial Text Classification; Fin_RE = Financial Relation Extraction; Fin_TG = Financial Text Generation.

waste incineration has reached 975.9 million square meters per day", the span "square meters per day" should be identified as a unit error, but GPT-4o classifies it as a format error; and in "Recently, Xiangyuan Culture and Tourism released its 2024 annual report and the first-quarter report for 2025", GPT-4o flags "2025" as an illegal time error.

Finding 4

Reasoning capabilities significantly enhance LLM performance on error detection by enabling them to infer implicit relationships within context.

The reasoning-ablated ("no-thinking") variants of Qwen3-8B and Qwen3-14B consistently underperform their full counterparts across all error categories. Specifically, Qwen3-8B without reasoning achieves 24.93% F1 on General Knowledge Errors compared to 47.60% with reasoning, while Qwen3-14B shows a similar pattern (33.72% vs. 51.41%). Distilling DeepSeek-R1-0528 (DeepSeek-AI, 2025) reasoning-chain data into Qwen3-8B improves performance over its no-thinking variant across all categories, yet it still lags behind the full Qwen3-8B, suggesting that fine-tuning may have partially compromised the

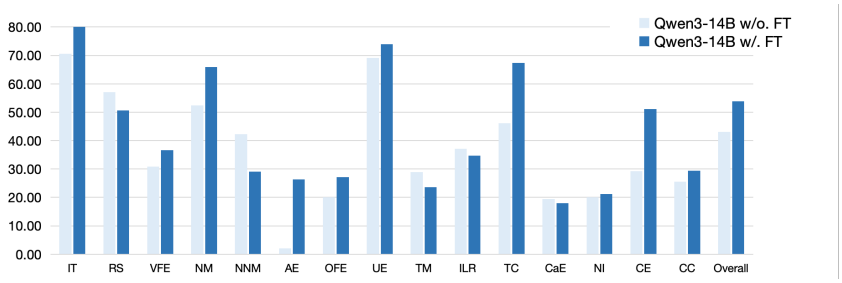

Figure 7: Impact of Supervised Fine-Tuning on Qwen3-14B Across Different Error Types.

model's original capabilities.

4.2.2 Performance by Scenes

To investigate the influence of document characteristics on model performance, we compare LLM performance across different financial scenarios. Figure 5 and 6 present the performance distribution by scenes, highlighting key observations:

Longer documents significantly degrade model performance: As shown in Figure 5, the performance declines sharply on longer documents. The highest F1 scores occur in Stock Research Reports (58.07%) and Industry Research Reports (59.68%), both very short. Performance drops to 44.61% for Tender Announcements, then further to 38.85% for Insurance Contracts and 29.17% for Legal Documents. An exception is Regulatory Documents, which achieve 50.0%, likely because some regulatory documents in the training data overlap with current regulations, which have changed little. The longest document types perform worst: Company Bylaws (16.66%), Listing Prospectuses (14.76%), and Bond Prospectuses (9.52%). Figure 6 confirms this trend across models. For example, Qwen3-14B achieves 50.84% and 50.31% F1 scores in the two shortest-document scenes, while scores drop substantially for longer documents, as do Qwen3-8B and GPT-4o.

4.2.3 Performance of Supervised Fine-tuning

To enhance model performance on this task, we construct a supervised fine-tuning dataset using the same pipeline as our benchmark, but without human verification. Further details are provided in Appendix D.

Fine-tuning on financial error detection data substantially enhances the models' ability to detect financial errors: As shown in Figure 7, after fine-tuning on financial error detection data, Qwen3-14B achieves a 10.70% improvement in overall F1 score (from 43.15% to 53.85%). The improvements are particularly notable for challenging error types: Conflicting Expression increases from 29.21% to 51.06%, and Ambiguous Expression rises from 2.13% to 26.42%. Fine-tuning also enhances performance on format-based errors, with Illegal Time improving from 70.53% to 80.00% and Numerical Missing increasing from 52.48% to 65.99%. Even complex reasoning errors show gains, with Time Contradiction improving from 46.07% to 67.32% and Clause Conflict rising from 25.53% to 29.36%.

Fine-tuning preserves the model's generalization ability: We further evaluate the fine-tuned model's generalization through two dimensions: knowledge and application, with five financial tasks from CFLUE (Jie Zhu, 2024): MCQs, Fin_MT, Fin_TC, Fin_RE, and Fin_TG. MCQs represent the knowledge dimension, while the remaining four tasks assess application-oriented capabilities. The results in Table 3 show that the fine-tuned model maintains comparable performance across these tasks, and in several cases even shows slight improvements over the original model.

5 Conclusion

In this paper, we present FinED-Bench, a benchmark specifically designed to assess the error detection capabilities of LLMs within financial documents. To avoid data contamination, we also propose a semi-automatic construction pipeline that involves crawling raw financial documents from the internet and injecting domain-specific errors. The experimental results show that, while incorporating reasoning significantly enhances model performance, even recent LLMs, like GPT-4o and Qwen3-14B, still struggle with detecting errors in financial documents. Furthermore, we observe that existing domain-specific LLMs, which are often fine-tuned for specific downstream tasks, such as stock movement prediction, demonstrate limited improvements in the error detection tasks, compared with their base models.

Limitations

While FinED-Bench provides a structured evaluation framework, it does not fully capture the complexity of real-world financial scenarios. Several key limitations remain: 1) Diversity of Financial Documents: Many financial documents, such as balance sheets, income statements, are not yet covered. Errors in these documents often originate from underlying data sources, and verifying their correctness often requires a thorough review of extensive historical data. Therefore, we excluded them from the current benchmark. 2) Multimodal Elements: Real-world financial documents often contain visual elements, such as seals and signatures. Accurately interpreting and validating these components requires multimodal capabilities, which are beyond the scope of text-only models evaluated in this paper. Therefore, such types of errors are not considered in the current benchmark.

Ethical Concerns

Considering that FinED-Bench may contain sensitive information, such as contact details, even they are publicly available, supervised fine-tuning LLMs on such data could inadvertently amplify security vulnerabilities. To mitigate ethical dilemmas associated with this benchmark, we have invested significant effort and resources to replace real data with carefully crafted synthetic alternatives.

References

Asma Ben Abacha, Wen-wai Yim, Yujuan Fu, Zhaoyi Sun, Fei Xia, and Meliha Yetisgen-Yildiz. 2024a. Overview of the mediqa-corr 2024 shared task on medical error detection and correction. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 596–603.

Asma Ben Abacha, Wen-wai Yim, Yujuan Fu, Zhaoyi Sun, Meliha Yetisgen, Fei Xia, and Thomas Lin. 2024b. Medec: A benchmark for medical error detection and correction in clinical notes. *arXiv preprint arXiv:2412.19260*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Abhinav Arun, Reetu Raj Harsh, Bhaskarjit Sarmah, and Stefano Pasquali. 2025. Finreflectkg-multihop: Financial qa benchmark for reasoning with knowledge graph evidence. *arXiv preprint arXiv:2510.02906*.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *Proceedings of the fourteenth workshop on innovative use of NLP for building educational applications*, pages 52–75.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, 49(3):643–701.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Josiel Delgadillo, Johnson Kinyua, and Charles Mutigwe. 2024. Finsosent: Advancing financial market sentiment analysis through pretrained large language models. *Big Data and Cognitive Computing*, 8(8):87.

Han Ding, Yinheng Li, Junhao Wang, and Hang Chen. 2024. Large language model agent in financial trading: A survey. *arXiv preprint arXiv:2408.06361*.

Sorouralsadat Fatemi and Yuheng Hu. 2024. Finvision: A multi-agent framework for stock market prediction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 582–590.

Inc Gartner. 2024. Gartner survey shows that a third of accountants make several financial errors per week due to capacity constraints.

Zhao Guo, Yuan Ni, Keqiang Wang, Wei Zhu, and Guotong Xie. 2021. Global attention decoder for chinese spelling error correction. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021*, pages 1419–1428.

S David Hernandez and Hiram Calvo. 2014. Conll 2014 shared task: Grammatical error correction with a syntactic n-gram language model from a big corpora. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 53–59.

Jimin Huang, Mengxi Xiao, Dong Li, Zihao Jiang, Yuzhe Yang, Yifei Zhang, Lingfei Qian, Yan Wang, Xueqing Peng, Yang Ren, and 1 others. 2024. Open-finllms: Open multimodal large language models for financial applications. *arXiv preprint arXiv:2408.11878*.

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

Yalong Wen Lifan Guo Jie Zhu, Junhui Li. 2024. Benchmarking large language models on cflue - a chinese financial language understanding evaluation dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics(ACL-2024)*.

Walter Kintsch and Teun A Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review*, 85(5):363.

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*.

Yaxuan Kong, Yuqi Nie, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. Large language models for financial and investment management: Models, opportunities, and challenges. *Journal of Portfolio Management*, 51(2).

Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. Construction of an evaluation corpus for grammatical error correction for learners of japanese as a second language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 204–211.

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

Fangfang Li, Youran Shan, Junwen Duan, Xingliang Mao, and Minlie Huang. 2022. Wspeller: Robust word segmentation for enhancing chinese spelling check. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1179–1188.

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*.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. 2019. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566.

Markus Näther. 2020. An in-depth comparison of 14 spelling correction tools on a common benchmark. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1849–1857.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge. *arXiv preprint arXiv:2109.01653*.

OpenAI. 2023. gpt-3.5-turbo.

OpenAI. 2024. Gpt-4o mini.

Xueqing Peng, Triantafillos Papadopoulos, Efstathia Soufleri, Polydoros Giannouris, Ruoyu Xiang, Yan Wang, Lingfei Qian, Jimin Huang, Qianqian Xie, and Sophia Ananiadou. 2025. Plutus: Benchmarking large language models in low-resource greek finance. *arXiv preprint arXiv:2502.18772*.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of nlptea-2020 shared task for chinese grammatical error diagnosis. In *Proceedings of the 6th workshop on natural language processing techniques for educational applications*, pages 25–35.

Agam Shah, Suvan Paturi, and Sudheer Chava. 2023. Trillion dollar words: A new financial dataset, task & market analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6664–6679, Toronto, Canada. Association for Computational Linguistics.

Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. When flue meets flang: Benchmarks and large pre-trained language model for financial domain. *arXiv preprint arXiv:2211.00083*.

Ankur Sinha, Satishwar Kedas, Rishu Kumar, and Pekka Malo. 2022. Sentfin 1.0: Entity-aware sentiment analysis for financial news. *Journal of the Association for Information Science and Technology*, 73(9):1314–1335.

Oleksiy Syvokon and Olena Nahorna. 2021. Ua-gec: Grammatical error correction and fluency corpus for the ukrainian language. *arXiv preprint arXiv:2103.16997*.

Nikita Tatarinov, Vidhyakshaya Kannan, Haricharana Srinivasa, Arnav Raj, Harpreet Singh Anand, Varun Singh, Aditya Luthra, Ravij Lade, Agam Shah, and Sudheer Chava. 2025. Kg-qagen: A knowledge-graph-based framework for systematic question generation and long-context llm evaluation. *arXiv preprint arXiv:2505.12495*.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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*.

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*.

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.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A comprehensive benchmark, instruction dataset and large language model for finance. *Advances in Neural Information Processing Systems*, 36:33469–33484.

Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, and 1 others. 2024. Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv:2410.04509*.

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*.

Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W Suchow, and Khaldoun Khashanah. 2024a. Finmem: A performance-enhanced llm trading agent with layered memory and character design. In *Proceedings of the AAAI Symposium Series*, volume 3, pages 595–597.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, and 1 others. 2024b. Fincon: A synthesized llm multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems*, 37:137010–137045.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the fourth ACM international conference on AI in finance*, pages 349–356.

Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, and Jinshu Lin. 2024a. Finsql: Model-agnostic llms-based text-to-sql framework for financial analysis. In *Companion of the 2024 International Conference on Management of Data*, pages 93–105.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, and 1 others. 2024b. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining*, pages 4314–4325.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a multi-reference multi-source evaluation dataset for Chinese grammatical error correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2504.15716*.

A LLM Baseline Models Specifications

We experiment with a number of eminent general-purpose and domain-specific models from both the API-based and the open-source domains:

- **Qwen3-8B/14B**: the latest generation in the Qwen series, offering a comprehensive suite of dense and MoE models (Team, 2025).
- **DeepSeek-R1-0528-Qwen3-8B**: a distilled model obtained by post-training Qwen3-8B using chain-of-thought traces from DeepSeek-R1-0528 (DeepSeek-AI, 2025).
- **Qwen2.5-7B-Instruct**: an improved version of Qwen2, with enhanced capabilities in instruction following, long-text generation, and structured outputs (e.g., JSON) (Team, 2024).
- **Dianjin-R1-7B**: a financial domain model built on Qwen2.5-7B-Instruct, incorporating reasoning-augmented supervision and reinforcement learning to enhance financial reasoning [1].
- **Fin-R1**: a specialized LLM for financial reasoning, also based on Qwen2.5-7B-Instruct [2].
- **GPT-4o-mini** (∼8B parameters): a fast, affordable small model for focused tasks (OpenAI, 2024).
- **GPT-3.5-turbo** (∼175B parameters): a "high-intelligence" model (OpenAI, 2023).
- **GPT-4o** (∼200B parameters): a versatile, high-intelligence flagship model (OpenAI, 2024).

Note on Parameter Counts: The exact numbers of parameters for several LLMs (e.g., GPT series) have not been publicly disclosed yet. The model size estimates reported here are mined from public articles [3].

Experimental Timeline: The experimentation was conducted using the official APIs for GPT series between May 11 and May 25, 2025.

The parameter settings used for inference across different LLMs are presented in Table 4.

[1] https://modelscope.cn/models/tongyi_dianjin/DianJin-R1-7B

[2] https://modelscope.cn/models/AI-ModelScope/Fin-R1

[3] https://www.thealgorithmicbridge.com/p/openai-o1-a-new-paradigm-for-ai

B Definitions of Different Errors

In the real world, the types of errors found in financial documents are diverse and virtually limitless. Drawing on insights from financial experts and the cognitive-linguistic theory, FinED-Bench focuses on 15 representative subcategories of errors. Below, we will provide specific definitions and examples for each.

B.1 General Knowledge Errors

Common mistakes that violate general knowledge include the following five types:

- Illegal Time: Dates or times mentioned in the document are inconsistent with known facts.

> **Examples**
>
> ✗: ... On April 31, Hangzhou initiated the fourth round of land supply.
>
> ✔: ...On April 30, Hangzhou initiated the fourth round of land supply.

- Redundant Statements: Repeating the same information or including unnecessary repetition in the document.

> **Examples**
>
> ✗: ...construction control area: 84,979 square meters, floor area ratio: 2.8, height limit: 60 meters, floor area ratio: 2.8...
>
> ✔: ...construction control area: 84,979 square meters, floor area ratio: 2.8, height limit: 60 meters...

- Value Format Errors: The format of attribute values (such as phone numbers, dates, etc.) does not meet standard specifications or expectations.

> **Examples**
>
> ✗: ...Contact number: 010.
>
> ✔: ...Contact number: 010-57365240.

- Numerical Missing: This error refers to the absence of a required numerical figure in a financial document.

Models	Max Tokens	Context Window	Temperature	TopP	TopK
Qwen2.5-7B-Instruct	120,000	32,768	0.7	0.8	20
Qwen3-8B (no thinking)	120,000	32,768	0.6	0.95	20
Qwen3-8B	120,000	32,768	0.7	0.8	20
DeepSeek-R1-0528-Qwen3-8B	120,000	32,768	0.6	0.95	20
Qwen3-14B (no thinking)	120,000	32,768	0.6	0.95	20
Qwen3-14B	120,000	32,768	0.7	0.8	20
GPT-3.5-turbo	18,000	16,384	0.9	-	-
GPT-4o-mini	18,000	200,000	0.9	-	-
GPT-4o	18,000	128,000	0.9	-	-
Fin-R1	120,000	32,768	0.7	0.8	20
Dianjin-R1-7B	120,000	32,768	0.7	0.8	20

Table 4: Parameter Setting for the inference stage.

Examples

✗: The land transfer area of plot JG0404-11 in Dingqiao Unit is square meters.

✔: The land transfer area of plot JG0404-11 in Dingqiao Unit is 30,514 square meters.

- Non-Numerical Attribute Value Missing: This error occurs when a non-numerical attribute (such as names, categories, etc.) misses its corresponding value.

Examples

✗: (1) Procurement project name: ; (2)...

✔: (1) Procurement project name: Office Software Procurement; (2)...

B.2 Financial Domain Knowledge Errors

- Terminology Misuse: Inaccurate or inappropriate use of financial or industry-specific terms, resulting in incorrect or misleading expressions.

Examples

✗: In 2024, the company obtained substantial equity funding through debt financing, thus strengthening its capital structure.

✔: In 2024, the company obtained substantial equity funding through equity financing, thus strengthening its capital structure.

- Incorrect Legal Reference: Incorrect references to clauses, regulations, or legal documents.

Examples

✗: The term "basic medical insurance" in this contract refers to the basic medical insurance stipulated in the Regulations on Government Investment of the People's Republic of China.

✔: The term "basic medical insurance" in this contract refers to the basic medical insurance stipulated in the Social Insurance Law of the People's Republic of China.

- Ambiguous Expression: The use of vague or unclear language that may lead to multiple interpretations.

Examples

✗: The purchaser shall not impose any unreasonable conditions on the winning bidder as prerequisites for contract signing, except in special circumstances.

✔: The purchaser shall not impose any unreasonable conditions on the winning bidder as prerequisites for contract signing.

- Numerical Unit Error: Use of incorrect or non-standard units for numerical values.

Examples

✗: Centralized drinking water source project... planned duration: 180 hours.

✔: Centralized drinking water source project... planned duration: 180 calendar days.

- Omitted Financial Element: Omission of commonly required financial elements, such as bidder qualification criteria, comparative growth rates, etc.

Examples

✗: In terms of pricing, the price of Grade I metallurgical coke at major ports was RMB 2,540/ton.

✔: In terms of pricing, the price of Grade I metallurgical coke at major ports was RMB 2,540/ton, down 3.79% week-over-week.

B.3 Financial Reasoning Errors

- Conflicting Expression: Statements in a document that contradict with each other in meaning or logic.

Examples

✗: The project is not eligible for bidding, and is now open for public bidding.

✔: The project is eligible for bidding, and is now open for public bidding.

- Time Contradiction: Inconsistent or logically conflicting time-related information within a document.

Examples

✗: Submission deadline: May 23... Bid opening: May 22...

✔: Submission deadline: May 22... Bid opening: May 22...

- Numerical Inconsistency: Discrepancies in numerical values cited in different sections of the same document.

Examples

✗: Tax rate: 6%... Tax rate: 8%

✔: Tax rate: 6%... Tax rate: 6%

- Calculation Error: Incorrect numerical computations or total values.

Examples

✗: Annual demand for hip joint systems: 285,995 units (Ceramic-Ceramic: 102,264; Ceramic-Polyethylene: 173,303; Alloy-Polyethylene: 1,042).

✔: Annual demand for hip joint systems: 285,995 units (Ceramic-Ceramic: 102,264; Ceramic-Polyethylene: 173,303; Alloy-Polyethylene: 10,428).

- Clause Conflict : Conflicting stipulations across different clauses or sections of the document.

Examples

✗: Article 10... Under no circumstances shall the contract be terminated early... Article 11... The contract may be terminated early.

✔: Article 10... Under no circumstances shall the contract be terminated early... [Remove Article 11]

An example document in FinED-Bench is shown in Figure 8.

C Details about Manual Annotation

C.1 Details about Annotators

Five financial experts served as annotators for this paper. The entire annotation process was conducted under stringent supervision and scrutiny of the first author of this paper.

C.2 Annotation Tasks and Goals

The purpose of the manual annotation tasks was twofold. The first goal was to obtain a comprehensive annotated dataset that could be used for model evaluation. The second goal was to modify the definitions of errors and error seeds, enabling LLMs to generate error instances that better reflect real-world scenarios and reducing human participation in the data construction process. All the detailed annotation tasks and targets are list in Table 5.

C.3 Annotation Consistency

To ensure the quality of our benchmark dataset, we adopted a majority voting mechanism among five annotators is adopted, with each generated error instance reviewed by three annotators. A high inter-annotator agreement (Fleiss' $\kappa = 0.89$) indicates that the annotations are consistent and of high quality.

C.4 Human Performance on the FinED-Bench

In addition to evaluating model performance, we conduct a human baseline study involving two sophomore students majoring in finance on FinED-Bench. Owing to time and cost constraints, we randomly sample 100 documents for manual evaluation and report their average performance in Table 7. The results indicate that even for human with

Scene

Bond Prospectus

Title

Summary of the Prospectus for the Public Offering of Corporate Bonds by Hangzhou Transportation Investment Group Co., Ltd. to Professional Investors in 2025

Content

…As of ***February 30, 2025,*** there are 80 wholly-owned and majority-owned subsidiaries of the Issuer that are included in the scope of consolidation, the basic situation is as follows:…Jiaoda Tiefa's offering price is 8.81 yuan per share, the offering ***price-to-book ratio*** is 12.6X, and the subscription date is May 27, 2025…

In 2024, the national railway passenger volume will reach 4.085 billion, a growth of **10.9%;**…

| Project | 2024 | 2023 | 2022 |

| national railway passenger volume (billion) | 4.085 **(5.7%)** | 3.865 | 1.610 |

…

Errors


```
[
  { "start_idx": [ 62 ],
    "error_span": [ "February 30, 2025" ],
    "error_type": "Illegal Time"
  },
  { "start_idx": [ 143 ],
    "error_span": [ "the offering price-to-book ratio is 12.6X, " ],
    "error_type": "Terminology Misuse"
  },
  { "start_idx": [ 276, 285 ],
    "error_span": [ "a growth of 10.9%; ",  " 4.085 (5.7%) " ],
    "error_type":  "Numerical Inconsistency"
   }
]
```


Figure 8: An Example in FinED-Bench.

Task	Requirements
Error verification	1. Delete if the sentence is actually correct. 2. Delete if its addition would introduce new errors, for example, injecting numerical unit errors may introduce numerical inconsistency. 3. Delete if its incorrectness cannot be inferred from the context. 4. Ensure that each document contains no more than four errors.
Seeds Update	1. Standardize and simplify the definitions of errors. 2. Update the examples of each error type so that LLMs can generate more qualified error instances.

Table 5: Annotation Requirements for Each Tasks.

relevant domain background, accurately identifying all errors in long financial documents remains highly challenging.

D Details of Supervised Fine-tuning Data

To improve the performance of weaker LLMs in the task of financial error detection, we construct a supervised fine-tuning dataset following the same pipeline as our benchmark, but without human verification. Specifically, after injecting errors, we employ two models (i.e., Qwen3-32B and GPT-4o) as judges to identify both the error type and the error span within the target fragment. An error instance is retained only if both judges correctly detect its span and type. Besides, we preserve the reasoning process generated by Qwen3-32B, located between the <think> and </think> tags.

The resulting dataset contains 9,515 error-free fragments and 8,697 error-containing fragments, each annotated with the error type, error span and corresponding reasoning process. This dataset is used to fine-tune the target LLM, enabling it to acquire relevant knowledge and improve detection capabilities.

Task Description

You are a professional financial document quality reviewer.
Your task is to carefully review the provided document (DOCUMENT) and identify all errors it contains, based on the accompanying List of Error Types with Definitions and Examples.

Error Definitions and Examples

[List of Error Types with Definitions and Examples]:
{{ERROR_DEFINITIONS_AND_EXAMPLES_PLACEHOLDER}}

Information of Document

[Document Background Information]:
Document Title: {{DOC_TITLE_PLACEHOLDER}}
Document Type: {{DOC_TYPE_PLACEHOLDER}}

[Document (DOCUMENT)]:
{{CHUNK_TEXT_PLACEHOLDER}}

Output Format

Please return all identified errors in a JSON list format. Each element in the list should be a JSON object representing a specific error instance.
Each error object should contain the following fields:

- "error_type": (string) The most appropriate error type selected from the List of Error Types.
- "erroneous_text": (list of strings) Contains the exact original sentence(s) or phrase(s) from the document excerpt that contain the error. If the error involves a pair of sentences (e.g., "Inconsistency in numerical values" or "Subject inconsistency"), both sentences should be included. Ensure that the extracted text is accurate and comes directly from the document excerpt.
- "explanation": (string, optional but recommended) A brief explanation of why the identified text is considered erroneous.

If no errors are found in the document excerpt, return an empty JSON list: [ ].
Please ensure that your output strictly follows JSON format, and that the contents of "erroneous_text" are verbatim excerpts from the provided document.

Figure 9: The prompt that guides LLM to perform error detection in financial documents.

E Prompt Details

This section primarily showcases two prompts used for the evaluation (Figure 9) and error generation (Figure 10).

Task Description

You are a financial document analysis expert. Given a piece of financial text, your task is to identify explicit or implicit numerical relationships that can be validated or derived using mathematical formulas. In addition to basic arithmetic expressions (e.g., addition, subtraction), you should also extract more complex formulas, including proportional relationships, ratio equalities, percentage changes, and weighted averages.

Output Format

Each computable relationship should be returned in a structured JSON format. Followings are several examples:
[{"expression": "value1 - value2 = value3"}]
[{"expression": "value1 / value2 = value3 / value4"}]
[{"expression": "(value1 - value2) / value3 = value4"}]

Constraints

- Do not fabricate values. All values must be explicitly present in the original text.
- There may be inconsistencies in units, please convert them.

Examples

Input: The company has cancelled 5.76 million shares (0.41% of the total share capital) that have been repurchased but not yet used, and the total share capital has changed from 1.411 billion shares to 1.405 billion shares.

Output:
[{"expression": "1.411 - 5.76/1000 = 1.405"},
{"expression": "1.411 * 0.41% = 5.76 / 1000"}]

Figure 10: The example of the prompt to find mathematical formulas within a text.

F Additional Experimental Results

F.1 Prompting Strategies

To find an effective prompting strategy for detecting errors in long financial documents, we evaluate several prompt designs, with the results summarized in Table 6. The findings reveal three key observations.

(1) **Whole-document prompting is more effective than chunking-based prompting.** As noted in Section 3.2.2, many generated errors are designed to span the entire document to increase the difficulty of the benchmark. Such cross-document inconsistencies cannot be reliably captured when the text is divided into isolated chunks, making whole-document prompts more effective.

(2) **Multi-error prompting outperforms single-error prompting.** Prompting the model to detect all error types simultaneously ('multi') consistently yields higher F1 scores than prompting for one error type at a time ('single'). This indicates that LLMs benefit from cross-error contextual cues when performing multi-type detection.

(3) **The combination of 'whole + multi + few' prompting achieves the best performance.** This strategy attains the highest average F1 (30.64%) across all tested models, as it provide complete document context, multiple error types, and informative demonstrations.

Strategies	Metrics	whole				chunk			
		multi		single		multi		single	
		few	zero	few	zero	few	zero	few	zero
Qwen3-8B (no thinking)	Pre.	**16.87**	15.70	8.71	7.39	8.00	6.57	13.30	12.78
	Rec.	20.49	19.35	33.82	**36.10**	33.98	35.93	21.30	18.86
	F1	**18.50**	17.33	13.85	12.27	12.95	11.11	16.38	15.24
Qwen3-8B	Pre.	**46.94**	39.07	19.58	13.25	17.19	11.98	39.12	35.83
	Rec.	34.96	33.98	42.76	43.90	43.74	**45.20**	34.80	35.77
	F1	**40.07**	36.35	26.86	20.36	24.68	18.94	36.83	35.80
DeepSeek-R1-0528 -Qwen3-8B	Pre.	**53.27**	44.90	19.09	13.79	18.33	14.21	48.31	43.75
	Rec.	18.54	17.89	17.72	14.96	17.89	17.40	18.54	**20.49**
	F1	27.51	25.59	18.38	14.35	18.11	15.64	26.80	**27.91**
Qwen3-14B (no thinking)	Pre.	21.35	**21.40**	11.12	10.42	10.71	9.73	17.32	18.16
	Rec.	28.78	29.27	39.51	43.25	40.16	**43.41**	31.54	29.59
	F1	24.51	**24.72**	17.36	16.79	16.91	15.90	22.36	22.51
Qwen3-14B	Pre.	**46.04**	41.20	20.63	15.50	18.97	13.16	43.15	34.18
	Rec.	39.67	39.19	49.76	53.50	53.98	53.82	40.98	39.19
	F1	**42.62**	40.17	29.17	24.04	28.07	21.15	42.04	36.51
Avg.	Pre.	**36.89**	32.45	15.83	12.07	14.64	11.13	32.24	28.94
	Rec.	28.49	27.94	36.71	38.34	37.95	**39.15**	29.43	28.78
	F1	**30.64**	28.83	21.12	17.56	20.14	16.55	28.88	27.59

Table 6: Performance Comparisons under Different Prompt Strategies. **Bold** indicates the best performance. Note: (1) whole: the entire document is provided in a prompt; (2) chunk: the document is segmented into small chunks and processed sequentially. (3) multi: the LLM is asked to detect all error types at once. (4) single: the LLM is asked to detect one specified error type per query. (5) zero: no in-context examples are provided. (6) few: a small number of examples are given.

Models	Metrics	General Knowledge Errors					Financial Domain Knowledge Errors					Financial Reasoning Errors					Overall
		IT	RS	VFE	NM	NNM	AE	OFE	UE	TM	ILR	TC	CaE	NI	CE	CC	
Qwen2.5-7B	Pre.	31.03	18.01	21.28	25.0	12.24	0.54	2.5	36.36	4.35	2.13	18.87	6.25	3.16	3.79	18.18	14.37
	Rec.	18.69	24.94	3.97	4.54	1.82	1.54	0.64	15.74	0.87	1.25	3.94	0.6	6.49	1.22	5.63	7.49
	F1	23.33	20.92	6.69	7.68	3.17	0.8	1.02	21.97	1.45	1.57	6.51	1.09	4.25	1.85	8.6	9.85
Qwen3-8B (no thinking)	Pre.	57.34	59.39	11.24	34.75	9.17	0.18	8.93	41.91	12.64	31.25	31.08	0.0	3.24	9.42	26.09	16.03
	Rec.	43.1	22.79	23.41	18.55	3.33	1.54	15.71	42.59	12.79	6.25	18.11	1.2	19.46	7.07	8.45	19.12
	F1	49.21	32.94	15.19	24.19	4.89	0.33	11.38	42.25	12.72	10.42	22.89	0.0	5.56	8.08	12.77	17.44
Qwen3-8B	Pre.	**76.89**	62.11	31.67	65.57	31.39	0.0	**27.12**	71.95	46.96	46.43	73.17	28.89	12.13	46.04	41.94	49.24
	Rec.	63.10	56.05	27.78	36.2	26.06	0.0	20.51	58.80	15.70	16.25	47.24	7.78	17.84	15.61	18.31	33.66
	F1	69.32	58.92	29.60	46.65	28.48	0.0	**23.36**	64.71	23.53	24.07	**57.42**	12.26	14.44	23.32	25.49	39.99
Deepseek-R1-0528-Qwen3-8B	Pre.	70.87	**65.14**	**41.54**	58.39	28.21	0.0	17.95	66.07	52.50	75.00	63.49	35.29	12.12	45.76	41.18	49.92
	Rec.	31.03	26.51	10.71	18.1	10.0	0.0	2.24	34.26	6.1	7.5	15.75	3.59	6.49	6.59	9.86	15.21
	F1	43.17	37.69	17.03	27.63	14.77	0.0	3.99	45.12	10.94	13.64	25.24	0.01	8.45	11.51	15.91	23.31
Qwen3-14B (no thinking)	Pre.	51.84	33.49	19.06	68.07	17.54	**23.18**	16.36	46.32	30.77	3.79	27.02	0.0	10.08	20.17	8.0	25.28
	Rec.	53.2	50.0	39.29	19.61	21.02	**30.72**	30.84	46.58	8.76	6.25	29.26	1.7	17.56	17.31	11.27	29.41
	F1	52.51	40.11	25.66	30.45	19.13	**26.42**	21.38	46.45	13.64	4.72	28.09	0.0	12.81	18.63	9.36	27.19
Qwen3-14B	Pre.	69.7	56.49	31.54	**73.77**	**38.11**	3.45	21.89	**78.7**	**69.23**	52.27	**73.91**	**42.86**	**17.55**	49.42	52.17	**50.77**
	Rec.	71.38	57.67	30.16	40.72	47.58	1.54	18.59	61.57	18.31	**28.75**	33.46	12.57	23.24	20.73	16.90	37.52
	F1	**70.53**	57.08	**30.83**	52.48	**42.32**	2.13	20.1	69.09	28.97	37.10	46.07	**19.44**	**20.00**	29.21	25.53	43.15
GPT-3.5-turbo	Pre.	32.86	0.00	9.38	0.00	25.00	0.00	0.00	7.32	0.00	**100.00**	0.00	0.00	16.67	0.00	14.29	13.14
	Rec.	46.94	0.00	10.91	0.00	1.75	0.00	0.00	16.22	0.00	1.25	0.00	0.00	8.82	0.00	1.41	4.93
	F1	38.66	0.00	10.08	0.00	3.28	0.00	0.00	10.08	0.00	0.02	0.00	0.00	11.54	0.00	2.56	7.17
GPT-4o-mini	Pre.	43.21	11.24	15.48	27.18	30.77	0.00	0.00	23.21	0.00	0.00	23.81	0.00	7.58	**50.00**	**75.00**	21.94
	Rec.	71.43	18.18	23.64	45.16	28.07	0.00	0.00	35.14	0.00	0.00	15.38	0.00	14.71	1.03	8.45	16.61
	F1	53.85	13.89	18.71	33.94	29.36	0.00	0.00	27.96	0.00	0.00	18.69	0.00	10.00	2.02	15.19	18.90
GPT-4o	Pre.	43.02	45.87	17.22	54.92	28.05	0.00	19.17	73.85	60.00	60.47	38.89	6.25	9.49	31.90	56.79	37.76
	Rec.	**90.69**	**82.78**	**49.21**	**74.44**	**68.00**	0.00	**48.57**	**90.00**	**66.67**	27.08	**80.00**	**22.22**	**60.00**	**67.68**	**43.40**	**67.13**
	F1	58.36	**59.03**	25.51	**63.21**	39.72	0.00	27.49	**81.13**	**63.16**	**37.41**	52.34	9.16	16.39	**43.37**	**49.20**	**48.34**
Fin-R1	Pre.	5.54	7.93	0.79	1.36	0.3	0.0	0.32	2.55	0.0	0.0	0.79	0.6	1.08	0.24	0.0	1.9
	Rec.	10.56	14.45	1.52	2.64	0.6	0.0	0.64	4.96	0.0	0.0	1.56	1.18	1.98	0.47	0.0	3.19
	F1	10.56	14.45	1.52	2.64	0.6	0.0	0.64	4.96	0.0	0.0	1.56	1.18	1.98	0.47	0.0	3.19
Dianjin-R1-7B	Pre.	53.96	20.29	18.94	38.67	8.55	8.33	3.2	60.98	18.92	33.33	30.0	18.75	5.65	27.03	13.33	27.25
	Rec.	25.95	19.58	9.92	13.15	3.03	1.54	1.29	28.94	2.03	2.5	15.35	1.8	3.78	2.44	2.82	11.13
	F1	35.05	19.93	13.02	19.63	4.47	2.6	1.83	39.25	3.67	4.65	20.31	3.28	4.53	4.47	4.65	15.81

Table 7: Performance Comparison of different Large Language Models in FinED-Bench across 15 errors categories. (%) **Bold** indicates the best performance and underlined indicates the second-best performance within each metric.

Model	Beginning	Middle	Ending
Qwen3-8B (no thinking)	7.76	6.01	9.11
Qwen3-8B	13.04	9.19	16.75
DeepSeek-R1-0528 -Qwen3-8B	6.52	4.95	7.23
Qwen3-14B (no thinking)	9.32	12.01	16.75
Qwen3-14B	22.67	13.07	20.63
GPT-3.5-turbo	1.55	11.06	3.56
GPT-4o-mini	0.00	0.00	13.70
DianJin-R1-7B	2.48	2.12	5.45
Fin-R1	0.62	0.00	0.63

Table 8: Performance different in potions of errors appear in the document.

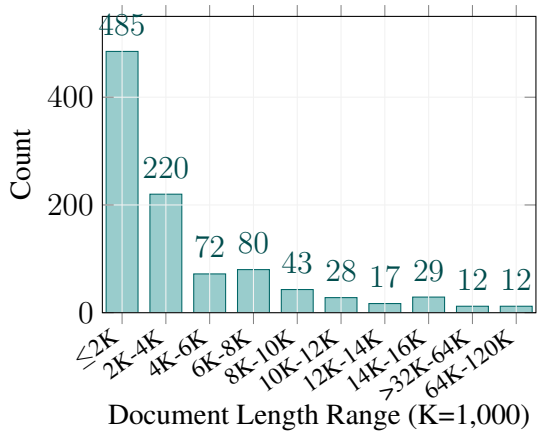

Figure 11: Document length distribution of FinED-Bench.

F.2 Fine-grained Performance on 15 Error Subcategories

We show the performance of all models across 15 subcategories of errors in Table 7.

F.3 Position-aware Error Detection Performance

Besides, we analyze model performance with respect to the positional distribution of errors within a document. Specifically, each document is divided into three segments (i.e., beginning, middle, and ending), as shown in Table 8. The results show that models are more effective at detecting errors located in the ending section, followed by the beginning, while errors in the middle section are the most difficult to identify. This pattern aligns with well-established cognitive phenomena in psychology, namely the Primacy Effect and the Recency Effect. Information presented at the beginning of a document is often repeatedly attended to as the model builds a global understanding of the context, whereas information near the end remains salient due to its proximity to the prediction step. In contrast, content in the middle is less likely to be revisited or emphasized during inference, making errors in this region easier to overlook.

Models	Pre.	Rec.	F1
Qwen3-8B (no thinking)	13.65	24.67	17.58
Qwen3-8B	39.31	38.00	38.64
DeepSeek-R1-0528-Qwen3-8B	54.41	24.67	33.94
Qwen3-14B (no thinking)	17.12	38.00	23.60
Qwen3-14B	47.37	42.00	44.52
Llama3.1-8B	12.50	13.33	12.90
GPT-4o-mini	22.56	20.00	21.20
GPT-5	**57.25**	**50.00**	**53.38**
Dianjin-R1-7B	14.06	6.00	8.41

Table 9: Performance of Different LLMs on English dataset (%). **Bold** indicates the best performance and underlined indicates the second-best performance within each metric.

F.4 Evaluation on the English Benchmark

To address concerns about generalization beyond Chinese, we additionally construct an English financial error-detection dataset sourced from English-language financial websites. This dataset comprises 56 documents with 198 annotated errors, with 91 general knowledge errors, 37 financial domain knowledge errors and 70 financial reasoning errors. The evaluation results are summarized in Table 9. Overall, the findings are consistent with those observed on the Chinese benchmark:

(1) **Reasoning-enabled models substantially outperform their non-reasoning versions.** For example, Qwen3-14B improves from 23.60% → 44.52% F1, and Qwen3-8B improves from 17.58% → 38.64% F1 once thinking is enabled.

(2) **Recall remains a major challenging across models.** Despite improvements in precision and overall F1, recall values generally remain below 50%, indicating persistent difficulty in comprehensively identifying all errors.

(3) **Financial-domain LLMs remain constrained by the capacity of their base models**. Although these domain-specific LLMs are fine-tuned for tasks like financial QA, text summarization, and classification tasks, only a limited amount of domain-specific knowledge is learned.

G Supplements to Related Work

We list financial models and their targeted tasks in Table 10, and existing error detection benchmarks in Table 12.

Models	Base Model	Paras.	Length	Tasks	Techniques	Chinese	Year	Open
Plutus (Peng et al., 2025)	Llama	8B	42000	SMP	IFT	✗	03/03/2025	✔
BloomberGPT (Wu et al., 2023)	BLOOM	50B	2048	SA, HC, NER, QA	PT, PE	✗	03/30/2023	✗
FinMA (Xie et al., 2023)	Llama	7B/13B	4096	SA, HC, NER, QA, SMP	IFT, PE	✗	06/01/2023	✔
InvestLM (Kong et al., 2024)	Llama	65B	4096	SA, HC, QA, Summ	IFT, PE, PEFT	✗	09/15/2023	✔
FinGPT-v3 (Wang et al., 2023)	Llama2	7B	4096	SA, HC, NER, RE	IFT, PE, PEFT	✗	10/12/2023	✔
DianJin-R1 (Zhu et al., 2025)	Qwen2.5	7B/32B	131072	FS, IR	IFT, RL	✔	04/23/2025	✔
Fin-R1 (Liu et al., 2025)	Qwen2.5	7B	131072	FC, FS	IFT, RL	✔	03/22/2025	✔

Table 10: A Summary of FinLLMs. The abbreviations correspond to Para. = Parameters, PT = Pre-Training, PE = Prompt Engineering, IFT= Instruction Fine-Tuning, PEFT = Parameter Efficient Fine-Tuning, RL = Reinforcement Learning; [SA] Sentiment Analysis, [HC] Headline Classification, [NER] Named Entity Recognition, [QA] Question Answering, [SMP] Stock Movement Prediction, [Summ] Text Summarization, [RE] Relation Extraction, [FS] Financial Services, [IR] Investment Research, [FC] Financial Coding.

Dataset	Tasks	Language	Year
FLUE (Shah et al., 2022)	TA, IE, QA	English	10/31/2022
PIXIU (Xie et al., 2023)	TA, IE, QA, FO, RM	English	01/08/2023
FinanceBench (Islam et al., 2023)	QA	English	12/20/2023
BizBench (Koncel-Kedziorski et al., 2023)	QA, IE, TG	English	03/12/2024
FOMC (Shah et al., 2023)	TA	English	2023
FinBen (Xie et al., 2024)	TA, IE, QA, FO, RM, DM	English, Chinese	2024
CFBenchmark (Lei et al., 2023)	IE, TA, TG	Chinese	05/21/2024

Table 11: Comparison of Different Financial Models. The abbreviations correspond to IE = Information Extraction, TA = Textual Analysis, QA = Question Answering, TG = Text Generation, RM = Risk Management, FO = Forecasting, DM = Decision-Making.

Datasets	Domain	Language	Avg. Tokens	Error Types	Year
CoNLL-2014 (Hernandez and Calvo, 2014)	General	English	602.9	grammatical, syntactic	05/2014
BEA-2019 (Bryant et al., 2019)	General	English	244.9	grammatical	08/02/2019
MEDEC (Abacha et al., 2024b)	Medical	English	126.5	semantic	01/02/2025
GMEG (Napoles et al., 2019)	General	English	20.7	grammatical	04/2019
NLPTEA-2020 (Rao et al., 2020)	General	Chinese	35.4	grammatical	12/04/2020
MuCGEC (Zhang et al., 2022)	General	Chinese	38.5	grammatical	07/10/2022
TEC-JL (Koyama et al., 2020)	General	Japanese	21.8	grammatical	05/11/2020
UA-GEC (Syvokon and Nahorna, 2021)	General	Ukrainian	15.9	grammatical	11/08/2022

Table 12: Comparison of Different Error Detection Benchmarks.