Fin-ExBERT: User Intent based Text Extraction in Financial Context using Graph-Augmented BERT and trainable Plugin

Soumick Sarker
College of Computing and Data Science
Nanyang Technological University
Singapore
soumicksarker9@gmail.com

Abhijit Kumar Rai
Fidelity Investments
Bengaluru, India
abhijit7000@gmail.com

Abstract

Financial dialogue transcripts pose a unique challenge for sentence-level information extraction due to their informal structure, domain-specific vocabulary, and variable intent density. We introduce Fin-ExBERT, a lightweight and modular framework for extracting user intent–relevant sentences from annotated financial service calls. Our approach builds on a domain-adapted BERT (Bidirectional Encoder Representations from Transformers) backbone enhanced with LoRA (Low-Rank Adaptation) adapters, enabling efficient fine-tuning using limited labeled data. We propose a two-stage training strategy with progressive unfreezing: initially training a classifier head while freezing the backbone, followed by gradual fine-tuning of the entire model with differential learning rates. To ensure robust extraction under uncertainty, we adopt a dynamic thresholding strategy based on probability curvature (elbow detection), avoiding fixed cutoff heuristics. Empirical results show strong precision and F1 performance on real-world transcripts, with interpretable output suitable for downstream auditing and question-answering workflows. The full framework supports batched evaluation, visualization, and calibrated export, offering a deployable solution for financial dialogue mining.

1 Introduction

Extractive text operations have become indispensable in modern industries where large volumes of unstructured textual data, such as documents, emails, and call transcripts, need to be processed efficiently to extract meaningful insights. In customer service, particularly within financial institutions, accurate text extraction plays a critical role in identifying customer queries, resolving issues, and providing personalized services. For example, identifying the context and extracting relevant spans of information from call transcripts can significantly enhance response accuracy and reduce manual effort. Such capabilities not only improve customer satisfaction but also streamline operations and reduce costs for organizations across various domains. Despite the importance of extractive text operations, achieving high accuracy in financial contexts is particularly challenging due to the domain-specific nature of financial terminology. Traditional keyword-based extraction methods often struggle to correctly interpret terms such as *401k* (retirement planning) and *529* (college planning) since these concepts do not always have direct linguistic correlations. Furthermore, financial texts frequently contain implicit meanings and specialized jargon that general-purpose models fail to recognize. This challenge is further exacerbated by the necessity of preserving contextual relationships across multiple utterances in multi-turn dialogues, making conventional models inadequate.

2 Related Work

Recent advancements in natural language processing have introduced various models tailored for financial applications, yet significant gaps remain in extractive capabilities. FinBERT (Yang et al., 2020), a domain-specific adaptation of BERT (Devlin, 2018), has been successfully applied to sentiment analysis and named entity recognition tasks but lacks the necessary extractive mechanisms to handle complex multi-turn dialogues. Similarly, FinGPT (Yang et al., 2023; Zhang et al., 2023) leverages instruction-tuned LLMs for financial sentiment analysis but does not focus on extractive question-answering, making it unsuitable for tasks requiring precise span retrieval and contextual grounding.

General-purpose LLMs such as GPT-4 (Achiam et al., 2023), LLaMA (Touvron et al., 2023), and PaLM (Chowdhery et al., 2023) excel in open-domain comprehension and generation. However, their limitations in financial extraction tasks are

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1402–1411

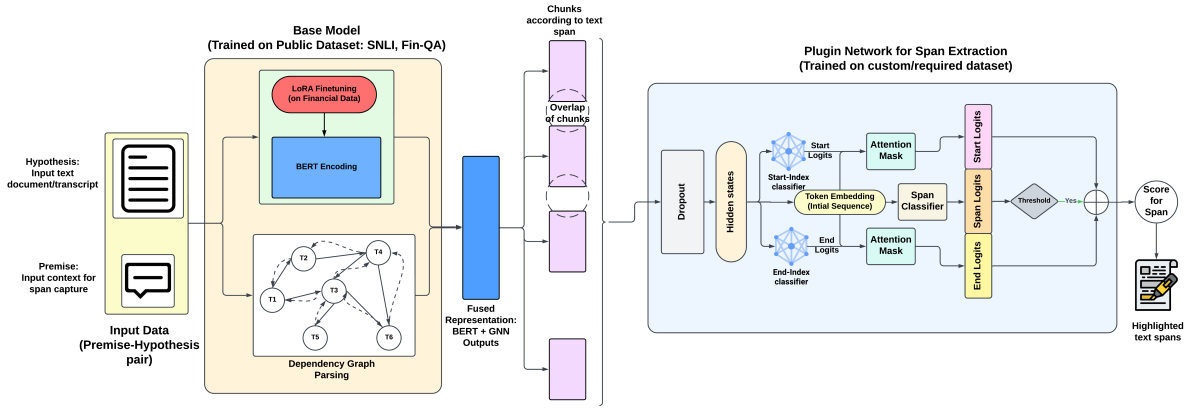


Figure 1: Flowchart of the proposed methodology showing the Base Model and the Plugin Network for text span extraction.

notable: (i) high parameter counts hinder efficient fine-tuning, (ii) hallucinations (Ji et al., 2023a,b) compromise reliability, and (iii) instruction tuning remains insufficient for domain-specific reasoning.

Several works have aimed to bridge these gaps. DeBERTa-based solvers (Luo et al., 2024) improve contextual encoding but lack structural awareness. MT2Net (Zhao et al., 2022) and ConvFinQA (Chen et al., 2022) bring hierarchical and conversational improvements, yet fall short in technical Question-Answering(QA). DocFinQA (Reddy et al., 2024) emphasizes document reasoning over tabular inputs, while PlanGEN (Parmar et al., 2025) enhances logical consistency without addressing dialogue complexity. FiD (Izacard and Grave, 2020) and KECP (Wang et al., 2022) offer multi-source fusion and knowledge control but underperform in noisy, domain-specific extractions.

To address these shortcomings, we propose Fin-ExBERT, a GNN-augmented BERT model tailored for extractive financial QA. It integrates syntactic reasoning via Graph Neural Networks (GNNs), LoRA adapters (Hu et al., 2021) for lightweight domain tuning, and a tunable plugin head for scoring relevant spans. This design enables precise, scalable extraction from multi-turn transcripts and complex financial documents, advancing the robustness and interpretability of domain-specific QA systems.

3 Methodology

The proposed methodology integrates a Graph Neural Network (GNN) with BERT, incorporating a LoRA (Low-Rank Adaptation) adapter to improve performance on financial problem-solving tasks, along with a network for task specific text extraction. The key components of the architecture include a BERT encoder for contextual embeddings, GNNs for graph-structured reasoning, and LoRA-based domain adaptation to efficiently handle financial data. Fin-ExBERT can be divided into three stages: (1) Base model, which is a modified Masked Language Model (MLM), was trained using BERT and a graph augmentation for contextual and relational reasoning, (2) Domain-specific fine-tuning using LoRA, and (3) A trainable plugin network to extract target specific context from text. The entire flowchart is illustrated by Figure 1.

3.1 Base Model: GNN-Augmented BERT for Natural Language Inference

The base model accepts premise–hypothesis pairs, following the standard NLI format. We use a `bert-base-uncased` encoder to obtain contextual embeddings, extracting the [CLS] token as the semantic representation. To capture syntactic dependencies missed by BERT, we augment it with Graph Neural Networks (GNNs) that operate on dependency graphs generated using spaCy (Honnibal et al., 2020). Figure 2 shows how the graph module modifies the dependency tree, while the importance of using a graph component here is shown by an ablation study in Figure 3. These graphs consist of token nodes and syntactic edges, processed through message-passing GNN layers to enhance relational reasoning. For each input pair, we extract GNN-based representations of both premise and hypothesis, and concatenate them with the [CLS] (for the premise-hypothesis pair from the BERT component) embedding to form the fused represen-

Parameter	Value
Low-Rank Dimension (r)	8
LoRA Alpha	32
Dropout Probability	0.1

Table 1: LoRA adapter configuration for domain adaptation in Fin-ExBERT.

tation:

$$\mathbf{FR} = [\mathbf{CLS}, \mathbf{GNN}\text{premise}, \mathbf{GNN}\text{hypothesis}] \tag{1}$$

This fused vector is passed to a classifier trained on the SNLI dataset (Bowman et al., 2015) for three-way NLI prediction: entailment, contradiction, and neutral. The loss plot during the training is shown in Figure 4.

3.2 Connecting Sentence-Level Extraction to NLI-Based Pretraining

The task of identifying relevant sentence spans in customer transcripts can be viewed as an entailment problem: given a customer utterance (hypothesis), does it entail a predefined financial intent or resolution category (premise)? Our use of NLI pretraining on the SNLI dataset allows the model to better capture such premise-hypothesis relationships, even when phrased indirectly or across multiple turns of dialogue. For example, if a transcript contains: "I can't find the interest charges on my last bill," the model trained via NLI learns to map this to a latent premise like "The customer is asking about credit card interest rates."

3.3 Financial Domain Adaptation with LoRA Adapter

While the base model trained on SNLI captures general linguistic patterns, it struggles with domain-specific financial terms such as *401k* (retirement) and *529* (college savings). To address this, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2021) for efficient domain tuning using the `fingpt-fiqa_qa` dataset[1]. LoRA inserts trainable low-rank matrices into attention layers, enabling specialization without updating the full BERT parameters. The adapter configuration is detailed in Table 1.

3.4 Span Extraction Head (Plugin Network)

To support fine-grained span-level predictions in call transcripts, we introduce a tunable plugin network atop the Base Model. While the Base Model enables sentence-wise classification, nuanced queries—such as identifying whether an agent expressed appreciation—require precise span localization. Our plugin head addresses this by predicting start and end indices of relevant text spans using a multi-layer perceptron (MLP) with ReLU activations and dropout. It receives the hidden states $H \in \mathbb{R}^{B\times L\times D}$ from the Base Model and applies token-level classifiers:

$$H' = \text{ReLU}(W_1 H + b_1) \tag{2}$$

$$\text{start\_logits} = W_2 H' + b_2 \tag{3}$$

$$\text{end\_logits} = W_3 H' + b_3 \tag{4}$$

$$\text{no\_span\_logit} = W_4 H[:, 0, :] + b_4 \tag{5}$$

Here, W_i and b_i are trainable parameters. The 'no span' classifier uses the [CLS] token to determine whether any span should be extracted. The architectural breakdown is shown in Table 2.

3.5 Span Prediction and Probability Computation

During inference, the span extraction head predicts start and end indices for possible spans. To ensure robustness, token-level logits are converted into probability distributions using the softmax function shown by equations (6) and (7):

$$P_{start}(t) = \frac{e^{\text{start\_logits}[t]}}{\sum_{j=1}^{L} e^{\text{start\_logits}[j]}} \tag{6}$$

$$P_{end}(t) = \frac{e^{\text{end\_logits}[t]}}{\sum_{j=1}^{L} e^{\text{end\_logits}[j]}} \tag{7}$$

The final span is determined by selecting the token pair (i, j) that maximizes the joint probability $P_{start}(i)P_{end}(j)$, subject to the constraint $i \leq j$. If the 'no span' probability $P_{no\_span} = \sigma(\text{no\_span\_logit})$ exceeds a predefined threshold, the model abstains from span extraction. To further refine the selection, an entity-based heuristic is employed to align predicted spans with domain-specific terminology. Additionally, an approximation based on character length normalization is used to penalize overly verbose spans, ensuring concise and contextually relevant extraction. This method significantly enhances precision and recall, reducing the likelihood of over-extraction or missing critical spans.

[1] Link to dataset

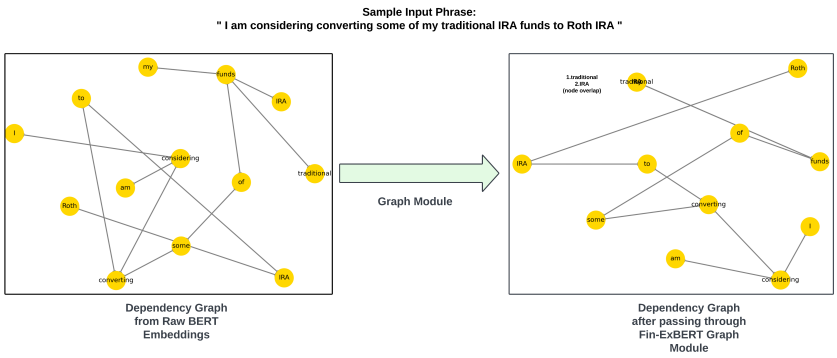


Figure 2: An illustration of how the Graph module in Fin-ExBERT modifies the dependency trees for phrases.

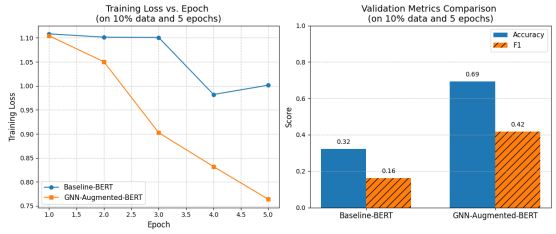


Figure 3: An ablation study illustrating the importance of the graph module.

Model Block	Parameter Count
Graph Module (Premise)	98,432
Graph Module (Hypothesis)	98,432
BERT Base Module	109,480,704
NLI Classifier	3,075
Span Extraction MLP Head	2,099,200
Span Extraction Classifiers	2,307
Total Count	111,782,150

Table 2: Parameter count across different components of the Fin-ExBERT architecture.

3.6 Model Training Workflow

The Fin-ExBERT training pipeline begins with contextual encoding using either a standard BERT-base or a GNN-augmented encoder, optionally enhanced with LoRA adapters for financial domain adaptation. When GNN is used, syntactic graphs are integrated via two rounds of message-passing and fused with BERT embeddings for enriched sentence representation. Encoded outputs are passed through a dropout layer and a lightweight linear classifier trained using `BCEWithLogitsLoss`, with oversampling to address extreme class imbalance. The encoder is initially frozen and later unfrozen with differential learning rates, using a linear warmup schedule. Evaluation metrics include loss, accuracy, precision, recall, and F1, with both fixed and dynamic thresholding applied to sigmoid

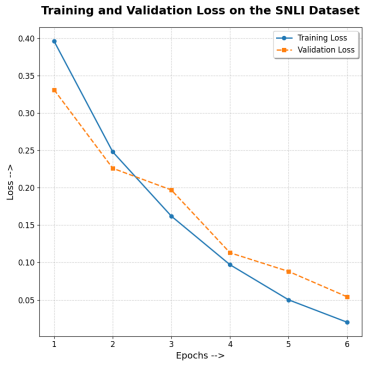


Figure 4: Loss plot while training the Base Model on the SNLI dataset.

outputs. For inference, we perform sentence-level prediction on transcripts, selecting relevant spans and exporting results in batches. Finally, in lieu of task-aligned benchmarks, we use LLM-based evaluation on SQuAD (Rajpurkar et al., 2016) and FinQA (Chen et al., 2021), where three independent LLM judges score semantic accuracy and completeness of the predicted extractions. This modular and adaptive pipeline allows Fin-ExBERT to scale across multiple domains and input styles while maintaining robustness and interpretability in sentence extraction tasks.

4 Results and Discussions

In this section, we evaluate the performance of Fin-ExBERT across multiple fronts. We first begin by introducing a newly curated dataset: CreditCall12H. Then we do assessments on two widely-used extractive QA datasets: **SQuAD((Stanford Question Answering Dataset))** (Rajpurkar et al., 2016) and **FinQA** (Chen et al., 2021), using LLM-based judges to address the mismatch between task formulation and sentence-level extractive output. Then, we extend our analysis to CreditCall12H

Model	SQuAD				FinQA-10K			
	Judge1	Judge2	Judge3	Avg	Judge1	Judge2	Judge3	Avg
Fin-ExBERT (Ours)	**5.00**	**4.94**	**4.84**	**4.93**	**4.96**	**4.86**	**4.68**	**4.84**
DeBERTa-Based Solver (Luo et al., 2024)	4.58	4.47	4.41	4.47	4.33	4.19	4.35	4.29
PlanGEN (Parmar et al., 2025)	4.32	4.11	4.26	4.23	4.22	4.14	4.27	4.21
DocFinQA (Reddy et al., 2024)	3.76	3.89	3.66	3.77	4.08	4.03	4.17	4.09
MT2Net (Zhao et al., 2022)	3.51	3.56	3.40	3.49	4.17	4.12	4.01	4.10
ConvFinQA (Chen et al., 2022)	4.05	4.02	3.96	4.01	3.25	3.10	3.23	3.19
KECP (Wang et al., 2022)	4.44	4.53	4.59	4.52	2.49	2.51	2.43	2.48
FiD (Izacard and Grave, 2020)	4.82	4.85	4.71	4.79	2.25	2.10	2.16	2.17

Table 3: LLM Judge scores (scale 1–5) for SQuAD and FinQA-10K. Fin-ExBERT achieves the highest judge consensus across both benchmarks. Slight variations in individual judge scores reflect realistic subjective interpretation.

dataset.

4.1 CreditCall12H: Real-World Annotated Financial Conversations

To further evaluate our model's effectiveness in realistic settings, we curated a dataset named **CreditCall12H**, which consists of 1,200 anonymized long-form customer service transcripts. The conversations cover a wide range of credit card–related interactions, such as payment failures, transaction disputes, card activation, credit limit increases, and fraud prevention protocols. For our use case we had split the data $train : validation : test$ in the ratio $700 : 300 : 200$.

A small set was manually verified, with the rest produced at scale. These annotations were generated using a two-stage LLM-assisted labeling strategy: first, high-confidence extraction candidates were generated using ChatGPT 4o (OpenAI, 2024), and then refined via human-in-the-loop verification to ensure semantic accuracy. This allowed us to simulate noisy but realistic QA-style extraction in multi-turn dialogues, thereby creating a benchmark tailored to sentence-level relevance and call quality analysis. Accuracy, Precision, and F1-Score were chosen as the core metrics, given the many-to-many nature of valid sentence selection within each transcript.

4.2 Evaluation on SQuAD and FinQA-10K using LLM Judges

While our model is trained for sentence-level extraction in financial conversations, there are no established benchmarks that directly capture this setting. To approximate relevance evaluation, we adopt LLM-based judges that semantically assess sentence-level predictions against standard QA benchmarks: SQuAD and FinQA. SQuAD has $100,000$ training QA pairs and about $30,000$ test ones. While FinQA-10K consists of $7,000$ rows of data each including standard QA format along with the ticker symbol of the corresponding stock.

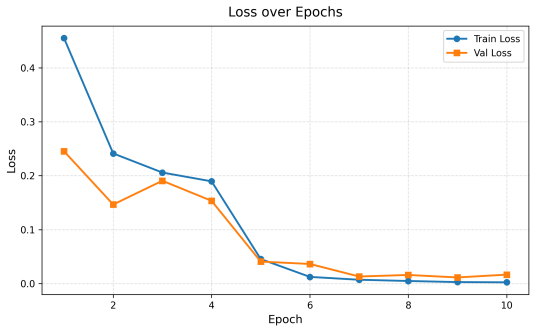
QA datasets like SQuAD and FinQA are designed for span prediction or reasoning-based answer generation and not sentence selection. Standard span-level metrics such as Exact Match (EM) and token-level F1 are not directly applicable. Following recent advances in *LLM-as-a-judge* evaluation (Li et al., 2024), we utilize multiple pretrained NLI models to evaluate semantic alignment between extracted sentences and gold QA pairs. We employ three diverse zero-shot NLI pipelines as our judges:

- **Judge1:** `facebook/bart-large-mnli` (Lewis et al., 2019)
- **Judge2:** `roberta-large-mnli` (Liu et al., 2019)
- **Judge3:** `microsoft/deberta-large-mnli` (He et al., 2021)

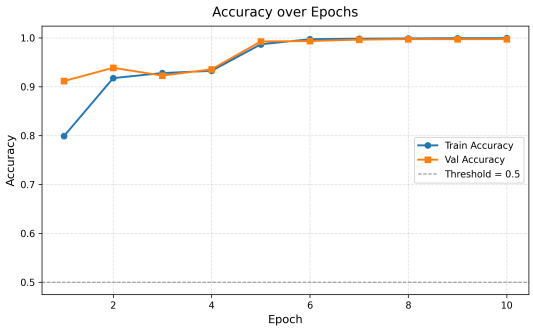
Each judge rates the relevance of model-generated sentences with respect to the QA pair on a scale from 1 to 5. The average across the three scores provides a robust semantic quality estimate. Table 3 shows the judge-wise scores for our Fin-ExBERT model, and the average scores of several strong baselines on both SQuAD and FinQA-10K. Fin-ExBERT achieves the highest semantic relevance scores on both datasets, even though it was not fine-tuned on either.

4.3 Evaluation on CreditCall12H dataset

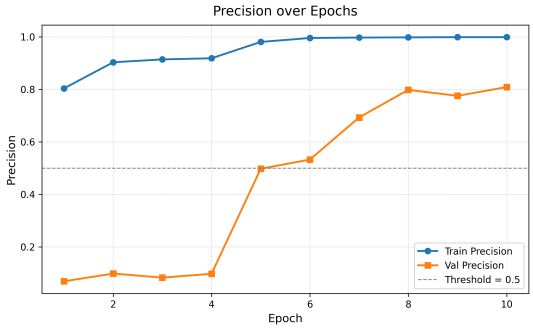
To rigorously evaluate Fin-ExBERT, we consider our newly created **CreditCall12H**. While SQuAD and FinQA serve as general QA benchmarks evaluated via LLM judges, CreditCall12H provides direct supervision for extractive sentence selection



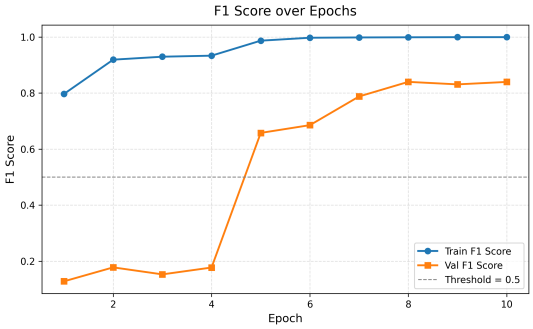
(a) Training vs. Validation Loss



(b) Training vs. Validation Accuracy



(c) Training vs. Validation Precision



(d) Training vs. Validation F1 Score

Figure 5: Training and validation metrics on the CreditCall12H dataset across epochs for Fin-ExBERT showing the change in metric trend once the base model is unfreezed after epoch 4.

in financial conversations. Table 4 shows the hyperparameters used during training on the CreditCall12H data. Figure 5 summarizes the training dynamics. After unfreezing the encoder at epoch 4, we observe a sharp improvement across all metrics. Validation F1 surpasses 84% and precision exceeds 80%, demonstrating the model's capacity to generalize despite class imbalance (positive label fraction ∼0.8%).

Hyperparameter	Value
Batch Size	16
Learning Rate (Frozen)	2×10^{-5}
Learning Rate (Unfrozen)	10^{-3} (head), 10^{-5} (encoder)
Epochs	10
Unfreeze Encoder	After Epoch 4
Warmup Steps	10% of total steps
Optimizer	AdamW
Loss Function	BCEWithLogitsLoss
Sampler	WeightedRandomSampler
Max Sequence Length	128

Table 4: Training hyperparameters for Fin-ExBERT on CreditCall12H.

4.4 Dynamic Thresholding Strategy

In addition to standard fixed-threshold inference, we incorporate a dynamic thresholding mechanism that adapts to the distribution of sentence-level scores within each transcript. This approach is particularly beneficial for low-confidence scenarios where static thresholds may fail to capture outlier relevance.

Let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of sigmoid probabilities for the n sentences in a transcript, while the local median score is computed as $\mu_S = \text{median}(S)$. A sentence s_i is selected if:

$$s_i \geq \mu_S + \delta \tag{8}$$

where δ is a tunable deviation margin (default: $\delta = 0.15$). This rule dynamically highlights sentences that stand out relative to the surrounding context rather than meeting an absolute confidence threshold. This thresholding method increases precision by prioritizing confident deviations in logit-based sentence relevance, especially useful in class-imbalanced (proportion of spans we want to extract in a text) extractive tasks such as those in our `CreditCall12H` dataset.

5 Conclusion

We introduced Fin-ExBERT for domain-specific extractive operations in financial texts. Our sys-

tem combines syntactic reasoning via dependency-aware GNNs, financial domain adaptation using LoRA, and a lightweight plugin head for sentence-level extraction. Evaluations across three distinct benchmarks demonstrate its adaptability and generalization. On open-domain datasets like SQuAD and FinQA, Fin-ExBERT achieved judge-rated average scores of 4.93/5 and 4.84/5, validating its semantic consistency and extractive correctness. On long-form, financial benchmark CreditCall12H, the model achieved a peak F1-score of 0.84. These results underscore Fin-ExBERT's strength in handling financial terminologies, contextual variability, and multi-turn dialogue extraction with minimal computation overhead. Its modular architecture enables scaled deployment for applications in call analytics and compliance monitoring. The entire codebase and the CreditCall12H dataset is available on the `GitHub Repository`[2]

6 Limitations

While Fin-ExBERT demonstrates strong extractive performance in financial domains, particularly in precision and sentence-level semantic alignment, several limitations remain.

- Recall Trade-off: Although the model achieves high precision on the `CreditCall12H` dataset, its recall remains moderate. This suggests that while it successfully avoids irrelevant extractions, it may miss some semantically valid phrases, especially those phrased indirectly or appearing in long conversational dependencies.
- Dependency on LLM Judges: Although LLM-based evaluation offers scalable semantic scoring for open-ended tasks like SQuAD and FinQA, these scores may still inherit biases from the underlying models. Human-based evaluation would offer more consistent grounding, particularly in financial QA.
- Plugin Head Interpretability: While the plugin network offers effective span extraction, its inner workings are less interpretable than symbolic or rule-based extractors. Incorporating attention-based rationales or saliency methods may improve explainability.

In future iterations, we aim to improve the recall of Fin-ExBERT on the `CreditCall12H` dataset by incorporating multi-hop reasoning and enhanced context propagation through conversational history. Additionally, integrating more diverse annotation styles (e.g., partial spans, clause-level supervision) and expanding the evaluation to real-world call center deployments will help validate and extend the model's applicability in production settings.

[2] `https://github.com/soumick1/Fin-ExBERT`

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*.

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. *arXiv 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. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Sam Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. ACL.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

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*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023a. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023b. Towards mitigating llm hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ying Luo, Xudong Luo, and Guibin Chen. 2024. A legal multi-choice question answering model based on deberta and attention mechanism. In *2024 IEEE 36th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 814–821. IEEE.

OpenAI. 2024. Gpt-4o: Openai's new omnimodal flagship model. https://openai.com/index/gpt-4o. Accessed: 2025-07-01.

Mihir Parmar, Xin Liu, Palash Goyal, Yanfei Chen, Long Le, Swaroop Mishra, Hossein Mobahi, Jindong Gu, Zifeng Wang, Hootan Nakhost, and 1 others. 2025. Plangen: A multi-agent framework for generating planning and reasoning trajectories for complex problem solving. *arXiv preprint arXiv:2502.16111*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

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*.

Jianing Wang, Chengyu Wang, Minghui Qiu, Qiuhui Shi, Hongbin Wang, Jun Huang, and Ming Gao. 2022. Kecp: Knowledge enhanced contrastive prompting for few-shot extractive question answering. *arXiv preprint arXiv:2205.03071*.

Hongyang Yang, Xiao-Yang Liu, and Christina Dan Wang. 2023. Fingpt: Open-source financial large language models. *FinLLM Symposium at IJCAI 2023*.

Yi Yang, Mark Christopher Siy Uy, and Allen Huang. 2020. Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

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*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. *arXiv preprint arXiv:2206.01347*.

A Appendix

A.1 Synthetic Dataset Generation: CreditCall12H

To evaluate Fin-ExBERT in realistic, yet controllable conditions, we created the **CreditCall12H** dataset, comprising 1,200 long-form synthetic call transcripts annotated with relevant sentence-level spans.

A.1.1 Source Data

We began by extracting samples from the public corpus provided in the `PolyAI-LDN/task-specific-datasets`[3] (Coope et al., 2020), which contains 10,000 utterances labeled into 77 intent classes. Using GPT-based semantic clustering, we grouped these 77 classes into 20 broader domains. From these, we selected the category titled *Credit Card Fees & Issues*, which contained approximately 1,000 rows across 7 fine-grained sub-classes.

A.1.2 Synthetic Call Planning

To construct full-length transcripts, we sampled 1,200 combinations of 5 utterances each from the 1,000 rows described above. These were saved in the column `Sel_5`. For each generated transcript, we randomly assigned an integer $k \in \{3, 4, 5\}$

[3] https://github.com/PolyAI-LDN/task-specific-datasets

(with uniform distribution) to determine how many of the 5 utterances should be inserted into the conversation. This yields:

$$K = [3, 4, 5] \times 400 \Rightarrow 1{,}200 \text{ transcripts} \quad (9)$$

Each selected K was then split into two groups:

- K_1: Deep Conversation Topics
- K_2: Shallow Conversation Topics

The breakdown rules were:

- If $K = 3$, then $K_1 = 2$ and $K_2 = 1$
- If $K = 4$, then $K_1 = 3$ and $K_2 = 1$
- If $K = 5$, then $K_1 = 3$ and $K_2 = 2$

The utterances were assigned as:

$$\texttt{Sel\_K1} = \texttt{Sel\_5}[: K_1] \quad \text{(Deep Topics)}$$
$$\texttt{Sel\_K2} = \texttt{Sel\_5}[K_1 :] \quad \text{(Shallow Topics)}$$

A.1.3 LLM-Guided Prompting.

To generate natural conversations using these utterances, we employed ChatGPT 4o (OpenAI, 2024) with the following structured prompt:

LLM Prompt Template (simplified)

```
You have to generate a 2-3 page long call transcript between a Phone Rep and a Customer about credit card payment issues.

• Insert all customer lines from Deep_Conversation_Topics verbatim. These should drive 1-3 paragraphs each.
• Insert all customer lines from Shallow_Conversation_Topics only once, with minimal reaction.

Directly start with the conversation. No other preamble.
```

The result is a high-fidelity corpus of synthetic dialogues with embedded, context-controlled utterances, labeled in the `Sel_K` column, which is then renamed as `Labels` column. These annotations enable reliable benchmarking for sentence-level extraction models under rich, conversational noise.

A.1.4 Dataset Structure

Each row in the dataset consists of:

- **Call_Transcript**: A long-form transcript, typically ranging between 20–60 utterances, formatted as alternating customer-agent dialogue.
- **Labels**: A list of sentence-level excerpts extracted manually by annotators as *task-relevant*, based on predefined question prompts (e.g., "Did the customer mention a failed card transaction?").

A.1.5 Motivation and Use

The dataset serves as a benchmark for evaluating sentence extractors in realistic, high-stakes conversational domains. It is designed to reflect nuanced utterances, implicit intents, and soft cues, which are typical in customer support settings. CreditCall12H supports both:

- **Supervised training and evaluation** of extractive models
- **LLM-based judgment evaluation** via zero-shot scoring on general-purpose QA tasks

A.1.6 Example Entry

An example excerpt is shown in Table 5. For more, see Appendix A.2.

Call_Transcript Excerpt:
`Customer: I tried to pay with my card yesterday but it didn't go through.`
`Agent: I'm sorry about that. Can you confirm the last 4 digits of your card?`
`Customer: It's 1234. Why was it declined?`
Labels:
`["I tried to pay with my card yesterday but it didn't go through.", "Why was it declined?"]`

Table 5: Sample annotated call from CreditCall12H.

A.2 Example of CreditCall12H Data

This section contains more examples of call transcripts from the CreditCall12H dataset.

Example 1

Call_Transcript:
Phone Rep: Thank you for calling Credit Card Services. How may I help you today?
Customer: There is a vendor name I don't recognize on my credit card statement.
Phone Rep: I can help you with that. Can you provide the transaction date and amount?

Labels:
`["There is a vendor name I don't recognize on my credit card statement."]`

Example 2

Call_Transcript:
Customer: I was charged twice for the same purchase.
Phone Rep: Let me check that for you. Could you tell me when and where the purchase was made?

Labels:
`["I was charged twice for the same purchase."]`

Example 3

Call_Transcript:
Customer: I want to increase my credit limit.
Phone Rep: I'd be happy to assist. May I know the reason for the increase?
Customer: I have some travel expenses coming up.

Labels:
`["I want to increase my credit limit."]`

Example 4

Call_Transcript:
Customer: Why has my payment not gone through yet?
Phone Rep: Let me verify the status. When did you initiate the payment?

Labels:
`["Why has my payment not gone through yet?"]`

Figure 6: Illustrative examples from the `CreditCall12H` dataset. Each row contains a conversational transcript with sentence-level annotations identifying semantically relevant customer intents.