Adaptive Agent-Based Personalized Contextual Table Summarization

Hithaishi Surendra hsurendr@asu.edu **Shiven Agarwal** sagar147@asu.edu **Poorvi Raddi** praddi@asu.edu **Cheng Guo** chenggu2@asu.edu **L.S Ganesh Rathnam** glalgudi@asu.edu

Abstract

Since the proposal stage, we have made progress in implementing the dynamic layer, achieving an accuracy of approximately 50 percent on the TAT-QA dataset using Gemini 2.5 Flash with zero-shot Chain of Thought (CoT) prompting. In addition, we have started coding two specialized agents: financial and logical. A benchmarking metrics spreadsheet has been created to compare performance between different approaches. Current efforts focus on improving agent functionality and experimenting with different prompting strategies. Moving forward, our goal is to enhance performance through extensive benchmarking, leveraging OpenAI models and the FINQA dataset for further experimentation.

1 Problem Statement

Financial question-answering (QA) tasks require complex reasoning that involves numerical calculations, logical inference, and multi-step problem-solving. Standard large language model (LLM) prompting techniques struggle with these challenges, particularly when dealing with arithmetic operations, table comprehension, and contextual reasoning. These limitations hinder the accuracy and reliability of LLM-based financial QA systems.

To address this, we propose a hybrid agent-based reasoning framework that integrates a dynamic reasoning layer with specialized agents to decompose complex queries into structured reasoning tasks. Specifically, our approach includes:

- A dynamic reasoning layer, which classifies and routes queries to the financial or logical agent.
- A financial agent, optimized for arithmetic operations and quantitative reasoning.
- A logical agent, responsible for inference, reasoning and logical deduction.

Our approach will leverage zero-shot, one-shot, and few-shot Chain-of-Thought (CoT) prompting to improve answer accuracy. Performance will be benchmarked against existing baselines on TAT-QA and FINQA, with systematic comparisons across prompting strategies and LLM architectures.

2 Literature Review

Large Language Models (LLMs) often struggle with complex reasoning tasks that require numerical computations, logical inference, and multi-step problem-solving, particularly in domains like financial question-answering (QA) and planning-based AI applications. Recent advancements in prompting techniques have introduced structured approaches that enhance LLMs' reasoning capabilities to address these limitations. Least-to-Most Prompting enables models to break down complex queries into a sequence of simpler subproblems, improving step-by-step reasoning accuracy (Zhou et al., 2022). Program of Thoughts (PoT) Prompting further disentangles numerical computation from logical reasoning by structuring multi-step reasoning as intermediate programs, enhancing precision in arithmetic and structured decision-making tasks (Chen et al., 2022). Additionally, research on improving Chain-of-Thought (CoT) Prompting has refined reasoning paths by incorporating explicit intermediate steps, optimizing performance in tasks requiring both logical and quantitative reasoning (Wei et al., 2022). These techniques collectively improve LLMs' ability to handle structured problem decomposition. They are well-suited for applications that demand rigorous multi-step reasoning, such as financial QA and AI-driven planning. Financial question-answering (QA) tasks pose unique challenges due to their reliance on numerical reasoning, logical inference, and multi-step problem-solving. Traditional large language models (LLMs) struggle with arithmetic

operations, table comprehension, and contextual reasoning, limiting their reliability in financial domains. This paper explores advanced techniques to enhance LLM performance on financial QA benchmarks such as TAT-QA and FINQA. Specifically, it investigates the effectiveness of Chain-of-Thought (CoT) prompting strategies, including zero-shot, one-shot, and few-shot methods, to improve model reasoning capabilities. Additionally, the study examines agent-based approaches that dynamically decompose complex queries into structured reasoning tasks, integrating specialized agents for arithmetic and logical inference. By systematically evaluating different prompting strategies and architectures, this work provides insights into optimizing LLMs for financial QA, bridging the gap between raw language understanding and structured quantitative reasoning. Beyond financial QA, multimodal question-answering on scientific papers presents additional challenges, requiring models to process and reason over both textual and visual data. The SPIQA dataset introduces a novel benchmark for evaluating LLMs on multimodal scientific question-answering tasks (Pramanick et al., 2024). SPIQA requires models to integrate information from scientific texts, tables, and figures, testing their ability to perform document-level reasoning across multiple modalities. This dataset provides a critical evaluation framework for assessing LLM performance in scholarly domains, contributing to the development of more robust multimodal reasoning systems.

3 Dataset & Benchmarking Status

We have finalized TAT-QA as our primary dataset for initial experimentation. At this stage, our dynamic reasoning layer is achieving approximately 50 percent accuracy on TAT-QA using Gemini 2.5 Flash with zero-shot chain-of-thought (CoT) prompting. To track progress, we have created a benchmarking metrics spreadsheet that captures performance across different models, prompting strategies and evaluation runs.

The development of both financial and logical agents is in progress (nearly complete). We are preparing to begin experiments on the FINQA dataset. FINQA provides a more structured benchmark for financial question answering and will allow us to better assess the performance of the model on tasks that require precise numerical reasoning.

For the next phase, steps involve integrating OpenAI models, exploring agent strategies, and running a comparative study on zero-shot, one-shot and few-shot prompting techniques to further improve accuracy and reasoning.

4 Methodology & Proof of Concept

Following the model design in the project proposal, we have implemented a basic version of the model in python with Google Gemini API.

Components in the system:

- One Gemini-based question classification function serving as the router
- One Gemini-based financial agent to answer questions classified as financial
- One Gemini-based financial agent to answer questions classified as logical

Router details: we use this prompt to classify a question as financial or logical.

```
prompt = f"""
Classify the following question into one of two categories:
- 'financial' (if it involves arithmetic, calculations, or financial terms)
- 'logical' (if it is about comparisons, reasoning, or non-mathematical analysis)

Question: "{question}"

Respond ONLY with 'financial' or 'logical'.
"""
```

Figure 1: Router prompt

Financial Agent details: we use this zero-shot CoT prompt to answer financial questions.

```
cot_prompt = (
    f"""Let's think step by step and perform the necessary calculations.
    If any data is missing, make reasonable assumptions. {prompt}
    """
)
```

Figure 2: Financial Agent prompt

Logical Agent details: we use this zero-shot CoT prompt to answer logical questions.

```
cot_prompt = f"Think logically step by step. {prompt}"
```

Figure 3: Logical Agent prompt

Architecture: We use the router to classify the original question from a given dataset, and the question is either classified as financial or logical. Then if it's financial, the financial agent will answer the question, otherwise the logical agent will answer the question.

Proof of Concept: The pipeline was built up but haven't been evaluated on benchmark metrics. So whether the method is working is unknown for now, but we are working towards it.

5 Preliminary Results and Analysis

The evaluation of Gemini 2.5 Flash on the TAT-QA financial QA dataset was conducted using three agent configurations – Dynamic, Arithmetic, and Logical – under two prompting conditions (zero-shot vs. single-shot Chain-of-Thought). Table 1 summarizes the accuracy results (in percentage) for each agent type with zero-shot CoT prompting (no example, just reasoning prompt) and single-shot CoT prompting (one worked-out example provided). As expected, providing a worked example for chain-of-thought generally boosted the performance of all agents, consistent with prior findings that step-by-step reasoning prompts can improved the accuracy on complex tasks.

Agent Type	**Zero-Shot CoT**	**Single-Shot CoT**
Dynamic Agent	49.6%	60.9%
Arithmetic Agent	54.7%	69.3%
Logical Agent	58.2%	74.1%

Table 1: Accuracy (%) on TAT-QA using Gemini 2.5 Flash under zero-shot vs. single-shot CoT prompting.

While the overall trend shows that one-shot CoT prompting improves accuracy for all agents, there are some limitations and peculiar findings. First, the improvement for the Dynamic agent, although positive, was less pronounced than for the specialized agents. This smaller gain could indicate that without a targeted reasoning strategy, the Dynamic agent cannot fully leverage the single example – a possible sign of its reasoning being too general.

Second, the specialized agents, despite their higher performance, are not without weaknesses. The Arithmetic agent occasionally still produced incorrect answers if a question required an unexpected form of logical reasoning beyond pure calculation, or if it made minor arithmetic mistakes not covered by the example. Likewise, the Logical agent, while strong in inference, showed occasional errors on questions that involved subtle numerical reasoning embedded in text, hinting at the limits of its specialization.

Another observed anomaly was that certain straightforward questions did not benefit much from CoT prompting at all – for example, a simple lookup question where a chain-of-thought is unnecessary sometimes saw the model over-complicating the answer. This highlights that CoT prompting, especially one-shot, is most beneficial for complex multi-step problems, but can be redundant or even introduce confusion for very simple queries.

6 Key Challenges Faced

Throughout the development of our Adaptive Agent-Based Personalized Contextual Table Summarization system, several key challenges have emerged:

Agent Classification Accuracy Designing an effective dynamic reasoning layer to accurately route questions to the appropriate agent (financial or logical) has proven to be non-trivial. Misclassifications can cascade into incorrect or irrelevant outputs, limiting the overall effectiveness of the pipeline.

Prompt Sensitivity and Inconsistency The performance of both the financial and logical agents is highly sensitive to the structure and phrasing of prompts, especially in zero-shot CoT (Chain-of-Thought) scenarios. Minor changes in prompt wording can lead to significant variations in reasoning accuracy and output consistency.

Limited Performance in Zero-Shot Settings Initial testing with Gemini 2.5 Flash in zero-shot CoT prompting yielded only 50

Dataset Complexity Both TAT-QA and FINQA present complex, multi-modal input formats (structured tables + unstructured text), requiring sophisticated contextual comprehension. Effectively aligning table values with textual descriptions remains a persistent challenge.

Evaluation Bottlenecks Although a benchmarking spreadsheet was created, the system has not yet been fully evaluated against standard metrics due to ongoing agent development and the integration of OpenAI models. This has delayed the ability to iterate on model improvements based on quantitative feedback.

Scalability and Adaptability Developing a modular architecture that not only generalizes across diverse datasets but also adapts to user-specific personalization layers is complex. Balancing model adaptability with computational efficiency remains an ongoing concern.

Tooling and Integration Complexity Managing multiple agents, model APIs (OpenAI, Gemini), and fine-tuning logic across modular components has added engineering overhead. Synchronizing these moving parts while maintaining a coherent pipeline has slowed development cycles.

7 Updated Plan and Timeline

Based on the current progress and challenges outlined in the intermediate report, the updated plan focuses on consolidating the agent-based pipeline and enhancing system performance through targeted iterations.

First, the immediate priority is to finalize and evaluate the financial and logical agents by refining their prompt designs using one-shot and few-shot Chain-of-Thought (CoT) prompting strategies. In parallel, OpenAI models such as GPT-4 Turbo will be integrated to compare performance and robustness with Gemini. Once both agents are stable, the dynamic reasoning layer (router) will be optimized with improved classification prompts, including few-shot examples and fallback heuristics to reduce misrouting errors.

Following this, the FINQA dataset will be integrated into the pipeline to expand testing beyond TAT-QA, allowing comparison across datasets and deeper analysis of generalization capabilities. Throughout this process, prompt optimization will continue, including structured experimentation with techniques like Least-to-Most and Program of Thoughts (PoT) prompting. A prompt library will be maintained to track variations and their impact on performance. To support robust benchmarking, the team will automate logging of key metrics such as accuracy, latency, agent type, and error categories, supplemented by a spreadsheet or visualization dashboard.

In the final stages, a basic personalization layer will be implemented, allowing user-specific adaptations such as preferred language tone or summary length. The project will conclude with a comprehensive evaluation, error analysis, and final report summarizing insights, performance gains, and recommendations. If time permits, a short demo video will be created to showcase the system's functionality and reasoning capabilities.

8 Team Member-wise Contribution

- Cheng Guo - Generation of API Keys, SOL Access, Documentation
- Ganesh Rathnam Lalgudi Shivakumar - Generation of API Keys, SOL Access, Documentation
- Hithaishi Surendra - Creation of benchmarking metrics spreadsheet, documentation, background exploration related to testing and benchmarking strategies, coordination.
- Poorvi Raddi - Code implementation for Arithmetic agent model.
- Shiven Agarwal - Code implementation for Dynamic layer agent and Logical agent.

9 Appendix

GitHub

10 Project Plan & Timeline for ref

The project is structured as follows:

Phase	Tasks	Duration
Phase 1	Literature Review, Dataset Selection and Exploration	2 weeks
Phase 2	Models Selection, Baseline Training	2 weeks
Phase 3	Model Fine-Tuning, Hyperparameter Optimization	3 weeks
Phase 4	Evaluation, Error Analysis	2 weeks
Phase 5	Refinement, Final Report	2 weeks

Table 2: Project Timeline

References

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*.

Wei, Jason, Tay, Yi, Bommasani, Rishi, Raffel, Colin, Zoph, Barret, Borgeaud, Sebastian, Yu, Adams Wei, Lester, Brian, Du, Nan, Dai, Andrew M., Xie, and Qizhe et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.