Adaptive Agent-Based Personalized Contextual Table Summarization

Team: Beyond The Prompt

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

Mentor: Abhijit Chakraborty
achakr40@asu.edu

Abstract

Addressing the challenge of complex numerical and contextual reasoning in table-based question answering, this project proposes and evaluates an Adaptive Multi-Agent Framework (AMAF) tailored for financial datasets. We developed two pipelines over the TAT-QA benchmark. The first follows a layered Chain-of-Thought (CoT) approach with specialized modules for routing, logical reasoning, and arithmetic operations and evaluated using Gemini 2.5 Flash—where single-shot prompting consistently outperformed zero-shot, achieving up to 75% F1 on logical tasks. The second pipeline implements a modular AMAF system comprising a Table Agent based on TAPAS for structured value extraction (25% F1), a Context Agent leveraging FLAN-T5 for generating enriched narrative context (29% F1), and a Summarization Agent that uses Mistral 7B to generate user-personalized CoT summaries. The multi-agent design introduces context integration as a core innovation, enabling user-specific summarization through agent specialization and CoT prompting for hybrid table-text reasoning.

1 Introduction

The rapid development of Large Language Models (LLMs) has significantly changed natural language processing. These models show impressive abilities in understanding and generating text for many different tasks (Peng et al.; Matarazzo et al., 2024; 2024). Because of this success, there's growing interest in applying them to more complex information processing challenges (Peng et al., 2024). An important area, and the focus of our work, involves understanding and reasoning about data that mixes tables and text (Zhu et al., 2021). This combination is common in important fields like finance, scientific research, and business analysis (Chen; Zhu; Zhao, 2021; 2024; 2023). Being able to accurately pull information from and summarize this combined data is very valuable for finding deeper insights and making complex information easier to work with.

However, handling this kind of mixed data presents significant challenges that are often difficult for standard LLM designs. While LLMs are good at processing plain text, they frequently struggle to correctly combine the structural information found in tables with the details provided in surrounding text descriptions (Wei et al., 2022). Tasks that need careful, step-by-step reasoning – such as performing calculations based on table numbers, drawing logical conclusions using both table and text, comparing values, or summarizing across different data points – often lead to errors (Zhou et al., 2022). Common problems include factual mistakes, incorrect calculations, or inconsistent reasoning steps (Srivastava et al., 2024). It's crucial that these models are reliable and trustworthy for these tasks, but achieving that reliability remains a challenge (Chang et al.; Laskar et al., 2023; 2024). Furthermore, generating summaries or answers that fit the specific context well, or could even be personalized for different user needs, is another area where improvement is needed and offers room for new ideas (Ghodratnama; Xiao, 2021; 2024).

Motivated by these issues and the need for better, more dependable tools for interacting with complex data, we decided to explore new methods in this project. Our goal was to improve how LLMs handle reasoning over these mixed table-and-text datasets and to develop an approach that is more robust than relying on a single, large model for everything. To achieve this, we investigated the design, implementation, and testing of an adaptive, multi-agent framework. Our core idea was that by using separate, specialized components (agents) – each focused on a specific part of the reasoning process like calculations or logical steps – we could improve the overall accuracy, reliability, and possibly the flexibility of the system when dealing with complex hybrid information.

2 Problem Statement

As introduced earlier, Large Language Models (LLMs) often face difficulties when tasks require detailed reasoning using information presented as a mix of tables and text. This type of hybrid data is common and important, especially in fields like financial analysis (Zhu et al.; Chen, 2021; 2021). While LLMs are skilled at processing language, they frequently struggle with the specific reasoning demands posed by these combined formats.

The core problem we address in this project is the insufficient accuracy and reliability of standard LLMs, even when using common prompting techniques, for questions requiring complex numerical and logical thinking based on these mixed table-text sources. Based on prior research and our own initial findings, we identified several key difficulties:

- **Numerical Reasoning Errors:** Models often make mistakes when performing necessary calculations (like addition, subtraction, or comparisons) using numbers found in tables. This is a well-documented challenge, particularly in financial question answering (Srivastava et al.; Zhu, 2024; 2024).

- **Logical Inference Gaps:** It proves difficult for models to correctly connect information logically across different parts of a table and relate it accurately to statements in the accompanying text (Zhou, 2022). They may fail to perform the needed sequences of reasoning steps.

- **Context Misalignment:** Correctly understanding how textual descriptions map to specific table rows or columns, and using this mapping accurately to answer questions, remains a significant hurdle. Information from one source can be easily misinterpreted or ignored in the context of the other.

- **Lack of Specialization and Adaptability:** Standard single-model approaches usually apply the same reasoning process to all questions, regardless of whether they lean more towards calculation or logical deduction (Yao et al., 2022). This lack of tailored processing can hurt performance. Furthermore, as we discovered during our project, simpler agent-based systems designed to route queries can introduce their own issues, such as misclassifying questions or losing critical context during handoffs.

These combined challenges significantly limit the practical use of current LLMs for dependable analysis or summarization of reports and datasets where tables and text are presented together. Therefore, our work directly targets these reasoning weaknesses. We developed and evaluated a specialized multi-agent approach specifically designed to improve numerical and logical reasoning accuracy for hybrid table-text data, aiming to overcome the limitations observed in both general-purpose models and simpler agent architectures.

3 Related Work

Our research connects to several areas within natural language processing. We primarily look at work on question answering using combined table and text data, techniques for improving how LLMs reason, and systems built using software agents.

3.1 Hybrid Table-Text Question Answering

Researchers have developed various methods for answering questions using both tables and text. For example, some early models like TagOP (Zhu, 2021) used a fixed set of reasoning operations and tried to pick the right one for each question. Others, like FinQANet (Chen, 2021) and UniRPG (Zhou, 2022), focused on generating step-by-step programs or reasoning paths to find the answer. This helped improve numerical reasoning. A strong recent model in this area was TAT-LLM (Zhu, 2024), which used a specific pipeline (Extractor-Reasoner-Executor) and performed well on financial datasets like TAT-QA and FinQA. While these structured approaches are powerful, they often depend on predefined steps or pipelines (Zhou; Zhu, 2022; 2021). This might make them less flexible when dealing with different kinds of questions or unexpected reasoning needs. Our approach is different because we use a team of agents that work together dynamically, rather than relying on one fixed process.

3.2 Enhancing Reasoning with Prompting Techniques

Getting LLMs to reason reliably, especially for complex problems, often requires more than simple prompts. Different techniques aim to guide the model's thinking process. Chain-of-Thought (CoT) prompting (Wei et al., 2022) is a key technique where the model is asked to "think step by step." This often improves performance on reasoning tasks and is a core part of our method. Related ideas include Least-to-Most prompting (Zhou et al., 2022), where models solve easier sub-problems first, and Program of Thoughts (PoT) (Chen et al., 2022), where models generate small pieces of code to help with calculations. These techniques show that guiding the reasoning process helps models perform better on tasks needing careful numerical or logical steps. We use these ideas by building CoT reasoning into our specialized agents.

3.3 Agent-Based Frameworks and Adaptability

Using software agents – specialized modules that can work together – is becoming a popular way to handle complex tasks. For instance, the ReAct framework (Yao et al., 2022) showed that a single agent could improve its reasoning by combining thinking steps with actions, like looking up external information. However, ReAct still uses only one main agent. Other systems like ManyModalQA (Hannan, 2020) used a router to send questions to different expert modules based on the data type (like text or table). While this allows for specialization, it can fail if the router makes a mistake, which was a concern we also had based on our early experiments. Outside of QA, work on adaptive summarization, using interactive feedback (Ghodratnama, 2021) or multiple refinement steps (Xiao, 2024), aims to create outputs that better match user needs or specific contexts (Zhao, 2023).

Our project combines ideas from these different areas. We use multiple specialized agents working together, aiming for more flexibility than single-agent systems like ReAct or systems relying only on initial routing like ManyModalQA. We incorporate structured reasoning methods like CoT within our agents to make them more reliable. Our goal was to create a system better suited for complex table-text reasoning compared to earlier models, taking inspiration from work on generating more adaptive and context-aware responses.

4 Dataset

To develop and evaluate our multi-agent framework, we utilized publicly available datasets specifically designed for question answering tasks involving complex reasoning over hybrid table-and-text data, particularly from the financial domain. Our primary benchmark dataset was:

- **TAT-QA (Tabular And Textual dataset for Question Answering)** (Zhu et al., 2021): This dataset features 16,552 questions associated with 2,757 contexts derived from real-world financial reports. Each context pairs a semi-structured table with relevant descriptive paragraphs. The questions often demand complex numerical reasoning (including operations like addition, subtraction, comparison, sorting) that requires integrating information from both the table and the text. Answers vary in format, including single text spans, multiple spans, or free-form text, sometimes with calculation derivations. Given its challenging nature and alignment with our target problem, we selected TAT-QA as the primary dataset for evaluating our approach's performance.

We also examined the following dataset, primarily for insights into structured numerical reasoning processes:

- **FinQA** (Chen, 2021): Containing 8,281 financial question-answering pairs curated by experts, FinQA's notable feature is the inclusion of fully annotated numerical reasoning programs (step-by-step calculations) for every question. This makes it particularly useful for analyzing how models handle multi-step quantitative logic. It includes standard training, validation, and test splits. While FinQA provides valuable examples of structured reasoning, our main quantitative results reported here are based on evaluations using the TAT-QA dataset.

Preprocessing: In preparing the data for our agent-based system, we applied a preprocessing step. For each data instance, the input was separated into its constituent parts: primarily (i) the relevant information extracted from the table cells and (ii) the associated surrounding textual context. Depending on experimental variations exploring personalization, (iii) other relevant metadata could potentially be included. This separation into distinct information types aimed to facilitate specialized processing by different agents within our proposed framework.

5 Methodology

To address the reasoning challenges over hybrid data discussed in Section 2, we developed and evaluated an Adaptive Multi-Agent Framework (AMAF). The core idea was to use specialized components working together, aiming for better performance than single, monolithic LLMs. Our methodology evolved through experimentation, leading from an initial design based on query classification to a more refined sequential pipeline.

5.1 Initial Approach: Router-Based Agent Delegation

Our first implementation tested a delegation strategy using a router agent paired with two specialized downstream agents, as illustrated in Figure 1. The components were:

- **Dynamic Router Agent:** Implemented using an LLM accessed via the Google Gemini API, this agent's role was to classify incoming user questions into predefined categories, primarily 'financial' (requiring arithmetic or calculation) or 'logical' (requiring comparison, inference, or non-mathematical analysis). The classification used a simple prompt asking the LLM to categorize the question based on these descriptions.

- **Specialized Agents (Financial and Logical):** Two separate agents, also implemented using the Gemini API, were designed to handle the queries routed by the dynamic layer. The Financial Agent was prompted to focus on calculations and numerical processing, while the Logical Agent was prompted to focus on reasoning and inference. Both agents utilized Chain-of-Thought (CoT) prompting (Wei et al., 2022) to encourage step-by-step processing, initially tested in a zero-shot setting (providing only the instruction to think step-by-step).

The workflow involved classifying the question, then passing it to the selected specialized agent for answer generation.

5.2 Rationale for Refinement

While this initial router-based approach allowed for specialization, evaluations and analysis revealed limitations. We observed that the router could misclassify questions, particularly those requiring a blend of numerical and logical reasoning. More significantly, this hard routing step often resulted in the loss of valuable context shared between the table and text, as only the question itself was typically passed on, potentially isolating the specialized agent from the full data needed for accurate reasoning. These observations motivated a shift towards a pipeline architecture designed for better context preservation.

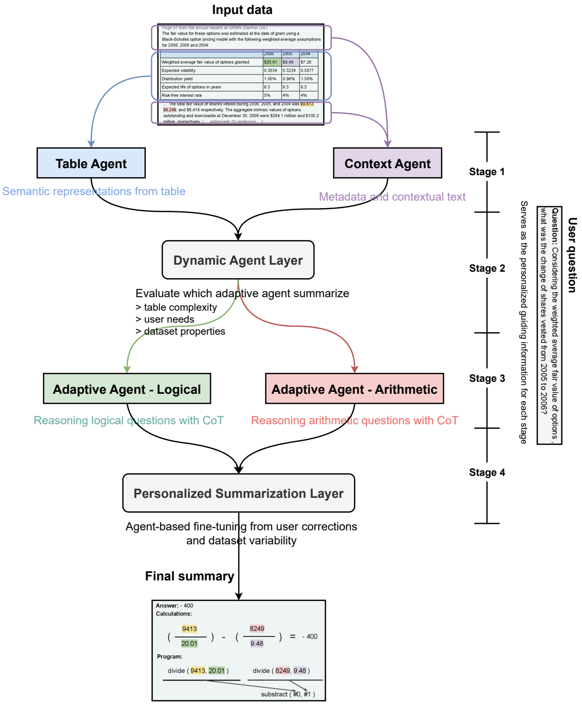
Input data
Page 91 from the annual reports of GRMN (Garmin Ltd.)
The fair value for these options was estimated at the date of grant using a Black-Scholes option pricing model with the following weighted-average assumptions for 2006, 2005 and 2004:
2006
2005
2004
Weighted average fair value of options granted
$20.01
$9.48
$7.28
Expected volatility
0.3534
0.3224
0.3577
Distribution yield
1.00%
0.98%
1.30%
Expected life of options in years
6.3
6.3
6.3
Risk-free interest rate
5%
4%
4%
… The total fair value of shares vested during 2006, 2005, and 2004 was $9,413, $8,249, and $6,418 respectively. The aggregate intrinsic values of options outstanding and exercisable at December 30, 2006 were $204.1 million and $100.2 million, respectively. (… abbreviate 10 sentences …)
Table Agent
Semantic representations from table
Context Agent
Metadata and contextual text
Dynamic Agent Layer
Evaluate which adaptive agent summarize
> table complexity
> user needs
> dataset properties
Adaptive Agent - Logical
Reasoning logical questions with CoT
Adaptive Agent - Arithmetic
Reasoning arithmetic questions with CoT
Personalized Summarization Layer
Agent-based fine-tuning from user corrections and dataset variability
Final summary
Answer: - 400
Calculations:
(9413 / 20.01) - (8249 / 9.48) = - 400
Program:
divide (9413, 20.01)
divide (8249, 9.48)
substract (#0, #1)
Stage 1
Stage 2
Stage 3
Stage 4
Serves as the personalized guiding information for each stage
User question
Question: Considering the weighted average fair value of options , what was the change of shares vested from 2005 to 2006?

Figure 1: Initial router-based multi-agent approach. A router classifies the query and delegates it to one of two specialized reasoning agents.

5.3 Refined Approach: Sequential Agent Pipeline

Our revised approach employs a sequential pipeline where agents collaborate more closely, passing richer information between stages. This architecture, shown in Figure 2, consists of:

- **Table Agent:** Processes the input table to extract relevant structured information based on the query context. Models specialized for table understanding (e.g., TAPAS-based approaches(Herzig et al., 2020)) fit this role.

- **Context Agent:** Integrates the structured output of the Table Agent with the relevant unstructured text from the input, creating a unified context. Models adept at text processing and fusion (e.g., FLAN-T5 based approaches(Chung et al., 2024)) could perform this function.

- **Selectra** is an agent designed to infer user preferences from natural language questions in the TAT-QA dataset, enhancing personalization within the pipeline. It uses the lightweight FLAN-T5 small model to perform user-type classification efficiently. Given a financial question, Selectra prompts the model to choose among predefined categories such as ’financial analyst’, ’business manager’, ’technical expert’, ’novice user’ or ’investor’. This classification is achieved through zero-shot prompting, allowing the agent to generalize to unseen inputs without additional training. By integrating this predicted user type into the reasoning context, Selectra ensures that downstream modules interpret and respond to questions in a manner that aligns with the level of expertise and intent of the user. This agent enables dynamic, context-sensitive adaptation of the pipeline, supporting both user satisfaction and improved task performance.

- **Summarization/Answering Agent:** Generates the final response by synthesizing the combined information from the context agent, performing the final reasoning and formulating the output. This requires strong generative capabilities, found in large language models such as those of the GPT series(Achiam et al., 2023)] or the Gemini family(Team et al., 2023).

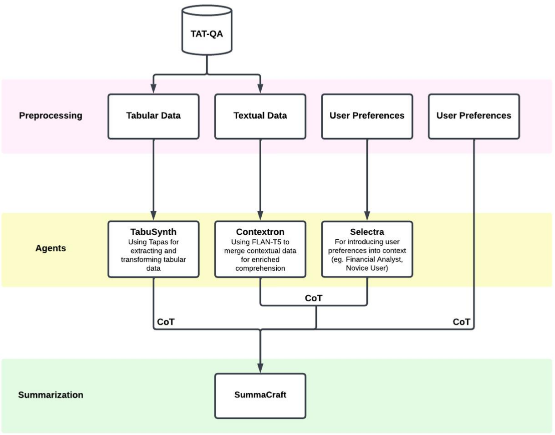
TAT-QA
Preprocessing
Tabular Data
Textual Data
User Preferences
User Preferences
Agents
TabuSynth
Using Tapas for extracting and transforming tabular data
Contextron
Using FLAN-T5 to merge contextual data for enriched comprehension
Selectra
For introducing user preferences into context (eg. Financial Analyst, Novice User)
CoT
CoT
CoT
Summarization
SummaCraft

Figure 2: Refined multi-agent pipeline. Specialized agents process table and text data sequentially, passing structured information and reasoning traces to a final agent for response generation.

5.4 Chain-of-Thought Integration

A key methodological element, applied and refined in both approaches but central to the sequential pipeline, is the use of CoT prompting. In the refined pipeline, the Table and Context agents generate not only their processed data output but also CoT traces detailing their reasoning steps. These traces flow through the pipeline, providing transparency and crucial context to subsequent agents, particularly the final Summarization Agent. Our experiments (detailed in Section 6) systematically compared zero-shot CoT with single-shot CoT (provided a relevant example in the prompt) to evaluate the impact on reasoning accuracy.

The rationale for the refined pipeline focuses on modularity, specialization, and improved context management through the sequential flow of information and reasoning traces, directly addressing the weaknesses identified in the initial router-based design. Implementation and testing relied primarily on large language models accessed via the Google Gemini API.

6 Experiments

We conducted experiments to evaluate the components of our multi-agent framework and understand the impact of different prompting strategies, focusing primarily on the information and results highlighted during our development process.

6.1 Experimental Setup

Our evaluation process was set up as follows:

- **Task:** The experiments centered on question answering using the hybrid table-and-text contexts from the selected dataset.

- **Dataset:** We used the **TAT-QA** dataset (Zhu et al., 2021) for the quantitative evaluations reported here, chosen for its complexity in financial reasoning which aligns with our project goals.

- **Core Model for Reported Results:** The main performance results presented were generated using **Google Gemini 2.5 Flash**, accessed via its API. This was the model used for the experiments evaluating our initial pipeline components.

- **Baselines:** While comparisons against external models like GPT-3.5 Turbo or TAT-LLM (Zhu, 2024) were part of the broader scope, the core experiments we focus on here involved evaluating our own agent designs under different conditions to understand their capabilities and the effects of prompting strategies.

6.2 Evaluation of Initial Router-Based Pipeline Components

Our primary quantitative experiments, performed using Gemini 2.5 Flash on the TAT-QA dataset, focused on evaluating key functional components inspired by our initial router-based architecture (see Section 5, Figure 1). We specifically measured the performance related to:

- The **dynamic routing/classification** functionality (referred to as the 'Routing Layer' or 'Dynamic Agent' component).
- The **arithmetic reasoning** capability (representing the 'Financial Agent' or 'Arithmetic Agent' role).
- The **logical reasoning** capability (representing the 'Logical Agent' role).

For these functional components, we systematically compared two distinct Chain-of-Thought (CoT) prompting strategies:

- **Zero-Shot CoT Prompting:** Agents were given step-by-step reasoning instructions but no specific examples.
- **Single-Shot CoT Prompting:** Agents received the step-by-step instructions along with one relevant example demonstrating the desired reasoning process.

The main goal here was to quantify the performance difference between these two prompting conditions for the different reasoning components.

6.3 Exploration of Refined Sequential Pipeline

For the refined sequential pipeline architecture (see Section 5, Figure 2), which involves distinct Table, Context, and Summarization agents working in sequence, our work during this phase centered more on its design and conceptual development. We outlined the specific roles for each agent (e.g., suggesting TAPAS-based models for table extraction, FLAN-T5 based for context merging, and GPT/Gemini-style models for final generation) and the mechanism for passing CoT traces to maintain context. While qualitative examples demonstrating this pipeline's potential were explored, conducting a comprehensive, quantitative end-to-end evaluation of this integrated system on TAT-QA was identified as an important next step, rather than a completed experiment with reportable metrics for this period.

6.4 Evaluation Metrics

The metrics used or considered for evaluation included:

- **Primary Metric:** For the quantitative results reported for the initial pipeline components (Section 7), the main metric is **Accuracy (%)**. This assessed the percentage of correctly answered questions or correctly classified queries compared to the dataset's ground truth.
- **Other Considered Metrics:** We acknowledged other relevant metrics for deeper analysis or future evaluations, such as **F1-score** (balancing precision/recall for QA), **BLEU** (Papineni et al., 2002) and **ROUGE** (Lin, 2004) for text generation quality, and **qualitative human assessment** for judging reasoning quality.

Our presented results primarily focus on the accuracy achieved by the initial set of agent components under the zero-shot versus single-shot CoT conditions.

7 Results and Analysis

This section presents the findings from our experiments (detailed in Section 6), focusing on the performance of our agent components for both pipelines and analyzing the key findings, primarily based on evaluations using Gemini 2.5 Flash and other models on the TAT-QA dataset.

7.1 Quantitative Results: Initial Pipeline Components

Our primary quantitative experiments compared the performance of key agent functionalities from the initial router-based pipeline (Figure 1) under zero-shot versus single-shot Chain-of-Thought (CoT) prompting conditions using Gemini 2.5 Flash. The accuracy results on the TAT-QA dataset are visualized in Figure 3.

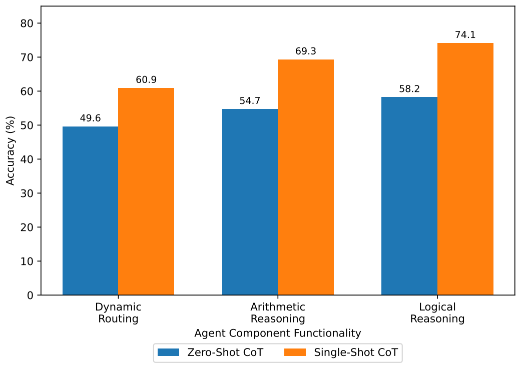

Figure 3: Accuracy comparison on TAT-QA using Gemini 2.5 Flash, evaluating initial agent components under zero-shot vs. single-shot CoT prompting.

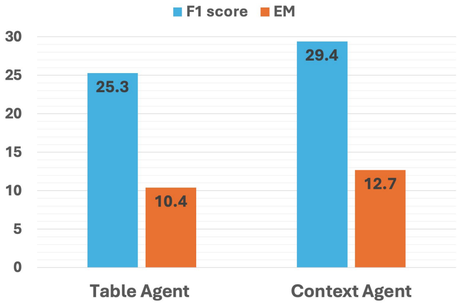

Figure 4: F1 and EM scores for individual agents (Table Agent, Context Agent) developed for the refined sequential pipeline, evaluated on TAT-QA.

Analysis: The results in Figure 3 clearly show that single-shot CoT significantly improved accuracy over zero-shot CoT across all tested components. This aligns with prior work (Wei et al., 2022) on the benefits of CoT exemplars. The specialized reasoning agents (Arithmetic, Logical) saw the largest gains (+14.6pp and +15.9pp respectively). While substantial, the initial zero-shot performance was modest; this can be partly attributed to using relatively small models (Gemini 2.5 Flash) without task-specific fine-tuning, highlighting the challenge of the task from a cold start. Even with single-shot CoT, remaining errors, especially outside an agent's specialization or on simple lookup tasks, indicate the complexity of the problem and limitations of the initial approach. Direct quantitative comparisons against external baselines like TAT-LLM (Zhu, 2024) were not performed.

7.2 Quantitative Results: Refined Pipeline Components

We also conducted evaluations on key individual agents developed for the refined sequential pipeline (Figure 2), specifically the Table Agent (TAPAS-based) and the Context Agent (FLAN-T5 based). Figure 4 presents the F1 and Exact Match (EM) scores obtained for these agents on the TAT-QA dataset.

Analysis: The Table Agent achieved F1/EM scores of 25.3/10.4, while the Context Agent reached 29.4/12.7. These scores reflect the difficulty of accurately extracting structured information (Table Agent) and generating contextually relevant narratives (Context Agent) for this complex task. While metrics like F1 and EM provide standard benchmarks, they may not fully capture the quality of context integration or the nuances of generated text, which can be subjective. Evaluating these contextual aspects is crucial as our novelty emphasizes context integration – an area less directly measured by standard QA metrics used for models like TAT-LLM (Zhu, 2024). Therefore, F1/EM scores serve as necessary, albeit potentially incomplete, indicators for component performance. They provide essential feedback for development and could potentially inform weighting schemes in more advanced evaluation setups (e.g., using a grader LLM). The relatively low scores further suggest the need for better models or fine-tuning for these specific sub-tasks.

7.3 Analysis Regarding Pipeline Architectures

The quantitative results from both sets of experiments inform our understanding of the different pipeline designs. The limitations observed with the initial router-based components (Section 7.1), even with prompting improvements, reinforce the rationale for the refined sequential pipeline (Figure 2). The component results for the refined pipeline (Section 7.2), while modest, provide initial benchmarks for the Table and Context agents. The sequential design, passing CoT traces, aims to mitigate context loss observed in the router approach, potentially improving the synthesis capabilities of the final Summarization Agent (which used Mistral 7B in our setup, as noted in the abstract[cite: 3], though its standalone results are not presented here). While end-to-end quantitative results for the complete refined pipeline are needed, Figure 5 (previously Figure 4) shows a qualitative example illustrating its intended output style.

Input Table Snippet (Contracts):

Type	2018 (M$)	2019 (M$)
Fixed Price	1,146.2	1,452.4
Cost Plus	67.5	68.8
Other	56.7	44.1

Input Context Snippet:

"Other" covers time-and-material deals, which tend to fluctuate...

Illustrative Pipeline Processing:

Table Agent Trace:

Fixed Price change: +$306.2M (+26.7%); Cost Plus change: +$1.3M (+1.9%); Other change: -$12.6M (-22.2%)

Context Agent Trace:

Linked "Other" contracts to time-and-material deals from text.

Summarization Agent Output:

"In 2019, time-and-material ('Other') contract revenues fell $12.6M (-22.2%), while Fixed-Price grew 26.7% and Cost-Plus 1.9%, indicating a possible strategic shift towards more predictable contract types."

Figure 5: Illustrative qualitative example showing the potential output style and synthesized information from the refined sequential agent pipeline.

7.4 Summary of Findings

Based on our experiments and analysis, the key takeaways are:

- Single-shot CoT prompts substantially improve component accuracy over zero-shot CoT for the initial agent designs. Low initial scores are partly due to using smaller models without fine-tuning.
- Individual agent components (Table, Context) in the refined pipeline show modest F1/EM scores, indicating task difficulty and potentially highlighting limitations of standard metrics for evaluating contextual aspects.
- Deficiencies in the router-based design motivate the refined sequential pipeline, which shows conceptual promise for better context handling.
- Further quantitative evaluation, especially end-to-end testing of the refined pipeline and comparison with baselines, is necessary.

8 Shortcomings and Future Work

While our multi-agent framework and the use of Chain-of-Thought (CoT) prompting showed positive results, this work has several limitations and opens avenues for future exploration.

8.1 Shortcomings and Limitations

Our primary limitations include:

- **Incomplete Evaluation:** Peak accuracy on the challenging TAT-QA dataset needs improvement. Our quantitative results focused on components of the initial architecture; the refined sequential pipeline (Figure 2) was designed but not fully evaluated end-to-end. Direct comparisons to state-of-the-art baselines like TAT-LLM (Zhu, 2024) were also not performed.
- **Methodological Aspects:** The initial router-based design had potential weaknesses (misrouting, context loss) which motivated the pipeline refinement. Agent performance also showed sensitivity to prompt phrasing, particularly in zero-shot settings.
- **Limited Scope:** Experiments primarily used one financial dataset (TAT-QA), limiting generalizability. The planned personalization features were not fully developed or tested. Reliance on external APIs also restricted model fine-tuning options.

8.2 Future Work

Key directions for future work are:

- **Pipeline Evaluation and Benchmarking:** Conduct a thorough end-to-end quantitative evaluation of the refined sequential pipeline and benchmark it against strong baseline models on relevant datasets (e.g., TAT-QA, FinQA).
- **Scope Expansion:** Test the framework's generalizability on diverse datasets. Fully develop and evaluate personalization features (Ghodratnama; Xiao, 2021; 2024). Investigate extensions for handling visual data or using techniques like reinforcement learning.
- **Vision-Text Fusion** Incorporate visual cues from charts/graphs (via tools like Pix2Struct or Donut) for document types where tabular data is embedded in rich visual contexts.

Addressing these areas would significantly advance the capabilities of this multi-agent approach.

References

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

Jonathan D. Chang, Kiante Brantley, Rajkumar Ramamurthy, Dipendra Misra, Wen Sun, and et al. 2023. Learning to generate better than your llm.

et al. Chen. 2021. Finqanet: A program-based question answering model for financial text. *Proceedings of EMNLP*.

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*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25:1–53.

et al. Ghodratnama. 2021. Adaptive summaries: Interactive personalization in text summarization. *arXiv preprint arXiv:2104.12345*.

et al. Hannan. 2020. Manymodalqa: Multimodal question answering across text, tables, and images. *Proceedings of ICLR*.

Herzig, Jonathan, Nowak, Peter, Muennighoff, Niklas, Eisenschlos, Julian Martin, Cacheaux, Jordan, Das, Ritasrhree Chatterjee, Clark, and Jonathan et al. 2020. TAPAS: Weakly supervised table parsing via pretraining. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Md Tahmid Rahman Laskar, Elena Khasanova, Xue-Yong Fu, Cheng Chen, Shashi Bhushan TN, and et al. 2024. Query-opt: Optimizing inference of large language models via multi-query instructions in meeting summarization. In *Proceedings of the 2024 EMNLP Industry*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Andrea Matarazzo, Riccardo Torlone, and et al. 2024. A survey on large language models with some insights on their capabilities and limitations.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

B. Peng, Y. Zhu, Y. Liu, and et al. 2024. Graph retrieval-augmented generation: A survey.

Pragya Srivastava, Manuj Malik, Vivek Gupta, Tanuja Ganu, and Dan Roth. 2024. Evaluating llms' mathematical reasoning in financial document question answering. *arXiv preprint arXiv:2402.11194*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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*.

et al. Xiao. 2024. Tri-agent generation: Generator-instructor-editor framework for personalized summarization. *arXiv preprint arXiv:2402.67890*.

Yao, Shunyu, Yu, Zhengxuan, Narasimhan, Karthik, Cao, Yuan, Lei, and Lili et al. 2022. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

et al. Zhao. 2023. Qtsumm: Query-focused table summarization with contextual reasoning. *Proceedings of NeurIPS*.

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*.

et al. Zhou. 2022. Unirpg: Unified reasoning as program generation for table-text qa. *Proceedings of ACL*.

Zhu, Fengbin, Lei, Wenqiang, Huang, Youcheng, Wang, Chao, Zhang, Shuo, Lv, Jiancheng, Feng, Fuli, Chua, and Tat-Seng et al. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

et al. Zhu. 2021. Tagop: A table-and-text based qa model for complex financial reasoning. *Proceedings of ACL*.

et al. Zhu. 2024. Tat-llm: Structured reasoning for table-and-text question answering. *arXiv preprint arXiv:2401.13223*.

A Appendix

A.1 Mandatory Artifacts

- **Google Drive Link:**
  https://drive.google.com/drive/folders/1PQiNhYxcX8DVj86bPiJ19ni0aJ6VW9iZ?usp=sharing

- **GitHub Repository Link:**
  https://github.com/ASU-CSE576-BeyondThePrompt/AdaptiveTableSummarization

A.2 Optional Artifacts

- **Metrics Sheet:**
  https://docs.google.com/spreadsheets/d/1pNLEsnge1vlhOEHfMuEFKqWTQ2v-Bsq1oXQraqcBm0A/edit?usp=sharing