DataMinds: A Generalizable Role-Based Multi-Agent Framework for Automated Data-to-Report Generation

Junjie Xiong Xiangyi Xiao Haipeng Zhang Quan Li *

School of Information Science and Technology
ShanghaiTech University

ABSTRACT

In the Agentic Vis Challenge, the primary goal is to develop agents capable of autonomously analyzing datasets and generating data reports. To meet this objective, we introduce *DataMinds*, a generalizable role-based multi-agent framework that automates the entire process—from data understanding to the generation of visualization reports. Through collaborative interactions among multiple agents, *DataMinds* produces accurate and clear visual reports, enabling users to intuitively grasp the trends and insights within the data. Notably, *DataMinds* is generalizable and not tied to any specific dataset, making it applicable across a wide range of data analysis and visualization tasks in various scenarios.

Index Terms: Large Language Model, Multi-Agent, Data Report.

1 INTRODUCTION

In the Agentic Vis Challenge, the objective is to build a domain-independent agent system that can automatically analyze the *VisPubData*[1] dataset and generate a data report with no human intervention. The challenge lies in automating the entire process, from data comprehension and insight generation to visual presentation, while ensuring the report is accurate, logically coherent, and insightful. To address this, we propose a role-based multi-agent collaboration framework. This framework simulates the task division and cooperation mechanisms of a real-world data analysis workflow, achieving the end-to-end automatic generation of a final report from raw data. The generated report, powered by *GPT-4o*, is available at https://www.visagent.org/api/output/e74393dc-8a36-4681-8c0e-b72fa06434cd

2 DATAMINDS FRAMEWORK

DataMinds is driven by six role-based agents and covers three key stages: *data understanding*, *analysis iteration*, and *report generation*. Through a clear division of tasks and staged collaboration, it effectively manages the entire workflow from initial data comprehension to the creation of a high-quality analysis report.

2.1 Data Understanding

The Data Understanding stage aims to build a comprehensive overview of the dataset. This phase focuses not just on quantitative attributes but also on comprehending the data's content. We introduce a Data Analyst agent to grasp the data's structure and distribution, and furthermore, to uncover the domain context and background knowledge embedded within it. This lays a solid foundation for subsequent analysis tasks.

The ***Data Analyst*** first receives a small sample of the raw data to automatically infer background information and the semantics of each field. It then uses data analysis tools to assess data quality and identify feature distributions, generating a structured data profile. This profile not only serves as a basis for subsequent analysis but also appears directly in the final report. During this process, the agent also identifies and organizes key considerations for handling each field (such as special data types or the need to split multi-entity fields), providing reliable support for later data processing decision.

2.2 Analysis Iteration

The Analysis Iteration stage is designed to continuously generate high-quality analysis tasks and insights through an iterative mechanism. Inspired by real-world workflows, we created three collaborative agent roles: *Task Planner*, *Task Executor*, and *Task Evaluator*. The entire data analysis process is progressively advanced and optimized through a cyclical iteration among these three agents, leading to precise and valuable analytical results. This "plan-execute-evaluate" closed-loop architecture has been applied and validated in several multi-agent frameworks [2, 4].

To ensure the framework's generalizability, we use the *Expert-Prompting* method [5] combined with in-context learning [1] to dynamically generate domain-specific expert personas. These roles are used to construct the three agents in the analysis iteration stage, enabling them to possess domain awareness and demonstrate professional analytical capabilities across different data contexts.

The ***Task Planner***, referencing the data overview and historical analysis results, formulates new analysis questions. To ensure a logical structure that progresses from simple to complex, we guide the *Task Planner* using Gartner's data analysis model [3], starting with descriptive analysis questions and gradually moving toward diagnostic ones. Additionally, the *Task Planner* receives feedback from the *Task Evaluator*, allowing for dynamic optimization of question direction and granularity.

The ***Task Executor*** takes the questions from the Task Planner and generates efficient data analysis code. It uses a code execution tool to perform numerical calculations and statistical analysis, extracting deep data insights from the results. The *Task Executor* also receives feedback from the *Task Evaluator*, which helps it correct potential logical errors and boost the accuracy of its findings.

The ***Task Evaluator*** systematically evaluates the analysis results to ensure their scientific soundness, factual accuracy, and the depth of insights they provide. On one hand, it checks the *Task Executor*'s computation process, identifies potential errors or false information, and provides feedback to optimize the data processing methods. On the other hand, it also retroactively examines whether the questions posed by the *Task Planner* hold sufficient analytical value and provides targeted improvement suggestions, promoting the continuous optimization of question design to guarantee the overall quality and depth of the analysis task.

2.3 Report Generation

The Report Generation stage transforms the outcomes of the previous two phases into a structured report enriched with visualizations that facilitate intuitive understanding of data trends and insights. To accomplish this, we introduce two agents—the *Chart Generator* and the *Report Organizer*—which collaborate to construct the

[1] https://www.vispubdata.org/

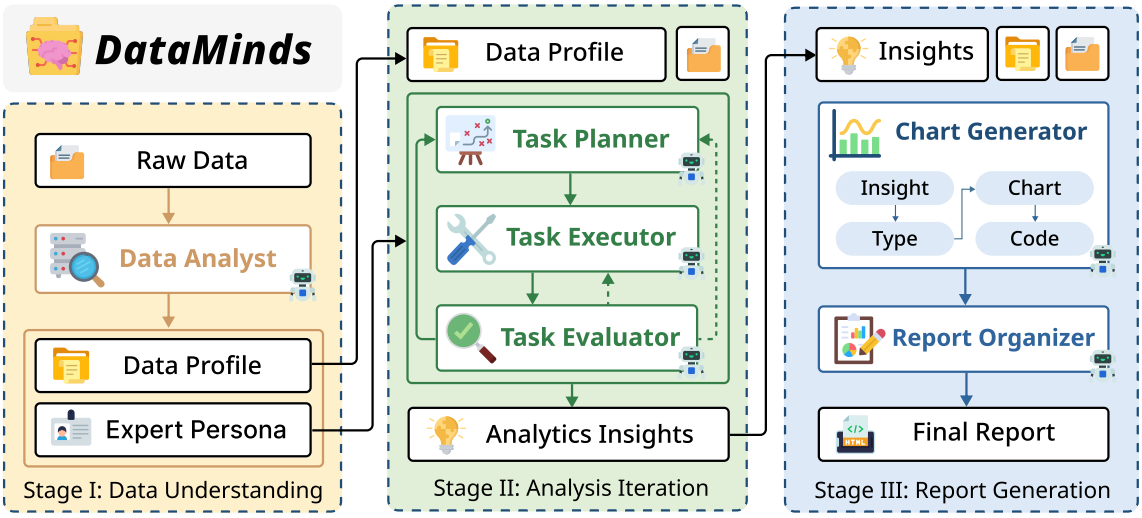

Figure 1: The *DataMinds* framework with three stages: (I) Data Understanding, (II) Analysis Iteration and (III) Report Generation

report. By combining automated visualization generation with content integration, the framework produces reports that are informative, well-structured, and interactive.

The ***Chart Generator*** uses a hierarchical visualization generation mechanism, dynamically adjusting its visualization strategy based on data characteristics and analysis goals. We adopt the *FT Visual Vocabulary*[2] as a reference for chart selection. First, the *Chart Generator* filters the most suitable high-level chart category (e.g., Ranking, Distribution) from nine major visualization types for the current analysis needs. Then, it refines its decision by considering data distribution features, variable properties, and specific display goals to determine the final chart form (e.g.Ordered bar, Boxplot). Once the chart type is determined, the *Chart Generator* automatically writes the corresponding Plotly visualization code to create highly interactive and operable charts. It then uses a code execution tool to generate the corresponding HTML code blocks, which will be used for organizing the final report.

The ***Report Organizer*** is primarily responsible for integrating all generated content—including the data overview, analytical questions, visualizations, and insights—into a predefined HTML template, producing a final report that is clear in structure and coherent in content. During the organization process, the *Report Organizer* focuses on the logical flow between analysis questions. It generates transitional sentences and refines insight statements to make the analysis more methodical and persuasive. At the end of the report, the *Report Organizer* creates a summary to review the overall analysis logic, extract core findings, and highlight key conclusions. This helps users quickly recall the report's main points and enhances the report's completeness.

3 Reflection

During the initial phase of *DataMinds* development, we identified several key issues that were progressively resolved through an iterative process. This section briefly reviews the main challenges and corresponding improvement strategies in different stages.

In the Data Understanding stage, we found that some fields could not be used for direct analysis and required further processing and interpretation. For example, the "Authors" field contains multiple names in a single string, which needs to be effectively split before analysis. Such complex fields not only increase the difficulty of parsing but also easily lead to information ambiguity or misleading insights. To maintain the framework's generalizability, we did not design a fixed pre-processing flow for a specific dataset. Instead, we empowered the Data Analyst to autonomously determine the field processing strategy, organize key considerations for each field's processing (such as splitting needs or type conversions), and pass this information to the next stage's agents. This mechanism effectively enhanced the system's adaptability in diverse data scenarios, providing stable support for subsequent analysis tasks.

In the Analysis Iteration stage, we encountered two particularly prominent challenges. First, the initial agents lacked a clear domain knowledge background, leading to biased interpretations in the generated insights. For instance, when the agent found that InfoVis, SciVis, and VAST had zero publications after 2020, it misinterpreted this as a decline of the fields, overlooking that it was caused by the merger of the conferences. To solve this problem while maintaining system generality, we introduced the ExpertPrompting strategy to dynamically generate "expert agent" roles with domain awareness for each round of analysis, enhancing the agents' ability to understand data context. Second, the task design lacked hierarchy, resulting in a lack of internal logical connection between analytical questions, making it difficult to construct a clear chain of insights. To address this, we leveraged Gartner's data analysis model to guide the *Task Planner* to start with descriptive questions and gradually move to diagnostic questions, establishing a "shallow to deep, layer-by-layer progression" question structure. This significantly improved the logic and value of the analysis process.

In the Report Generation stage, we found that large language models tended to over-rely on common charts(line and bar), making it difficult to adapt to complex data structures and diverse analysis needs. To address the issue, we adopted a hierarchical generation mechanism for chart generation. The *Chart Generator* first determines the high-level visualization category based on the analysis goal, then refines the selection of the specific chart form by considering field properties and display requirements. This significantly increased the diversity and expressive power of the charts.

Despite optimizations, *DataMinds'* efficiency remains average due to the complexity of the task and the impact of multi-agent collaboration. Each analysis takes about 15 minutes and consumes over 400,000 tokens. To control computation costs, users can set an upper limit on the number of iteration rounds. In the future, there remains further room for optimization to improve efficiency.

4 Conclusion

We developed *DataMinds*, a generalizable role-based multi-agent framework designed for generating data reports, to meet the objectives of the Agentic Vis Challenge. *DataMinds* automates the entire workflow, from data interpretation and iterative analysis to final report generation. Through several rounds of refinement and iteration, *DataMinds* has been continuously enhanced, showcasing excellent generalizability and the capacity to deliver high-quality analytical results.

[2] http://ft.com/vocabulary

REFERENCES

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[2] Y. Huang, F. Cheng, F. Zhou, J. Li, J. Gong, H. Yang, Z. Fan, C. Jiang, S. Xue, and F. Chen. Romas: A role-based multi-agent system for database monitoring and planning. *arXiv preprint arXiv:2412.13520*, 2024. 1

[3] I. H. Sarker. Data science and analytics: An overview from data-driven smart computing, decision-making and applications perspective. *SN Computer Science*, 2(5):377, 2021. doi: 10.1007/s42979-021-00765-8 1

[4] Y. Xiao, J. Liu, Y. Zheng, X. Xie, J. Hao, M. Li, R. Wang, F. Ni, Y. Li, J. Luo, et al. Cellagent: An llm-driven multi-agent framework for automated single-cell data analysis. *arXiv preprint arXiv:2407.09811*, 2024. 1

[5] B. Xu, A. Yang, J. Lin, Q. Wang, C. Zhou, Y. Zhang, and Z. Mao. Expertprompting: Instructing large language models to be distinguished experts. *arXiv preprint arXiv:2305.14688*, 2023. 1