Leveraging Agentic Reasoning and Networks to Advance Conversational Data Interfaces In Context of Flood Risk Management Systems

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Varun Pasupuleti

Fall, 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor

Rich Nguyen, Department of Computer Science

Leveraging Agentic Reasoning and Networks to Advance Conversational Data Interfaces In Context of Flood Risk Management Systems

Varun Pasupuleti[a]
[a]University of Virginia School of Engineering, Department of Computer Science, Charlottesville, Virginia

Abstract. This paper explores the development and evaluation of an AI-driven assistant aimed to improve how users interact with and access data within the Floodwatch platform. Building upon previous work, this project introduces agentic reasoning and networks to improve conversational capabilities, personalized responses, and improve overall interaction quality. A user study with 10-15 participants was conducted to assess usability, agentic feature effectiveness, and overall satisfaction. Results show improvements in user experience, with usability rated at 4.8/5 for general inquiries and 4.4/5 for vague queries. These findings reflect the potential of agentic systems in environmental monitoring platforms.

Keywords: Natural Language Processing, Artificial Intelligence, Agentic Reasoning, Agentic Networks, Multi-Agent Systems, Machine Learning, Deep Learning, Flood Prediction, Accessibility, Human-Computer Interaction, Artificial Intelligence Agency.

*Varun Pasupuleti, vjz3qz@virginia.edu

1 Introduction

Floodwatch is a predictive platform for flood risk assessment and weather forecasting. The platform initially featured only a map-based interface but has since expanded with the integration of a natural language processing (NLP)-powered assistant. The first iteration of the assistant was a single-shot agent that relied on prompt engineering and incorporated weather data into the prompt. While functional, it had significant limitations, such as its inability to handle complex queries, adapt dynamically to user preferences, or retain historical context. Building on this foundation, the current project adds agentic reasoning and a network of specialized agents. This addition improves the assistant's ability to handle vague inputs, address complex queries, and tailor responses to user preferences effectively.

Agentic reasoning refers to the process by which an autonomous agent (artificial or human) makes decisions or draws conclusions based on its goals, context, and environmental feedback. This ability allows the assistant to adapt its responses dynamically and improve interactions over time. Agentic networks, on the other hand, are systems of interconnected agents that work together to perform tasks. These networks use distributed intelligence to achieve individual and collective goals, making it easier to process user queries and provide accurate, context-aware responses. These concepts form the foundation of this study, which focuses on their implementation within the Floodwatch platform and evaluates their impact through a user study.

2 Background and Related Work

2.1 Existing Technologies

The original Floodwatch platform used Mapbox Tileset for flood visualization and APIs for weather forecasting. This approach was functional but had limitations in user interactivity. Recognizing these limitations, an earlier version of the platform introduced NLP to handle text-based queries (Barfield Jr., 2024). However, the system was limited in personalization, as it did not retain user preferences or historical conversation history.

Likewise, conventional single-shot assistants often struggle when faced with vague, ambiguous, or multi-faceted queries. Without agentic reasoning or multi-agent collaboration, responses tend to be generic and lack contextual depth. Though challenges such as managing complex queries and delivering personalized responses remained, this version laid the groundwork for a more dynamic user interface. By integrating agentic reasoning and networks, agent systems overcome these limitations and are able to offer more comprehensive and adaptive responses.

2.2 Literature Review

Agentic reasoning and networks emerge as a novel approach to improve conversational data interfaces. These techniques enable systems to dynamically process user inputs and adapt responses based on context. For instance, an assistant with agentic reasoning can interpret vague user queries like "Will it flood tomorrow?" by inferring the user's location from the conversation history. This reasoning is iterative, allowing the system to refine outputs based on feedback and environmental changes (Masterman et al., 2024). Recent advancements in large language models (LLMs) further complement these approaches by incorporating agentic behaviors, such as memory, planning, and action. Techniques like the ReAct pattern (Reason+Act) allow LLMs to function as planners, generating thoughts and actions based on environmental descriptions and goals. Memory-enhanced architectures, such as the RAISE framework, emulate human-like short-term and long-term memory to maintain context and continuity in dialogues (Liu et al., 2024). These techniques significantly improve adaptability in complex conversations.

The current project leverages these concepts by introducing agentic reasoning and memory, enabling the assistant to recall prior interactions and tailor responses effectively. By implementing user personalization, iterative feedback mechanisms, and collaborative multi-agent task execution, the system addresses gaps in earlier iterations. Together, agentic reasoning, networks, and memory-enhanced architectures create a strong framework for building adaptive conversational systems.

3 System Overview

3.1 Agent Configuration

The Floodwatch assistant is built around a modular, agent-based framework. It is designed as an interconnected system of agents, with each specializing in a specific task. This modular design allows individual agents to be updated or replaced without disrupting the system. By distributing responsibilities across the network, the agents collaborate effectively to handle higher-level, complex workflows like understanding vague queries, refining responses, and generating follow-up questions.

Each agent within the Floodwatch assistant serves a specific purpose and contributes to a better user interaction. For example, the Suggestion Agent predicts possible queries by analyzing previous and ongoing conversations. The Understanding Agent processes the last five messages in a conversation to clarify user intent and refine its prompts. The Location and Date Agents extracts geospatial information and temporal information from the user query, respectively. Meanwhile, the Moderator Agent reviews the assistant's responses to make sure they are accurate, relevant, respectful.

Other features include the Agentic Loop, which iteratively refines the assistant's responses based on feedback from the Moderator Agent, and the Follow-Up Agent, which generates context-aware follow-up questions. The assistant also incorporates memory capabilities that allow the assistant to store and recall prior interactions as well as user preferences. Finally, the Response Agent synthesizes information from all other agents and external sources to construct concise, actionable, and contextually appropriate responses.

At the core of the system is the Orchestrator, which acts as a managing script, coordinating the flow of data and interactions between agents. The assistant leverages agentic reasoning to dynamically adapt its responses based on user context, preferences, and feedback.

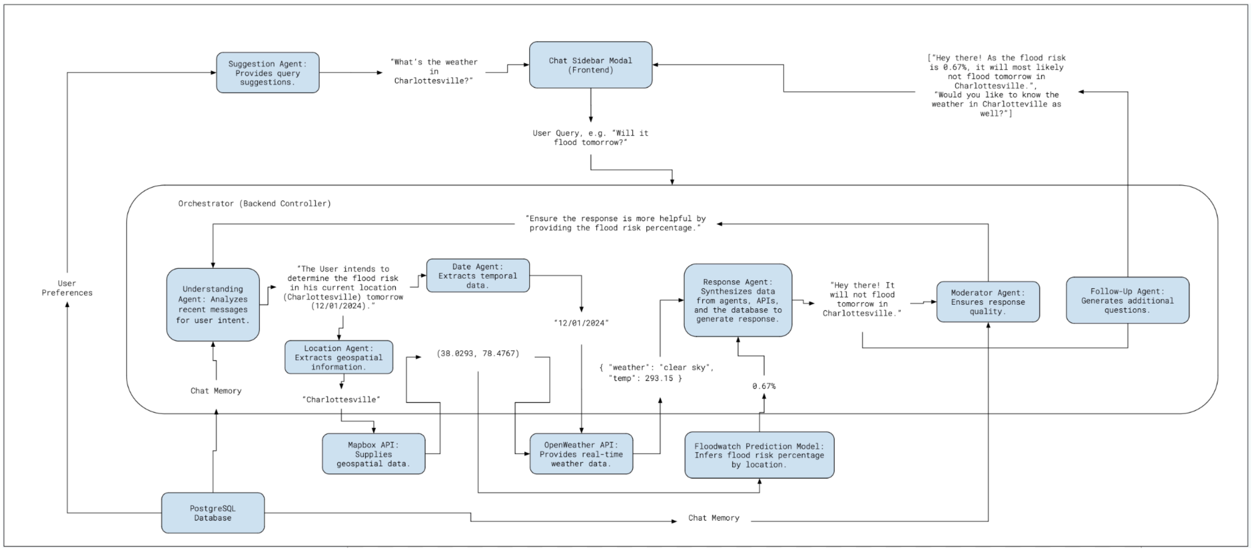

Fig 1 Updated System Architecture of the Floodwatch Assistant. This diagram illustrates the data flow in Floodwatch, from the natural language interface through agents, APIs, and prediction models, to deliver flood risk assessments and weather updates.

3.2 Integration into Floodwatch

The assistant interacts with the Floodwatch backend to fetch real-time weather data, flood risk predictions, and historical records. By interacting with external APIs for real-time weather and flood risk data, the assistant maintains relevance and accuracy in its responses.

3.3 Technologies Used

The assistant is built using TypeScript for type safety and scalability, with Prisma as the ORM for database interactions. It uses a PostgreSQL database to store memory and personalization data, allowing the assistant to recall past interactions and adapt to user preferences. OpenAI's GPT-4 API powers its intelligent conversation capability. The Chat Modal is built as a client component using JSX elements and TypeScript. It updates in real time and offers features like Markdown rendering. The backend is accessed via tRPC API routes to handle chat histories, messages, and suggestions. For external data the assistant integrates with the OpenWeather API for weather data and Mapbox for coordinate data.

4 User Study

4.1 Methodology

To evaluate the assistant's effectiveness, a user study was conducted with 10 participants, with the selected participants representing a mix of technical and non-technical backgrounds. Participants were tasked with interacting with the assistant across three predefined scenarios, each designed to test specific functionalities. The first scenario required participants to provide general weather inquiries in order to assess the assistant's general ability to provide accurate responses based on the user's location and preferences. In the second scenario, participants provided vague, problem-specific queries to assess how well the assistant could interpret ambiguous inputs and generate follow-up questions. The third scenario tested the assistant's handling of contradictory feedback to evaluate its ability to refine future responses. The scenarios aimed to simulate real-world interactions with diverse conditions.

4.2 Surveys

Participants completed surveys before starting, after each scenario, and at the end. The pre-study survey gathered demographic data and baseline expectations for assistant interactions. After each scenario, participants rated the assistant on usability, interaction quality, and the effectiveness of responses using a Likert scale. Questions included prompts such as "How accurately did the assistant understand your query?" and "Was the assistant's response tailored to your needs?" Post-scenario surveys also included open-ended questions to capture qualitative insights about user satisfaction and areas for improvement. Finally, the post-study survey consolidated overall impressions and ranked the most impactful features.

4.3 Results

The results of the user study provided valuable insights into the assistant's strengths and areas for improvement. In the first scenario, which focused on general weather inquiries, participants rated usability highly, with an average score of 4.8 out of 5. Users highlighted the assistant's ability to deliver concise responses and accurate flood risk details as key strengths. The second scenario, involving vague problem-specific queries, received moderate satisfaction scores, averaging 4.4 out of 5. While participants recognized the assistant's understanding capabilities, they noted that the comprehension process for ambiguous inputs could be more robust. In the third scenario, where contradictory feedback was tested, the assistant performed with an average interaction quality score of 3.9 out of 5. Similar to the ambiguous inputs, participants acknowledged the refinement process, though they mentioned that this process could be faster and more precise.

5 Discussion

5.1 Key Findings

The Floodwatch assistant demonstrates significant strengths, particularly in its ability to provide personalized, contextually relevant responses. The Understanding Agent and Suggestion Agent were particularly effective in addressing general user queries and offering tailored suggestions, respectively. Participants showed interest in the memory capabilities that allowed the assistant to recall prior interactions and adapt responses based on stored preferences, though it was not highlighted during the study. While the memory and personalization features improve user interactions, they are still in their early stages and do not fully capture complex user preferences or extensive conversation history in the current implementation. Additionally, the assistant's ability to interpret ambiguous inputs and generate follow-up questions was positively received, although some users stated that this feature could benefit from further refinement to handle contradictory input more efficiently.

5.2 Challenges and Improvements

Despite its strengths, the assistant revealed areas that need improvement. Transparency into the flood risk calculation process was identified as a limitation, as participants expressed interest in better understanding how predictions were generated. Furthermore, providing actionable advice alongside responses, such as safety measures or localized recommendations, could prove effective in production. Participants also suggested incorporating more extensive data metrics, such as historical patterns or projections, to provide a more comprehensive understanding of flood risks. Additionally, there is still room for improvement in refining contradictory inputs, where the assistant could respond with deductive follow-ups to determine the users' intent more effectively. While the user study provided valuable insights, it was limited by a small sample size of participants, all of whom were United States residents aged 20–22. Future studies should aim to increase the sample size and include a more diverse participant pool, varying in age and country of origin.

5.3 Future Directions

Future development of the Floodwatch assistant should focus on a few key areas to improve its capabilities and user experience. One priority is enabling the assistant to update user preferences in real time during conversations. This would make interactions more flexible and tailored to users' changing needs. Another improvement is adding access to interactive visualizations, like live flood data overlays on maps. This would give users a clearer and more intuitive way to explore important information. The assistant could also provide practical advice from web searches, such as tips from local government or disaster response agencies, offering location-specific safety measures. To improve fault tolerance, it could include retries for failed data fetches, making sure information is delivered reliably. Deepening the assistant's reasoning abilities would improve its overall capability to address complex, multi-layered queries. Lastly, the assistant could be tooled with the ability to choose and run the most relevant Floodwatch machine learning models based on the situation and available data. This would optimize performance and ensure accurate responses. Overall, these changes would make the assistant a more comprehensive tool for the Floodwatch platform.

6 Conclusion

The Floodwatch assistant represents a significant advancement in how users interact with the platform. The primary goal of integrating agentic reasoning and networks was to improve the relevance and personalization of its responses and increase platform utility. This study's findings indicate that the addition of memory features, dynamic query handling, and multi-agent collaboration significantly improve the conversation interface of the platform. As such, the broader impact of this study is immense. This approach is not only applicable to flood risk management but also serves as a blueprint for developing intelligent systems in other critical areas, such as healthcare, urban planning, and environmental sustainability. By enabling users to interact with complex datasets through natural language, these systems can democratize access to valuable information, empower decision-making, and ultimately save lives. While the study revealed areas for growth, such as providing actionable advice and expanding reasoning capabilities, the results validate the assistant’s ability to effectively bridge the gap between complex data and user needs. As future improvements are implemented, such as real-time preference updates, web search capabilities, and interactive visualizations, the Floodwatch assistant can become an even more robust tool for disaster management.

7 Acknowledgements

I’d like to thank the Floodwatch team and my principal investigator, Dr. Rich Nguyen, for their valuable guidance and mutual support. I am also grateful to the study participants for their feedback, which was instrumental in evaluating the assistant’s capabilities.

References

Barfield Jr., C. (2024, May 10). AI Agency in Information Retrieval: Advancing Data Access Through Natural Language Interfaces; Enhancing Global Accessibility and Information Flow: Bridging Language Divides with Natural Language Processing and Deep Learning. https://doi.org/10.18130/174e-vh36

Liu, N., Chen, L., Tian, X., Zou, W., Chen, K., & Cui, M. (2024). From llm to conversational agent: A memory enhanced architecture with fine-tuning of large language models. arXiv preprint arXiv:2401.02777.

Masterman, T., Besen, S., Sawtell, M., & Chao, A. (2024). The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. arXiv preprint arXiv:2404.11584.