Open Street Map and Elevation-based Visualization of Evacuation Pathways In Case of Emergent Situations

Luke Lee
Lake Forest Academy
Lake Forest, United States
luke.lee@students.lfanet.org

Rakil Kim
UCLA Henry Samueli School of Engineering and Applied Science
University of California, Los Angeles
Los Angeles, United States
rakilkim0216@gmail.com

Jack Yang
Tandon School of Engineering
New York University
New York, United States
ljy4367@nyu.edu

Rolando Sanchez III
College of Letters & Science
University of California, Santa Barbara
Santa Barbara, United States
rolandosanchez@ucsb.edu

Seungeun Eo
College of Human Ecology
Yonsei University
Seoul, South Korea
eosveunvg@gmail.com

Hayoon Kim
College of Human Ecology
Yonsei University
Seoul, South Korea
hayoonk115@gmail.com

Youngchae Kim
College of Human Ecology
Yonsei University
Seoul, South Korea
yc208007kim@gmail.com

Jin-Kook Lee
College of Human Ecology
Yonsei University
Seoul, South Korea
leejinkook@yonsei.kr

***Abstract*—Personal emergencies and urgent situations requiring immediate evacuation or shelter-seeking support continue to be a challenge for those who need access to reliable and more specific guidance. Current emergency and navigation systems often cannot give the appropriate localized information and require technical expertise or user input, which can delay the individual's response time. This research proposes an LLM-based emergency navigation system that combines data from multiple sources, including OpenStreetMap and Google, to assist users in finding shelters and evacuation routes. The system employs the LangChain framework with geospatial APIs to orchestrate a variety of tools to coordinate data processing, pathfinding, and visualizations to give contextual guidance through natural language interaction, not requiring prior technical knowledge. By combining geospatial data and AI agents with a conversational interface, this paper addresses the need for a more straightforward, personal assistance during emergent situations.**

Keywords—Agentic AI, Emergency Response, Rapid-Onset Emergencies, Geospatial Visualization, Tool Orchestration

I. Introduction

Rapid-onset emergencies – such as natural disasters, armed conflicts, and sudden accidents – continue to cause preventable harm even though, in the case of natural disasters, the total number affected is declining globally [1]. Studies show that in these individual emergent situations, rapid discovery of safe, nearby resources is crucial for alleviating casualties, especially in urban areas where existing services are often underutilized due to confusion or misinformation. [2, 3].

Several efforts have aimed to develop emergency response solutions using Large Language Models and agentic systems. For instance, Geode – a zero-shot geospatial Question-Answering(QA) system – was able to enhance the LLM model's ability to process real-time geospatial data to address multi-modal, spatial-temporal queries [6]. Moreover, WildfireGPT, a RAG-based multi-agent system, demonstrated how LLMs can coordinate in real-time to provide expert-level guidance during fast-evolving wildfire scenarios by synthesizing hazard data, scientific reports, and user queries into actionable advice [7]. A recent review of the potential of artificial intelligence (AI) to understand extreme climate events has also shown the importance of AI resilience and real-time data integration in extreme-weather contexts [8]. While these solutions excel at macro-level analysis or expert-focused QA interfaces, they generally do not support the rapid, conversational discovery of nearby emergency shelters or resources when individuals are on the move. Moreover, as identified by Camps-Valls et al., a limitation in disaster-AI systems is the lack of timely, localized, and trustworthy data, particularly during the first critical hours of a crisis [8]. This proves the importance of an AI system that is rapid, autonomous, and user-centered in its design.

Hence, this paper introduces a proactive agentic system that automatically activates on server startup. Unlike prior systems that depended on user queries, this system assumes emergency context by default and accesses public sourcesic response to emmergencies – such as RSS feeds, government alerts, and social media – for detecting real-time disasters. This allows users to start an application and prepare for an emergency while the system is processing. Using natural language processing, geospatial APIs, and semantic similarity models, nearby resources such as shelters, medical services, and emergency sites are ranked and shown. Users can further refine results based on their context (e.g., group size, pet accommodation) by providing feedback to the model, which then the system updates its recommendations accordingly.

Furthermore, supplemental geospatial data, such as elevation, is used to identify steep areas for more specific guidance.

II. Background

While casualties from large-scale natural disasters have dropped due to improved infrastructure and forecasting [1], emergencies like wildfires, flash floods, and heatwaves continue to cause preventable harm, especially in urban areas. For instance, heatwaves now cause more deaths in the United States than hurricanes and floods combined [2, 3].

For these rapid and hazardous crises, rapid access to trustworthy, localized, and real-time information is crucial, but factors like internet outages, communication breakdowns, and misinformation hinder effective evacuation for survival [4]. This is especially critical in urban settings since emergency services exist but remain underutilized due to confusion, inaccessible information, or lack of digital access, for disproportionately affected marginalized groups such as the elderly and low-income communities [3]. While Static information provides necessary information for evacuation, these are often outdated, require users to have prior knowledge [4], and demand time for research, hence being a difficult source to rely on during stressful, anxious emergencies [5].

Recent progress in the development of emergency-aware AI has introduced promising tools for disaster response, such as systems leveraging geospatial reasoning and multi-agent frameworks to process real-time hazard data [6], [7]. However, as Camps-Valls et al. note, most Crisis-Response AI systems struggle with data reliability and robustness during the critical early hours of a crisis, particularly under degraded conditions like network disruptions [8]. Moreover, the lack of intuitive, conversational capabilities with fast response speed to guide individuals seeking nearby emergency resources urges the development of one.

To that end, this study integrates LLMs, semantic search, and real-time geospatial data (e.g., via APIs like Google Maps and OpenStreetMap) for a rapid, autonomous, and user-centered emergency response system that automatically detects crises, infers user intent, and recommends nearby life-saving resources without requiring user input. This would allow non-expert users to quickly receive ranked, location-specific recommendations based on semantic relevance and geographic proximity without needing any technical expertise. In addition to automatically initiating to provide immediate support, the chat interface allows for supplemental queries to provide more personalized outputs. Since existing platforms rarely combine these capabilities in an accessible way to diverse populations during crises, the development of such a system will enhance access to life-saving resources.

III. System design

The proposed agent-driven emergency navigation system is designed to provide real-time, context-aware recommendations during sudden-onset emergent situations. To that end, the proposed system addresses recent advances in agentic AI, which enable autonomous agents to reason about complex problems and follow multi-step workflows with tool orchestration [9]. Built using the LangChain framework, it allows LLMs to interact with various tools involving APIs, databases, and other systems for building automated systems [10].

A. System Overview

The system interacts with the user through a web-based application using Python's FastAPI framework, utilizing a LangChain-based agent for the core reasoning engine. The server immediately and automatically responds to disasters as soon as they start, while the web user interface allows for further querying for user feedback. The agent coordinates tool invocations for location inference, data collection, and CSV export, with user feedback (e.g., mobility constraints, organization size) dynamically incorporated into the decision loop and geospatial visualizations to advocate evacuation. The agent is capable of employing static datasets (e.g., public CSVs of emergency shelters) with dynamic geospatial data from Google Maps Places API, Google Maps Elevation API, and Overpass API from OpenStreetMap (OSM). The Google APIs are used for their data-richness and reliability. OSM was adopted for the free access it gives to diverse, community-driven datasets on various geospatial features that are difficult to acquire from commercial platforms. Specifically, the agent manages accessibility-related features such as stairs, ramps, and elevators.

B. Data Architecture

Two primary data sources are used: Static Local Data and Real-Time API Data. Static Local data is a curated CSV file containing evacuation shelters in Seoul City, which is used as a foundational source for the proposed system. The CSV file includes name, address, coordinates, capacity, accessibility, and review metadata. For the social media context of constructing an emergency, AI-generated texts were included. For Real-Time API Data, the Google Maps Places API is utilized and produces location results based on user location and search intent. Both data sources are cross-validated by resource category and availability when the system initiates. Metadata such as ratings, reviews, and open status enhance semantic relevance scoring. Each data source undergoes validation and schema normalization to ensure consistent fusion during ranking. OSM's Overpass API and Google Maps Elevation API are used for visualizing valid routes to the shelters, with steep areas that lack accessibility features marked for additional guidance.

C. Modular Tools

The proposed system orchestrates a single or a group of modular tools for different purposes, as described in Table I.

I. Modular Tool Group and Function of the System

Modular Tool Group	Function
User Input Parsing	The agent receives a natural language query, along with optional location coordinates and feedback.
Location Inference	If location is missing or ambiguous, infer_location uses LLM reasoning or defaults to saved coordinates.

Emergency Context Construction	On initialization, a sub-agent detects RSS feeds, tweets, and alerts to create an emergency context around the user(e.g., flood or heat alerts in the user's city).
Query Refinement	*extract_query* isolates the type of resource needed and *rephrase_query* reformulates vague or colloquial input into survival-critical queries.
Resource Search	*csv_resource_search* prioritizes local static datasets, and *geospatial_search* supplements with dynamic Google Maps API results. Both sets are deduplicated by name and address and ranked by proximity.
Feedback Input	The tools *determine_resource_limit* and *suggest_keywords* take user feedback (e.g., company size, accessibility needs) and re-operate the Resource Search if the user updates feedback. If the user selects a destination, *calculate_shortest_distance* estimates walking time and distance. Otherwise, the default output location is printed.
Explanation Generation	The agent returns a short response explaining why specific resources were chosen, tailored to the query and context using LLM.

D. Application Pipeline

Fig. 1 represents the overall design of the proposed system, which emphasizes agent-driven flow with a modular decision group that dynamically adapts to user input and contextual data.

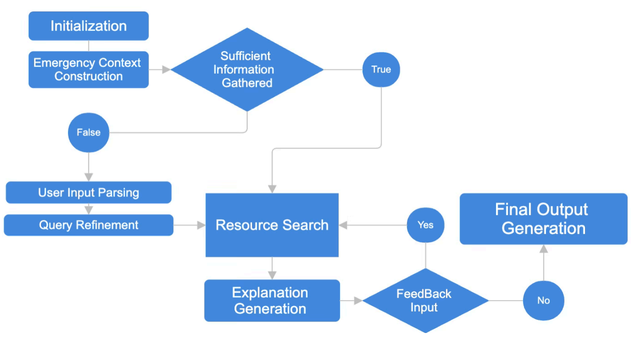


Fig. 1. Agentic Control Flow for Emergency Recommendation System

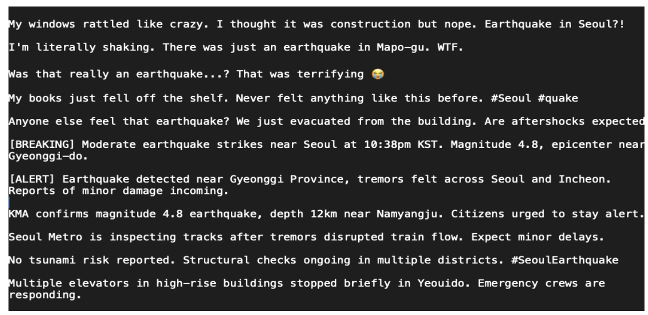
The following steps sequentially occur: the server initializes and assumes an emergency has occurred, initiating Emergency Context Construction. This step stimulates information gathering using Twitter posts and RSS feeds. When the number of keywords relevant to the emergency gained through scraping is equal to or larger than the minimum alert threshold required for activation, the system directly initiates Resource Searching through the local CSV file and Google Map. In the case of a smaller number of keywords, the system falls back to manual mode. User Input Parsing and Query refinement of the user's input are done to extract the user's situation and intent. After that, Resource Search initiates. Then, Explanation Generation operates to provide a short reason for the selected locations. If additional feedback is inserted, the user input is updated, and Resource Search re-operates to find a place fitting the updated user input. Otherwise, short reasons and selected locations are generated as an output, and the loop terminates.

IV. Demonstration

To evaluate the proposed system's performance, an earthquake simulation in Seoul was created. Manual mode represents user user-input-dependent situation, which requires a query that provides base information about the user's emergency. Two cases were conducted, where the minimum threshold of alert of 10 was set as the critical point for activating manual mode or not. The setting of the threshold value was controlled to test two cases: one simulating sufficient information gathered and one not simulating sufficient information.

A. Scenario Setup

A fake earthquake was simulated by using realistic mocked tweets and RSS feeds generated by ChatGPT, as shown in Fig. 2.


My windows rattled like crazy. I thought it was construction but nope. Earthquake in Seoul?!

I'm literally shaking. There was just an earthquake in Mapo-gu. WTF.

Was that really an earthquake...? That was terrifying 😭

My books just fell off the shelf. Never felt anything like this before. #Seoul #quake

Anyone else feel that earthquake? We just evacuated from the building. Are aftershocks expected

[BREAKING] Moderate earthquake strikes near Seoul at 10:38pm KST. Magnitude 4.8, epicenter near Gyeonggi-do.

[ALERT] Earthquake detected near Gyeonggi Province, tremors felt across Seoul and Incheon. Reports of minor damage incoming.

KMA confirms magnitude 4.8 earthquake, depth 12km near Namyangju. Citizens urged to stay alert.

Seoul Metro is inspecting tracks after tremors disrupted train flow. Expect minor delays.

No tsunami risk reported. Structural checks ongoing in multiple districts. #SeoulEarthquake

Multiple elevators in high-rise buildings stopped briefly in Yeouido. Emergency crews are responding.

Fig. 2. Mocked Tweets and RSS feeds for Mimicking an Earthquake in Seoul, South Korea.

To mimic real-time social responses to an emergency, these tweets and RSS feeds illustrate diverse responses: human-like, informal reactions, shaking reports, evacuation notices, infrastructure impacts, and safety warnings. Moreover, the system also utilized a local CSV file of earthquake evacuation shelters from South Korea's open data portal. The CSV file was composed of metadata such as name, capacity, and accessibility, consistent with geolocated shelters within Seoul.

B. Case 1: Simulation with Sufficient Information

To evaluate the system under realistic emergency conditions, a simulated earthquake scenario in the Seoul metropolitan area was tested. The system was initialized with a corpus of mocked earthquake-related tweets and RSS feed entries to understand the emergency, as shown in Fig. 2. The system was tested on Yeonhee Hall located in Yonsei University.

Fig. 3 represents the map visualization on an interactive Google Map within the web application. Upon startup, the system autonomously constructed an emergency context without requiring user input. With the user location inferred by Google's website permissions, potential nearby shelters that are relevant to the emergency context were identified. The results were rendered in real time on the user interface.

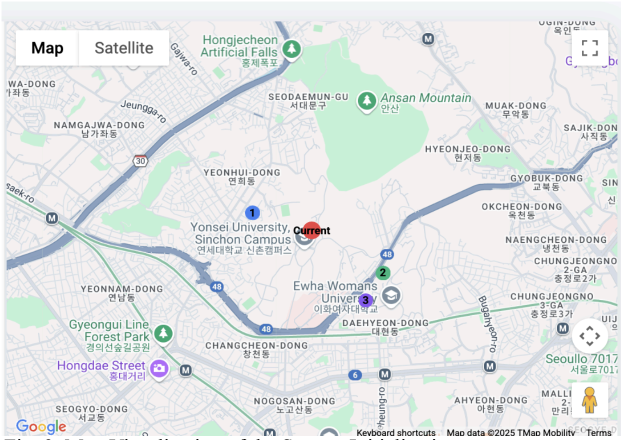


Fig. 3. Map Visualization of the System Initialization

Fig. 4 represents the ranked nearby locations to evacuate to, where the Google Places API and a local CSV file are used to validate locations.

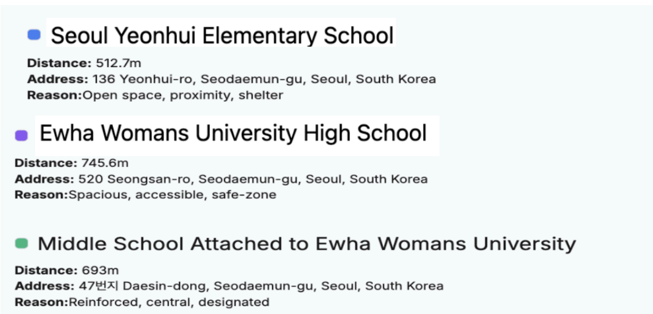


Fig. 4. List of Ranked Locations for Case 1

The system provided the distance from the user's location, the address of the location, and three keywords of reason. Each location was accompanied by metadata, including address, demonstrating the system's capability to retrieve, rank, and present multi-source data dynamically.

System messages indicated that all submodules – location detection, static data loading, and map rendering – executed successfully. This validated the seamless orchestration of modular tools within the LangChain agent pipeline, despite the complexity.

The result demonstrates the autonomous operation of the proposed system when sufficient information is available from external sources. In the absence of manual input, the system was able to build the context of an emergency, analyze user context, and recommend nearby safe locations.

For cases in which the user requires more customized assistance, the agent is also equipped with a query interface. With access to tools for retrieving and analyzing data from OSM's Overpass API and Google Maps Elevation API, the agent is also able to produce instructive visualizations through additional querying. Fig. 5 demonstrates this, where the red circle represents the user's location, the green lines represent the routes to the shelters, colored in black, and the orange circles represent steep areas with stairs that lack basic accessibility features such as ramps and elevators.

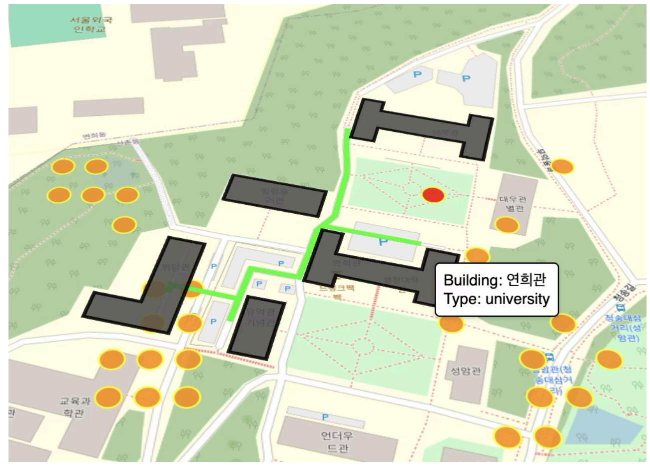


Fig. 5. Walkable Paths to Shelters Visualized Using deck.gl.

Fig. 6 illustrates the agent's tool orchestration process for an example shelter-finding scenario. When given the natural language request 'Guide me to nearby shelters with easy access,' the agent automatically plans and executes a multi-tool workflow spanning data retrieval, spatial analysis, and visualization. This demonstrates the agent's ability to execute complex geospatial workflows utilizing multiple datasets.

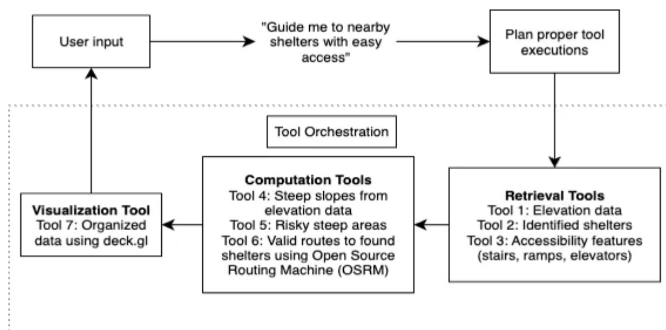


Fig. 6. Example of Agent's Elevation Visualization Workflow

C. Case 2. Insufficient Information

An additional case for an insufficient dataset was done to test the proposed system's vulnerability against a lack of information. The minimum alert threshold was set to 100, and the CSV file was depleted from the local dataset to simulate the lack of information. Fig. 7 shows the system message produced when manual mode was activated.

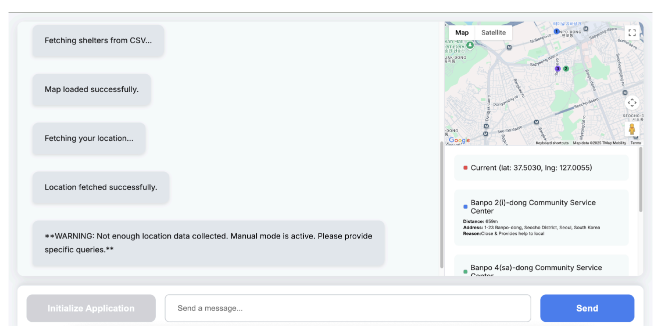


Fig. 7. Chatting Interface of Case 2

The activation allowed the user to insert queries. The following query was inserted: *"Find the nearest place to* *evacuate from the earthquake."* As a result, common nearby resources were extracted using Google Maps API with Case 1; however, non-relevant places such as a nearby bar named "Shelter" have also appeared as an output, alerting to the problem of hallucination of the proposed system when given inadequate resources.

V. Conclusion

This study was able to demonstrate a successful agent-based emergency response system that provides immediate guidance during sudden-onset emergencies. The automatic detection of crisis contexts and integration of them with geospatial resources allowed the system to effectively advocate for the evacuation of an emergency. The combination of an LLM-based agent and specialized tools proved effective in multi-step reasoning and executing complex workflows for emergent situations.

The simulated earthquake scenario in Seoul validated the system's end-to-end pipeline, from context construction to the visualization of potential shelters, with extra querying enabled for additional utility. Although challenges such as hallucinations still exist, by designing a system to fuse verified local data with real-time APIs, this study demonstrates that we can create a more reliable and distinct countermeasure than systems that rely on a single source of information.

Future studies should aim to combine more real-time datasets, such as traffic and infrastructure status, for enhanced context creation. Since the agent's current capabilities are restricted to the tools being used, the development of an agent capable of managing multiple tasks correlated to user safety, using advanced models, frameworks, and standardized protocols, is feasible.

Acknowledgment

This research was carried out within the Space Management and Design IT Lab, under the supervision of Professor Jin-Kook Lee, as part of the YISS undergraduate research internship program at Yonsei University.

This work is supported in 2025 by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant RS-2021-KA163269).

References

1. M. Sherwood, G. N. Jones, and A. M. Mechan-Losnegard, "Accelerated increase in the frequency and intensity of heatwaves, floods, and wildfires under anthropogenic climate change," Nature, vol. 650, no. 2, pp. 112–119, Apr. 2025.
2. Han, Z., Wu, G. Why do people not prepare for disasters? A national survey from China. *npj Nat. Hazards* **1**, 1 (2024).
3. R. Patel, L. Mehta, and J. Smith, "Time to survival: impact of evacuation delays on mortality rates during hurricanes and wildfires," Proc. 2024 IEEE Conf. Humanitarian Tech., pp. 45–52, Nov. 2024.
4. A. Gomez et al., "Social vulnerability and disaster resilience: impacts of income and age on evacuation outcomes during extreme weather events," Science Adv., vol. 10, no. 5, eabf1234, May 2024.
5. Hartley, Catherine A, and Elizabeth A Phelps. "Anxiety and decision-making." Biological psychiatry vol. 72,2 (2012): 113-8. doi:10.1016/j.biopsych.2011.12.027
6. Gupta et al., "Geode: A zero-shot geospatial question-answering agent with explicit reasoning and precise spatio-temporal retrieval," arXiv:2407.11014, 2024.
7. Xie et al., "A RAG-based multi-agent LLM system for natural hazard resilience and adaptation," arXiv:2504.17200, 2025.
8. Camps-Valls, G., Fernández-Torres, MÁ., Cohrs, KH. et al. "Artificial intelligence for modeling and understanding extreme weather and climate events." Nat Commun 16, 1919 (2025).
9. Sapkota, Ranjan, Konstantinos I. Roumeliotis, and Manoj Karkee. "Ai agents vs. agentic ai: A conceptual taxonomy, applications and challenges," arXiv preprint arXiv:2505.10468, 2025.
10. Qu, Changle, et al. "Tool learning with large language models: A survey," Frontiers of Computer Science 19.8, 2025.