Generative AI-Driven Spatial Data Extraction in OpenStreetMap using Natural Language

Vohra, M.A[1], Singh, T. P.[1], Illayaraja, K.[2] and Shah, S. K.[1*]
[1]Symbiosis Institute of Geoinformatics, Symbiosis International (Deemed University), Pune, India
E-mail: sahilshahwnr@gmail.com*
[2]iSpatial Techno Solutions Pvt. Ltd., Hyderabad, India
Corresponding Author

DOI: https://doi.org/10.52939/ijg.v21i6.4233

Abstract
With the rising availability and support of geospatial data and tools, geospatial data analysis is increasing rapidly. However, geospatial data is challenging to extract and understand by individuals with limited or no prior knowledge of handling such data. This study presents a novel platform that integrates generative artificial intelligence (Gen. AI) and prompt engineering techniques for geospatial data retrieval and analysis. This is achieved by firing natural language queries and integrating a generative pre-trained transformer GPT-3.5 for data retrieval and analysis. The platform translates unstructured natural language inputs into structured Overpass API queries, retrieving detailed geospatial data from OpenStreetMap (OSM). The system streamlines the process, from query to visualization, enabling users without technical geospatial expertise to access spatial information seamlessly. It supports geospatial data retrieval tasks such as Point of Interest (PoI) extraction, proximity queries, and attribute-based retrieval. The experimental results show that the proposed approach outperforms existing tools such as Google Earth Engine (GEE), GeoGPT, GeoInsight, MapQA, OSM-GPT with an average query-execution time of 17.3 seconds and an average accuracy of 95%. It shows a significant improvement in usability over manual Overpass query construction. The proposed framework achieves higher performance while maintaining a lightweight design that does not require model fine-tuning or external training data. Unlike existing tools that heavily rely on fine-tuned transformers with tightly coupled components, the proposed framework is modular, prompt-driven, and API-based, which enables its rapid deployment and minimal resource usage. This lightweight architecture helps to improve system maintainability, scalability, and makes it easily accessible for real-time applications and end-users with limited technical infrastructure. Overall, the framework offers a scalable, accessible, and extensible solution for spatial data querying in open-source GIS workflows. This study can transform conventional geospatial data analysis practices into a more inclusive and user-friendly approach that features a geointelligent environment.

Keywords: Artificial Intelligence, Geospatial Data, GIS, Generative AI, GPT, Natural Language Processing, OpenStreetMap, Prompt Engineering

1. Introduction

Geospatial analysis is the science of collecting, processing, and interpreting geographic data, which has become essential in various fields [1]. Applications that use geospatial data can be found in urban planning, environmental monitoring, disaster management, agriculture, etc. [2]. The geospatial data can be accessed and analyzed using geographic information systems (GIS) tools, integrated programming platforms, cloud computing, and natural language processing (NLP) based frameworks. The traditional geospatial data analysis tools often present significant barriers to non-expert users due to their complexity and the specialized knowledge required to effectively use them [2][3][4][5] and [6]. For example, the traditional problem of locating nearby schools using geospatial tools involves a) specification of location coordinates, b) distance range, c) conditions like proximity to gardens and transportation services, and traffic conditions. For this, the Buffer tool is used, followed by the Intersect tool, to filter the results for locating the schools. From this example, it is pertinent that users without technical GIS knowledge find it challenging to adapt to such workflows. This limits the accessibility of powerful geospatial tools to a broader audience.

ISSN: 1686-6576 (Printed) | ISSN 2673-0014 (Online) |

It also affects the ability of the users and decision makers to make informed decisions driven by spatial data analysis. Many times, officers involved in urban planning and survey fields do not have GIS technical expertise for formulating the query or defining a sequential workflow through a series of geospatial operations for geospatial data analysis. This necessitates a requirement of a user-friendly platform for accessing geospatial data without a specific prerequisite for GIS knowledge and/or commands.

Recent developments in digital mapping, web-based platforms, and artificial intelligence (AI) have opened new opportunities for geospatial data analysis. The confluence of AI and GIS has garnered attention in recent years, culminating in the emergence of GeoAI as a dedicated research frontier [4] and [6]. One of the most promising developments in this regard is integrating NLP technologies [7] and [8] with geospatial data. It enables users to interact with complex systems using natural language. ChatGPT is an AI chatbot built upon OpenAI's large language models (LLMs), which is used for the generation of ideas, content, code, and multimedia data, translation, and getting specific answers pertaining to any specific query [9][10] and [11]. LLMs can solve complex tasks with a chain-of-thought strategy [12]. This strategy involves getting the answer from LLM through a series of steps (few-shot prompting) against directly answering a given question (zero-shot prompting). It is further seen that these LLMs return their results in text-based solutions. Existing natural language interfaces for spatial data retrieval exhibit a number of structural and functional limitations that restrict their applicability in broader geospatial contexts. Systems such as NLMaps [13] and MapQA [14] rely heavily on manually curated grammars and predefined templates to parse user queries. While effective within narrow domains, such approaches lack flexibility and tend to underperform when exposed to open-ended or syntactically irregular input. More recent tools like OSM-GPT [15] attempt to leverage general-purpose language models to bridge this gap. However, these implementations fail to capture the semantic nuance of user queries. In many cases, they generate a syntactically correct Overpass query that is logically misaligned with the users' intent, particularly when dealing with ambiguous questions.

Inspired by the capability of LLMs to solve complex tasks, easily accessible geospatial data using OSM [16] and [17], and limitations of existing studies, this study presents an innovative framework for the retrieval of geospatial data using OSM. The preliminary work undertaken in this study involves:

a) Design and Development of an Integrated GPT-powered Chatbot for geospatial data analysis.
b) Utilization of prompt engineering for efficient spatial data extraction.
c) Development of an open-source, low-cost platform for comprehensive geospatial data analysis.

The proposed framework involves a web-enabled framework supporting the following stages: 1) User queries are accepted in natural language, 2) These natural language queries are converted to well-curated prompts for feeding them to GPT 3.5. 3) Structured queries returned by ChatGPT are integrated with the Overpass API of OSM to retrieve geospatial data. This helps in smoother retrieval and analysis of geospatial data, making it accessible to all individuals irrespective of their background. Figure 1 shows the methodological flow of the proposed framework. The Overpass API [18] is a web service that enables advanced querying of OSM data using a specialized query language. It supports real-time extraction of spatial features based on tags and geographic constraints, making it a key component in automating the data retrieval process in this study. The remaining sections of this article are organized as follows: Section 2 discusses related work, while the methodology of the proposed technique is discussed in section 3. Section 4 highlights the results of the findings, followed by the discussion. The proposed framework sets a new standard for accessibility and usability in geospatial data analysis, paving the way for more inclusive and efficient spatial data solutions. Section 5 concludes with a focus on the role of the proposed technique in enhancing geospatial technology, with a discussion on future avenues.

2. Related Work

The evolution of geospatial technologies over the past few decades has transformed the utilization of spatial data. This is achieved by providing tools for visualization, analysis, and decision-making [1]. The advancement in digital and information communication technologies (ICT) has enabled techniques that have made geospatial data more accessible and interactive [19] and [20]. Currently, to solve any geospatial problem, GIS tools, integrated programming frameworks, cloud computing platforms and/or Natural-language processing platforms with support from large language models (LLM) are used for accessing and analysing the geospatial data. Web-based GIS platforms like ArcGIS Online [21] and Google Earth Engine (GEE) [22] allow users to visualize and analyze spatial data using a web browser.

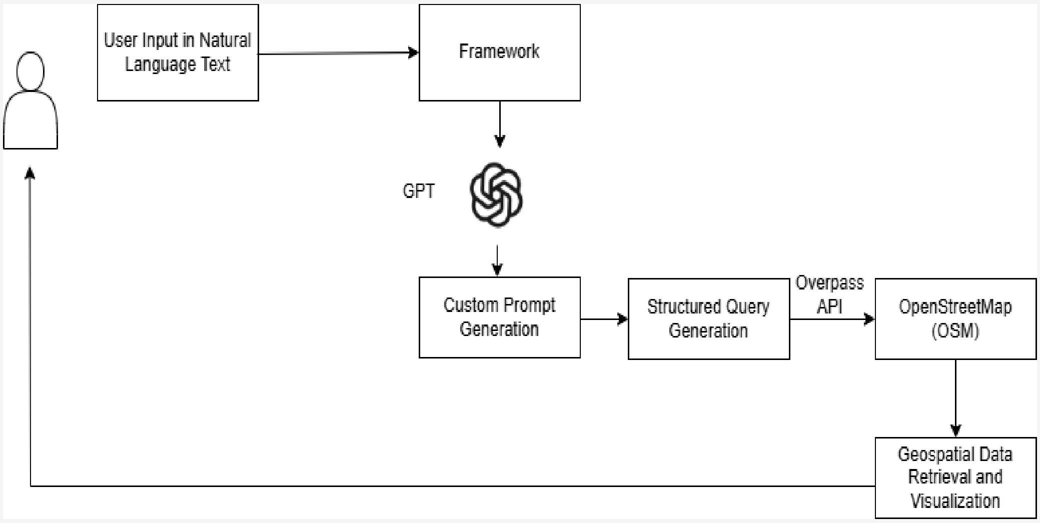

Figure 1: Flow of Proposed Framework
(User queries in natural language are transformed to structured queries using custom prompts)

Whilst ArcGIS Online supports natural language search to a limited extent, complex spatial queries still require manual configuration and scripting using ArcPy. Moreover, it is proprietary to use and access the geospatial data using this platform. It limits the capability of normal users to analyse the geospatial data. While GEE supports free cloud-based planetary data retrieval and access with ease, it requires users to be familiar with scripting and geospatial data structures. This allows users to run complex algorithms on satellite imagery and geospatial datasets through a commodity of parallel servers on the fly. Query formulation in GEE is explicit and programmatic, making it less accessible for non-specialists. Moreover, GEE is optimized for raster-based remote sensing data and is less focused on the fine-grained vector queries often supported by Overpass queries. Mapbox is a platform for building custom maps and visualizations. This is widely used in navigation applications [23].

Recent advances in NLP, such as OpenAI's GPT-3, have enabled machines to understand and generate human-like language [24]. GPT-3 has been used in the application GeoInsight to translate user queries into geospatial data requests [25] and [26]. Such systems focus on domain-specific spatial analytics that prioritize interpretation over flexible data extraction. These systems tend to encapsulate common use cases but offer limited support for arbitrary or novel spatial queries. Models such as GPT-4 [27], PaLM [28], PaLM-2 [29], and LLaMa [30] have been used in recent research contributions. These models present zero-shot or few-shot performance in tasks like semantic question answering [31] and semantic reasoning [32] and [33]. LLaMA-based models are typically deployed as general-purpose language models. Their integration into geospatial systems requires significant infrastructure for interpretation, fine-tuning, and prompt engineering. These models require larger computing resources for overall processing and analysis. LLMs were used to generate more concise descriptions of objects of interest to produce better images [34] and [35]. In the field of GIS, researchers have also investigated whether LLMs can understand and then solve geospatial tasks [36][37] and [38]. The spatial semantic reasoning ability of ChatGPT is assessed on geospatial tasks in [39]. GeoGPT [40] attempts to integrate natural language interfaces with geospatial reasoning by fine-tuning transformer models on spatial datasets. It is observed that the performance of this model is sensitive to the quality and scope of its training data, with the need for task-specific tuning. MapQA [14], an open-source question answering system for geospatial data retrieval using OSM, requires heavy computational resources for model training and fine-tuning. Further, the system shows performance drops while generalizing it with complex or multi-tag queries. OSM-GPT [15] offers a notable implementation that leverages GPT-3.5 to convert natural language queries into Overpass QL, facilitating OSM data retrieval. While it provides a user-friendly interface and demonstrates commendable performance, it fails to showcase the promising outcome in semantic query understanding. From the comprehensive literature review, it is observed that existing applications and web frameworks require domain knowledge of GIS while accessing these tools for geospatial data retrieval and analysis.

Table 1: Analysis of findings from existing works against proposed framework

Tool/ Frame-work	Interface	Average Accuracy (%)	Average Execution Time (s)	Findings and Analysis	Usage / Licensing	Reference
MapQA	Geospatial QA system using OSM, BERT and GPT-4	89	20-23	Requires fine-tuning and requires a larger computational resource	Open-source	[14]
OSM-GPT	Prompt-based, GPT-3.5	88	22-24	Sensitive to phrasing and word sequencing	Open-source but needs valid API-key	[15]
ArcGIS	GUI + Python Scripting (ArcPy)	92	60, can vary depending on script for advanced operations	Scales with enterprise resources; supports both vector and raster operations	Proprietary	[21]
GEE	Script-based (Java Script/Python)	93	Depends on script and algorithms, spatial and temporal extent	Scales well with global data; optimized for raster operations	Free tiers with options for paid subscription	[22]
GeoInsight	Dashboard-based NLP	89	19-24	Limited to predefined layers and dashboards	Proprietary	[25] and [26]
LLaMA	General-purpose LLM	84	26-28	Requires significant memory and computing Performance depends on the input task and LLaMa version	Free usage with options for paid subscription for using advanced functionality	[30]
GeoGPT	Fine-tuned LLM	91	20-24	Computationally heavy for large-scale use	Free usage with options for paid subscription for using advanced functionality	[40]
Proposed Framework	Prompt-based GPT-3.5 with query templates		16-20	Performs well on dense OSM layers	Open-source	This study

Inspired by this challenge and to make a more user-inclusive platform, irrespective of their backgrounds, this study presents a web-enabled framework that utilizes queries in natural language for accessing and analyzing the geospatial data through an interactive dashboard. Table 1 presents a comparative analysis of existing platforms against the proposed framework. It is observed that the proposed framework excels in terms of faster query response time, precise outcomes using a lightweight structure in an open-source environment against existing frameworks.

3. Materials and Methods

In this section, we introduce a web-enabled framework developed to assist non-geospatial users in retrieving geospatial data from the open-source environment. This is achieved by accessing OSM with the aid of structured queries generated using GPT. Figure 2 presents the overall architectural representation used in developing the proposed web-enabled framework. This framework offers features like selection of area of interest (AOI), user input processing, custom prompt generation, structured query generation, and processing through Overpass API, followed by geospatial data retrieval and visualization. Each of these phases is discussed below.

3.1 Web-Enabled Framework

User interaction is crucial for providing a user-intuitive experience while handling the geospatial data by non-experts. This is carefully considered while designing the UI with facilities that support easy navigation. For the development of web application, Reacts' Next.js framework [41] and [42] has been utilized.

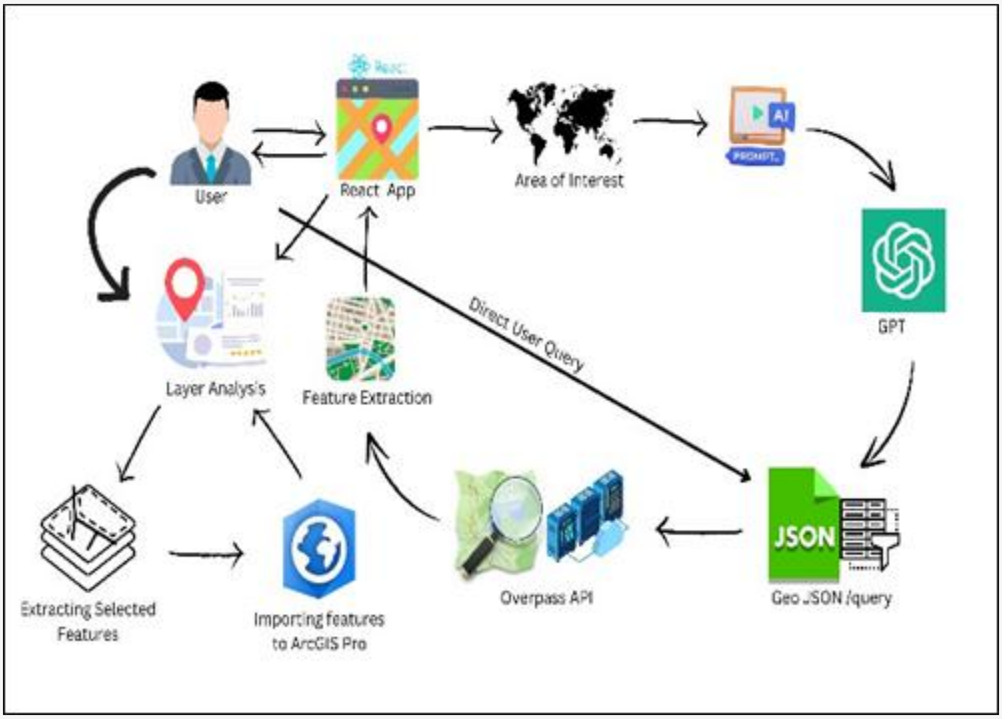

Figure 2: Functionalities of Proposed Web-enabled Framework

This is chosen considering its capabilities in server-side rendering and static site generation. This framework provides a dynamic and responsive user interface for easily accessing geospatial data. Furthermore, TypeScript and Tailwind CSS were utilized to develop a user-friendly interface. Node.js is used to provide backend services, considering its ability to handle multiple concurrent requests. This ensures frameworks' continual availability and responsiveness while handling larger geospatial data.

3.2 Components and Modules

Users can interact with the developed web framework in two ways: a) Specification of Overpass API query in a structured format directly b) Specification of query in Natural language. If the query is specified in a structured format, the framework retrieves geospatial data and presents it for visualization and export in GeoJSON format. If the user specifies the query in natural language, the user is facilitated with the selection of AOI, query specification, data retrieval, extraction of selected features, followed by visualization over the map using a sequential pipeline. This is achieved with the help of the generative AI model GPT 3.5 for generating custom prompts from user queries, which are further directed to OSM through the Overpass API for geospatial data retrieval. The natural language queries are fed to GPT 3.5 along with custom prompts, sample input queries, and expected outputs. The structured query as per the requirements of Overpass QL will be generated by GPT, which is directed to OSM, and finally retrieved geospatial data outcome will be presented to the user. The user is provided the facility to visualize, export, and manage the retrieved geospatial data. Each of the modules in the framework is discussed below:

3.2.1 Selection of AOI

Users can select an area of interest (AOI) by browsing through the interactive world map. This ensures easy data retrieval with a focus on users' interests. This further provides facilities like Zoom, PAN, and a selection of areas easily.

3.2.2 Input using natural language

With the integration of GPT services, this platform allows users to input their queries using natural language. For instance, users can direct queries such as "Show all parks" to this platform. These queries are then interpreted based on the user's selected AOI and converted to Overpass API requests in a structured format. This integration aids in the extraction of geospatial data and visualization without the requirement of any technical expertise.

ISSN: 1686-6576 (Printed) | ISSN 2673-0014 (Online) | © Geoinformatics International

3.2.3 Processing

This stage involves the generation of a structured query from natural language input, which is required for retrieving the geospatial data using an Overpass query from OSM. It uses custom prompt templates involving prompt initialization, specification of output format, followed by sample examples. These carefully curated prompts help GPT to understand the semantic context of the query before responding with the output. The generated structured query is used further to retrieve geospatial data from the open-source environment OSM. The details of each stage are discussed in the following sections.

3.2.3.1 Query generation through custom prompt

This stage involves designing specific prompts to guide the GPT model in creating precise responses. The main steps involved include understanding the user intent by referring to their past interactions and crafting the prompt required for instructing GPT 3.5. It translates user input into a structured Overpass query (Figure 3). The transformation of natural language queries into structured Overpass queries depends on carefully crafted prompts. This helps to produce syntactically valid and semantically relevant queries. The design of these custom prompts is driven by three core components: structural guidance, semantic grounding, and failure containment.

Structural Guidance:

Prompts consistently include a fixed template to ensure syntactic conformity with the Overpass query. These templates define the expected structure for the output format and query blocks.

Semantic Grounding:

In this, contextual hints that map user intent to OSM tagging conventions are embedded in the prompts. For example, if a user asks for “public transportation stations” the prompt may include a dictionary of synonymous OSM tags like "railway=station", “airport=station”, etc.

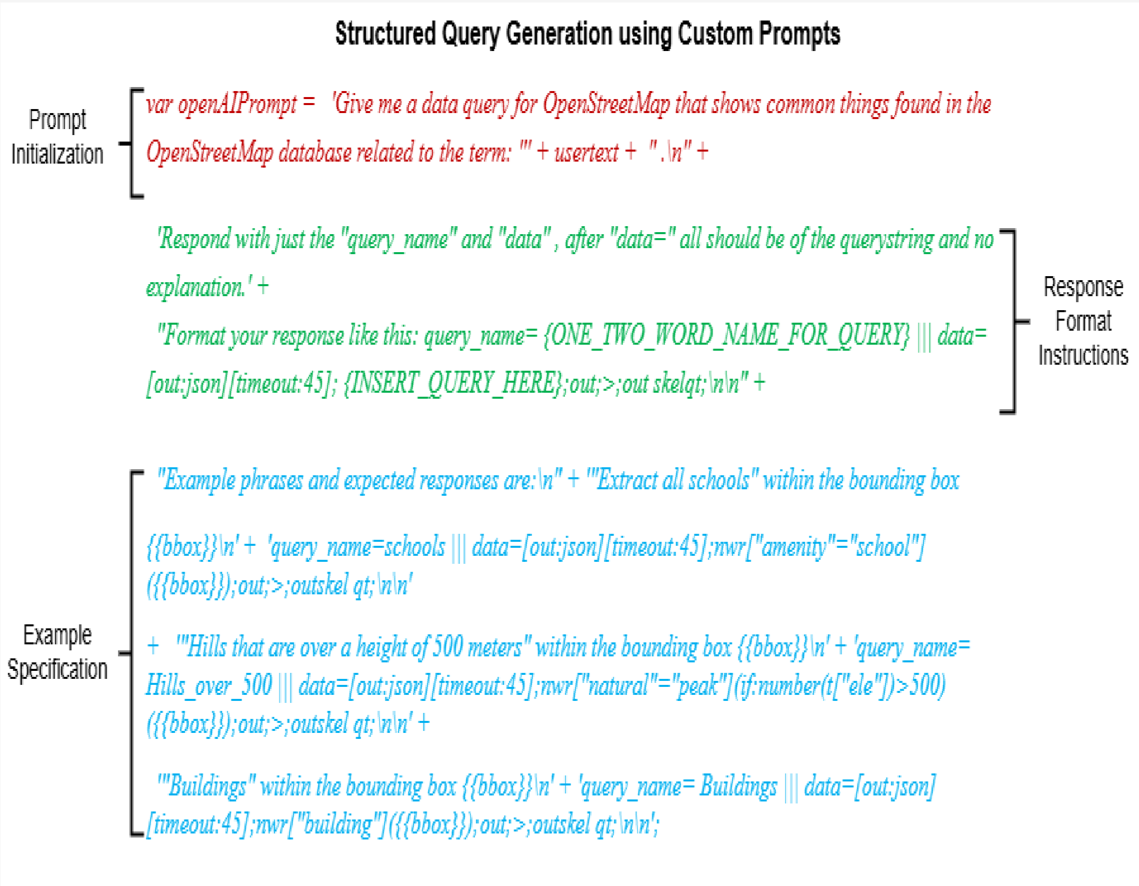

Figure 3: Workflow of structured query generation

ISSN: 1686-6576 (Printed) | ISSN 2673-0014 (Online) | © Geoinformatics International

Failure Containment & Iterative Refinement: Each of the generated query is validated. If a generated query fails validation or returns no results, the system includes fallback prompts that ask GPT to reformulate the query using wider tags. For example, queries such as "Extract all schools" or "Hills that are over a height of 500 meters" within a bounding box are provided with formatted responses to illustrate the expected output. The application captures the model's responses, extracts the relevant data query, and prepares it for execution. The responses are parsed to separate the query name and the data string, ensuring they can be directly used to query the OSM database. The components in the template of the custom prompt are as follows:

1. Prompt Initialization
Start with a request for a data query related to a specific term by passing user input text in natural language.

2. Response Format Instruction
Specify the format that includes "query_name" and "data."

3. Example queries
Provide example queries and expected responses to guide the model. When executed through GPT, these custom prompts will generate structured queries in the format required by the Overpass API.

Table 2 shows the taxonomy of various prompt categories used while generating the custom prompt templates, which are later used for the generation of the structured Overpass query.

3.2.3.2 Data retrieval
The structured query generated from GPT by processing user queries is used to retrieve geospatial data. This is carried out with the help of Overpass API. The Overpass API is a powerful query interface designed to extract structured data from the OSM database. It allows users to formulate complex queries based on tags, spatial constraints (e.g., bounding boxes, radii), and logical operators. In this study, Overpass API serves as the backend data source for all spatial queries generated by the proposed framework [18]. The steps involved in data retrieval are:

1. Fetch Request to Overpass API
The application sends a POST request to the Overpass API endpoint with dynamically generated queries. The signal parameter from overpassAbortController allows the request to be aborted if necessary. The request includes the Overpass API query specified using geographical bounds.

2. Processing the API Response
The output format is converted to JSON for easier processing. This contains the requested geospatial data.

3.2.3.3 Handling and visualizing data
Features, if available, are added to the map as new layers with unique identifiers and colors. These features include points, lines, and polygons, which are added and visualized on an interactive map as distinct layers. These layers are customizable, which ensures better usability. Multiple features can be extracted simultaneously through this platform. This helps in advanced geospatial analysis while handling complex geospatial data.

3.2.4 Data visualization
The data retrieved from the Overpass API into GeoJSON format and displays it as interactive layers on the map using the GeoJSON layer component. Each layer represents specific geographical features like roads, buildings, or parks. The application facilitates handling multiple data layers simultaneously. Layers are dynamically added based on the user queries. Each layer is displayed hierarchically to avoid obscuring important features. Users can modify the appearance of each map layer in terms of color, opacity, and order.

Table 2: Taxonomy of prompt categories

Prompt Category	Usage	Example
Direct Conversion	Conversion to Overpass Query	Translate this query: ‘bike racks near train stations in Mumbai’
Contextual Rewriting	Clarify vague intent before translation	Rewrite and then convert: ‘places for eating near garden’
Spatial Constraints	Enforce geolocation-specific results	Use bounding box for: ‘schools in Pune’
Multi-entity Queries	Extract multiple categories in one prompt	Find parks with sports facilities in Delhi using Overpass QL”

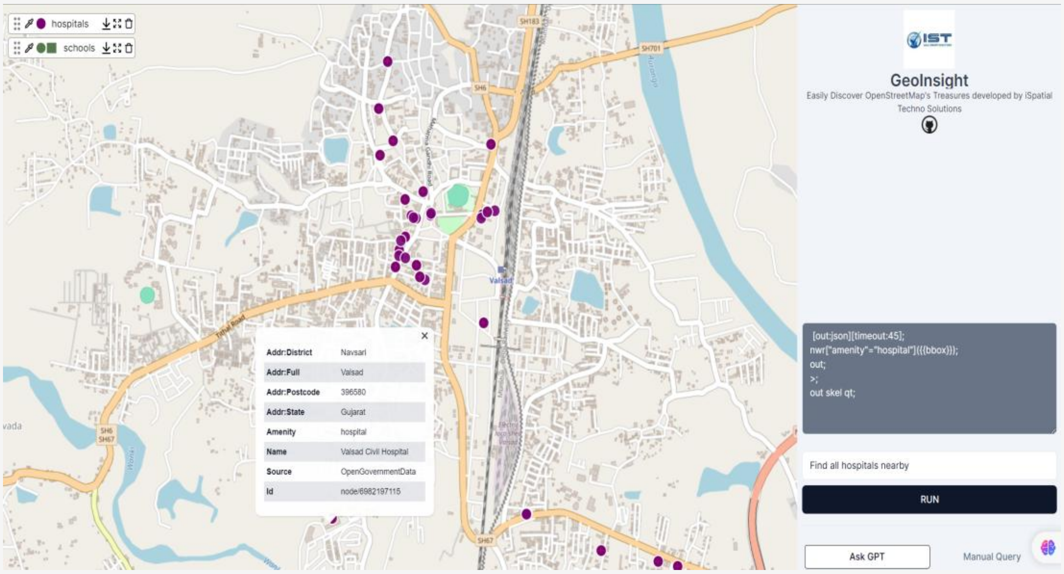

Figure 4: Outcome generated by proposed application

This allows for a personalized and comprehensible map view. Toggling layer visibility and zooming to specific layers are also facilitated through this application. These customization features ensure that users can tailor the map presentation to their specific needs.

4. Results and Discussion

The retrieval and analysis of geospatial data are facilitated through the web-enabled framework through functions that support the specification of input queries using natural language. Users can specify their queries in plain English, which are converted to precise, structured queries using tailored custom prompts passed to GPT, which are then passed to OSM for geospatial data retrieval. The following example shows a sample user input, its corresponding query generated through custom prompts of GPT, followed by the results returned by our application (Figure 4).

User Specified Query in Natural Language:

"Find all nearby hospitals"

Structured Query generated through GPT:

{
"query_name":"hospitals","osmquery":
[out:json][timeout:45];nwr['amenity'='hospital']({{bbox}});
out;>;out skel qt;"
}

Results obtained through OSM:

The results show the points indicating hospitals near the user's current location. These points are automatically added as features on the interactive map. The proportion of points that are precisely returned is comparable to available commercial solutions. The user can further customize these depending on the requirements. The application allows users to select AOI by drawing a bounding box around the desired area while formulating the query. This ensures that the application returns only the corresponding geospatial data from the selected area.

For example, for a user query "Show me the parks within the bounding box", the application returns all the parks within the specified area. This can help urban planners who may not have geospatial knowledge to seamlessly retrieve the data for green space assessments. This application provides the facility for interacting with geospatial data and downloading it using GeoJSON format. The downloaded data can be easily imported into GIS software. The obtained results are precise, demonstrating an average performance of 95% accurate geospatial data retrieval. In applications such as urban density analysis and transportation, this framework can be utilized directly for faster and more precise geospatial data extraction. Users can export the building data and perform urban density analysis using our application. This is helpful for city planners in making informed decisions related to future developments. Figure 5 shows the results obtained for extracting specific features like college buildings to analyze nearby buildings/other features. This can aid the respective government officials to analyze this data more suitably, and it helps in planning for further infrastructure.

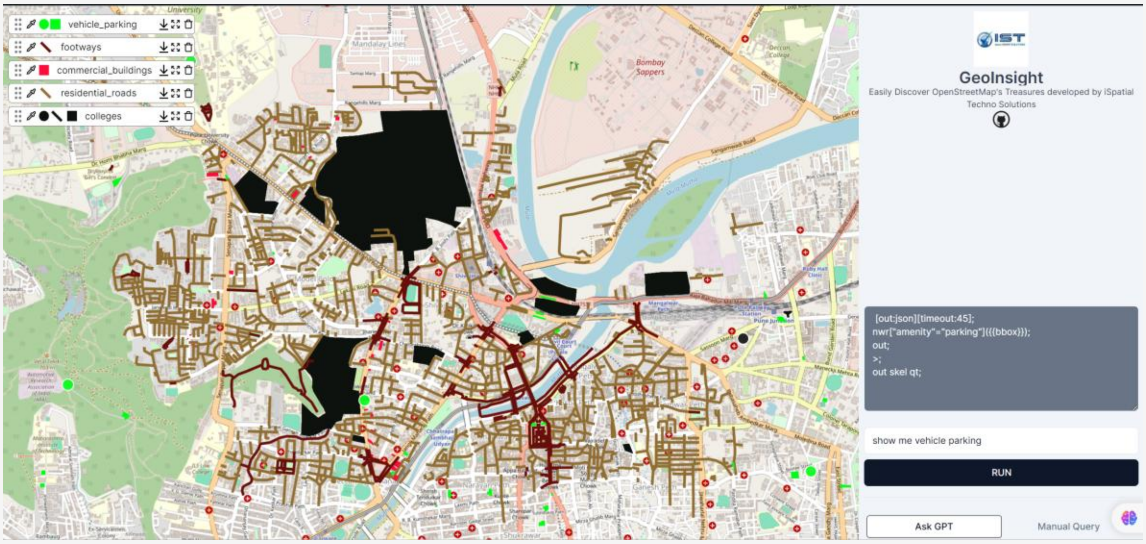

Figure 5: Results generated for extraction of specific feature

Analysis can also help to know the status of various features like analysis of parking spaces and total commercial area near the college. On the other hand, planners involved in managing transportation networks can use this application to learn the present status of bus stops and bus routes.

Model Validation:

All the experimentation was conducted on a terminal with an i5 processor having 16 GB of RAM without graphics support on the Windows 11 operating system. Design and development of web-enabled dashboard and its integration with GPT 3.5, custom prompt design, etc., was undertaken using this terminal. A freely available API key for accessing GPT 3.5 and Overpass API was procured through respective portals. The proposed framework and the script used for integration do not require model fine-tuning or support from external training data, making it lightweight and faster.

To support the reported average accuracy of 95%, a structured evaluation was conducted on a test suite of 50 user queries covering diverse spatial data types and query patterns. Each query was evaluated on two criteria: a) Syntactic validity, which assesses whether the generated Overpass QL executed successfully. b) Semantic correctness for assessing if the output matched the user's intended feature class and spatial scope. Manual verification and sample-based spatial overlays were used to validate the outputs. The overall average accuracy was 95%. However, a significant variance was observed across query categories, as shown in Table 3. The framework was tested on a wide range of query categories, and the system outcome was assessed by human evaluators. The accuracy reported is the average accuracy results obtained on multiple queries of the same category. It was calculated as the ratio of correct responses against the total number of responses returned by the framework. The correctness of the response was assessed by geospatial experts. The responses and failure modes pertaining to each of the query categories were also assessed.

Further to validate our findings, a task-specific comparison of the proposed framework against existing works was undertaken, which focused on spatial data extraction pertaining to three common tasks in geospatial data analysis:

1. PoI Extraction (e.g., "Find all hospitals in South Delhi")
2. Buffer/Proximity Queries (e.g., "Restaurants within 1 km of IIT Bombay")
3. Complex/Multi-tag Queries (e.g., "Schools with playgrounds in Pune")

It is observed that the proposed framework outperforms in terms of average accuracy of the retrieved geospatial data in raster and vector formats across PoI extraction and buffer queries, while it shows some lag and vagueness in the outputs produced for complex queries, as shown in Table 4.

Ethical Considerations and Model Bias:

The response generated by LLMs are directly dependent on their training data, including assumptions about geographic context (e.g., overrepresentation of urban contexts). These biases may result in incorrect or skewed Overpass query outputs. To address this, we designed carefully curated prompt templates that encourage explicit geographic referencing and discourage implicit assumptions.

Table 3: Result analysis of proposed framework across various query categories

Query Category	Sample Queries	Average Accuracy (%)	End-End Execution Time (s)	Failure Mode(s)
Point of Interest (PoI) Extraction	"Hospitals in Mumbai"	98	13.6	Minor tag mismatches
Multi-tag Features	"Tourist spots with cafes"	89	23.9	Misordered logic; missed spatial filters
Vague Queries	"Public spaces around"	94	19	Ambiguous tags; overly broad Overpass filters
Proximity-based Spatial Queries	"Bus stops within 250m of railway stations"	97	14.2	Incorrect radius filter syntax or bounding box failure
Land Use & Infrastructure	"Residential zones in Delhi-NCR"	95	15.2	OSM tag gaps; underspecified labels
Contextual Named Entities	"Near Gateway of India'	94	20.4	Named entity resolution ambiguity

Table 4: Task-specific analysis of proposed framework vs. existing frameworks

Framework	Task Type and Average Accuracy			Key Observations
	PoI Extraction	Buffer-based Queries	Complex Queries	
OSM-GPT [15]	90	83	86	Fixed prompt templates
ArcGIS [21]	92	90	91	1.Requires manual layer selection 2.Scripting using Python
GEE [22]	90	87	89	1.Requires JavaScript and vector data conversion 2. No native support for buffering
GeoGPT [40]	88	81	85	In some cases, buffer spatial extents are misunderstood, and incorrect outputs are produced
Proposed Framework	98	97	89	Structured prompt pipeline with custom and categorized prompts

The prompt validation checks are performed through manual interventions and validation to ensure spatial relevance. The proposed framework follows the OpenStreetMap Foundation's Open Database License (ODbL) by attributing all data sources and clearly indicating the source of extracted features wherever applicable.

Discussion and Analysis of Results:

The present study attempts to provide a highly effective and interactive application for accessing and analyzing geospatial data. The advanced capabilities of GPT are integrated into a web-enabled framework. Users can specify their queries in natural language using customized prompts. The integration of generative AI for geospatial data extraction via the Overpass API represents a significant shift in the field of geospatial technology. By transforming natural language queries into structured prompts, this technique improves both accessibility and operational efficiency. Existing techniques that use generative AI for the extraction of geospatial data [43][44][45] and [46] need high computational resources and a larger response time. It requires more effort and person-hours, depending on the complexity of the task. The proposed technique outperforms existing works with improved speed in a simpler and more accessible framework (Figure 6). Through experimental results, it is observed that the proposed technique outperforms conventional systems with an average accuracy of 95% in precise geospatial data extraction against various geospatial tasks. Further, the average execution time required for end-to-end response by the proposed system is significantly lower in comparison with existing works (Table 1, and Table 3). The platform's integration of customizable map layers allows for more dynamic and detailed visualizations, facilitating a better understanding of spatial data. This flexibility is essential for applications in fields like environmental monitoring, urban development, and logistics, where varying layers of data need to be analyzed concurrently. It strengthens the geospatial technology by significantly reducing the time required to retrieve data, maintaining high accuracy, and simplifying user interaction.

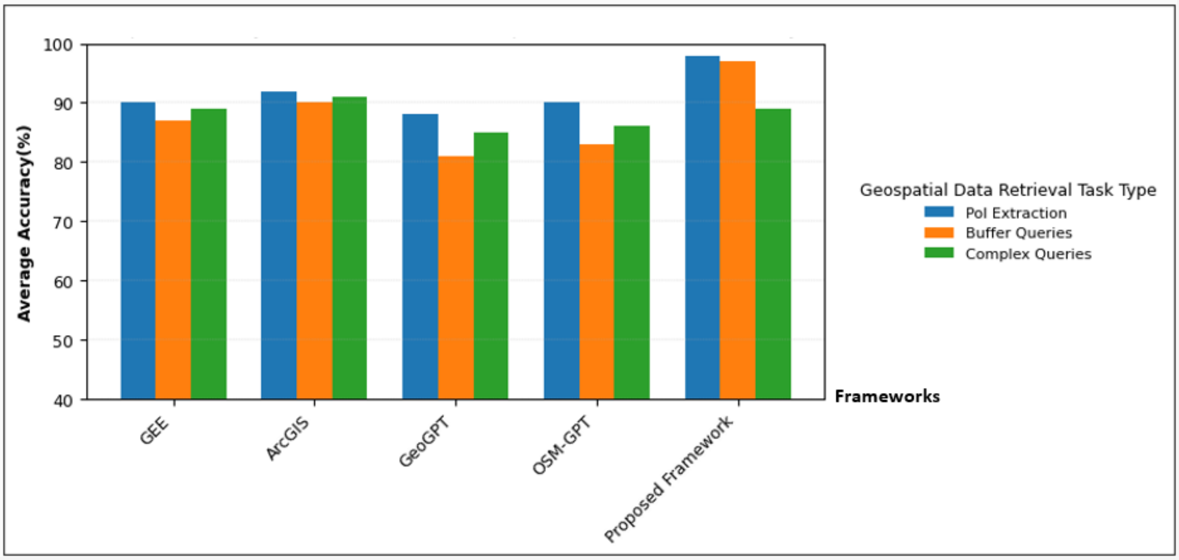

Figure 6: Comparative analysis of frameworks/tools for geospatial data extraction/analysis

Although the proposed framework demonstrates good performance, it has some limitations. The model struggles with ambiguous queries, such as “show important areas nearby," where it lacks cues to translate the intent into precise spatial filters. The complex queries that involve multiple constraints or logical operations (e.g., "parks near hospitals with parking") are not always precisely converted to structured queries for the Overpass API. This highlights a need for enhanced query parsing or multi-step prompt strategies in the future. Additionally, the reliance on GPT-3.5 introduces dependency on external APIs that limit local or offline deployment scenarios. The framework's performance in dense urban environments shows some bottlenecks, such as slower response time while dealing with the larger number of features and/or larger areas. To overcome these limitations, techniques like data chunking, where large queries are divided into smaller bounding boxes, and parallel execution of sub-queries can be explored. In addition, a result caching layer for frequently queried locations can be introduced to reduce repetitive API calls to OSM and improve responsiveness.

5. Conclusion

This study presents an interactive web-enabled framework for translating user queries in natural language to structured Overpass queries and retrieving geospatial data through OSM. It presents a comprehensive data pipeline that leverages GPT 3.5 for user query translation and geospatial data access with the aid of custom prompts. It bridges the usability gap between non-technical users and complex spatial databases by automating query generation and execution. Our evaluation shows that the system delivers competitive accuracy and faster response times when compared to contemporary GIS workflows and NLP baseline models. This affirms the suitability of this approach in real-time applications like disaster management and urban planning. Including customizable map layers further strengthens the platform's functionality, offering users dynamic and intuitive ways to visualize spatial data. Notably, the approach enhances usability without sacrificing spatial correctness, making it suitable for exploratory tasks and lightweight spatial analysis. However, limitations exist in processing ambiguous or highly compound queries, which sometimes lead to misinterpretation or incomplete data extraction. We also observed minor scalability constraints when handling large spatial extents in dense urban areas. Future work will focus on integrating multi-turn query refinement, prompt chaining techniques, and improved UI responsiveness via data chunking and caching mechanisms. In addition, a result caching layer for frequently queried locations can be introduced to reduce repetitive API calls to OSM and improve responsiveness. We also plan to expand the system’s multilingual capabilities and test it across broader geographies and query types. This work lays the foundation for more intuitive spatial data access through large language models within open-source ecosystems.

Acknowledgments
Authors are grateful to Symbiosis Institute of Geoinformatics, Symbiosis International (Deemed University), Pune, India, for providing the necessary resources and a conducive environment for undertaking this research work.

References

[1] Chang, K. T., (2008). *Introduction to Geographic Information Systems*, Vol. 4. Boston: McGraw-Hill.
[2] Maier, F. and Weinberger, M., (2024). Metaverse Meets Smart Cities Applications, Benefits, and Challenges. *Future Internet*, Vol. 16(4). https://doi.org/10.3390/fi16040126.
[3] Mooney, P., Cui, W., Guan, B. and Juhász, L., (2023). Towards Understanding the Geospatial Skills of Chatgpt: Taking a Geographic Information Systems (GIS) Exam. *GeoAI '23: Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, 85–94. https://doi.org/10.1145/3615886.36277.
[4] Tang, Z. and Kejriwal, M., (2023). Evaluating Deep Generative Models on Cognitive Tasks: A Case Study. *Discover Artificial Intelligence*, Vol. 3(1). https://doi.org/10.1007/s44163-023-00067-3.
[5] Tang, Z. and Kejriwal, M., (2023). Can Language Representation Models Think in Bets?. *Royal Society Open Science*, Vol. 10(3). https://doi.org/10.1098/rsos.221585.
[6] Janowicz, K., Gao, S., McKenzie, G., Hu, Y. and Bhaduri, B., (2020). GeoAI: Spatially Explicit Artificial Intelligence Techniques for Geographic Knowledge Discovery and Beyond. *International Journal of Geographical Information Science*, Vol. 34(4), 625-636. https://doi.org/10.1080/13658816.2019.1684500.
[7] Fuller, A., Fan, Z., Day, C. and Barlow, C., (2020). Digital Twin: Enabling Technologies, Challenges, and Open Research. *IEEE Access*, Vol. 8, 108952–108971. https://doi.org/10.1109/ACCESS.2020.2998358.
[8] Mansourian, A. and Oucheikh, R., (2024). Chatgeoai: Enabling Geospatial Analysis for Public through Natural Language, with Large Language Models. *ISPRS International Journal of Geo-Information*, Vol. 13(10). https://doi.org/10.3390/ijgi13100348.
[9] KKasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., Weller, J., Kuhn, J. and Kasneci, G., (2023). Chatgpt For Good? On Opportunities and Challenges of Large Language Models for Education. *Learning and Individual Differences*, Vol. 103. https://doi.org/10.35542/osf.io/5er8f.
[10] Pierdicca, R. and Paolanti, M., (2022). GeoAI: A Review of Artificial Intelligence Approaches for the Interpretation of Complex Geomatics Data. *Geoscientific Instrumentation, Methods and Data Systems*, Vol. 11(1), 195–218. https://doi.org/10.5194/gi-11-195-2022.
[11] Jiang, Y. and Yang, C., (2024). Is ChatGPT a Good Geospatial Data Analyst? Exploring The Integration of Natural Language into Structured Query Language Within a Spatial Database. *ISPRS International Journal of Geo-Information*, Vol. 13(1). https://doi.org/10.3390/ijgi13010026.
[12] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, Ed H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J. and Fedus, W., (2022). Emergent Abilities of Large Language Models. arXiv preprint arXiv:2206.07682. https://doi.org/10.48550/arXiv.2206.07682.
[13] Lawrence, C. and Riezler, S., (2016). Nlmaps: A Natural Language Interface to Query OpenStreetMap. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. 6-10.
[14] Li, Z., Grossman, M., Kulkarni, M., Chen, M. and Chiang, Y. Y., (2025). MapQA: Open-domain Geospatial Question Answering on Map Data. *arXiv preprint arXiv:2503.07871.* https://doi.org/10.48550/arXiv.2503.07871.
[15] Gautam, R., (2023*). OSM-GPT: An Innovative Project Combining GPT-3 and the Overpass API to Facilitate Easy Feature Discovery on OpenStreetMa*p. Available https://github.com/rowheat02/osm-gpt [Accessed May 25, 2025].
[16] Haklay, M. and Weber, P., (2008). OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, Vol. 7(4), 12–18. https://doi.org/10.1109/MPRV.2008.80.
[17] Kaur, J., Singh, J., Sehra, S. S. and Rai, H. S., (2017). Systematic Literature Review of Data Quality within OpenStreetMap. *2017 International Conference on Next Generation Computing and Information Systems*

(ICNGCIS), 177–182. https://doi.org/10.1109/ICNGCIS.2017.35.

[18] Ehrig-Page, J. C., (2020). Evaluating Methods for Downloading OpenStreetMap Data. *Cartographic Perspectives*, Vol. 1(95), 42-49. https://doi.org/10.14714/CP95.1633.

[19] Murayama, Y., (2012). *Progress in Geospatial Analysis (Ed.)*. Springer Science & Business Media.

[20] Sui, D. Z., (2009). *A Review of "Geospatial Analysis: A Comprehensive Guide to Principles, Techniques and Software Tools" Michael J. de Smith, Michael F. Goodchild, and Paul A. Longley.* Leicester, UK: Troubador Publishing, 2007.

[21] West, H. and Horswell, M., (2018). GIS has Changed! Exploring the Potential of ArcGIS Online. *Teaching Geography*, Vol. 43(1), 22-24. https://www.jstor.org/stable/26455213.

[22] Tamiminia, H., Salehi, B., Mahdianpari, M., Quackenbush, L., Adeli, S. and Brisco, B., (2020). Google Earth Engine for Geo-Big Data Applications: A Meta-Analysis and Systematic Review. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 164, 152-170. https://doi.org/10.1016/j.isprsjprs.2020.04.001.

[23] Rzeszewski, M., (2023). MapBox. *Evaluating Participatory Mapping Software*. Cham: Springer International Publishing. 21-40.

[24] Floridi, L. and Chiriatti, M., (2020). GPT-3: Its Nature, Scope, Limits, and Consequences. Minds and Machines, Vol. 30, 681-694. https://doi.org/10.1007/s11023-020-09548-1.

[25] Tao, R. and Xu, J., (2023). Mapping with ChatGPT. *ISPRS International Journal of Geo-Information*, Vol.12(7). https://doi.org/10.3390/ijgi12070284.

[26] Yang, J., Jang, H. and Yu, K., (2023). Geographic Knowledge Base Question Answering over OpenStreetMap. *ISPRS International Journal of Geo-Information*, Vol. 13(1). https://doi.org/10.3390/ijgi13010010.

[27] Achiam, O. J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., Bernadett-Shapiro, G., Berner, C., Bogdonoff, L., Boiko, O., Boyd, M., Brakman, A., Brockman, G., Brooks, T., Brundage, M., Button, K., Cai, T., Campbell, R., Cann, A., Carey, B., Carlson, C., Carmichael, R., Chan, B., Chang, C., Chantzis, F., Chen, D., Chen, S., Chen, R., Chen, J., Chen, M., Chess, B., Cho, C., Chu, C., Chung, H. W., Cummings, D., Currier, J., Dai, Y., Decareaux, C., Degry, T., Deutsch, N., Deville, D., Dhar, A., Dohan, D., Dowling, S., Dunning, S., Ecoffet, A., Eleti, A., Eloundou, T., Farhi, D., Fedus, L., Felix, N., Fishman, S.P., Forte, J., Fulford, I., Gao, L., Georges, E., Gibson, C., Goel, V., Gogineni, T., Goh, G., Gontijo-Lopes, R., Gordon, J., Grafstein, M., Gray, S., Greene, R., Gross, J., Gu, S.S., Guo, Y., Hallacy, C., Han, J., Harris, J., He, Y., Heaton, M., Heidecke, J., Hesse, C., Hickey, A., Hickey, W., Hoeschele, P., Houghton, B., Hsu, K., Hu, S., Hu, X., Huizinga, J., Jain, S., Jain, S., Jang, J., Jiang, A., Jiang, R., Jin, H., Jin, D., Jomoto, S., Jonn, B., Jun, H., Kaftan, T., Kaiser, L., Kamali, A., Kanitscheider, I., Keskar, N. S., Khan, T., Kilpatrick, L., Kim, J. W., Kim, C., Kim, Y., Kirchner, H., Kiros, J. R., Knight, M., Kokotajlo, D., Kondraciuk, L., Kondrich, A., Konstantinidis, A., Kosic, K., Krueger, G., Kuo, V., Lampe, M., Lan, I., Lee, T., Leike, J., Leung, J., Levy, D., Li, C., Lim, R., Lin, M., Lin, S., Litwin, M., Lopez, T., Lowe, R., Lue, P., Makanju, A., Malfacini, K., Manning, S., Markov, T., Markovski, Y., Martin, B., Mayer, K., Mayne, A., McGrew, B., McKinney, S.M., McLeavey, C., McMillan, P., McNeil, J., Medina, D., Mehta, A., Menick, J., Metz, L., Mishchenko, A., Mishkin, P., Monaco, V., Morikawa, E., Mossing, D. P., Mu, T., Murati, M., Murk, O., M'ely, D., Nair, A., Nakano, R., Nayak, R., Neelakantan, A., Ngo, R., Noh, H., Long, O., O'Keefe, C., Pachocki, J. W., Paino, A., Palermo, J., Pantuliano, A., Parascandolo, G., Parish, J., Parparita, E., Passos, A., Pavlov, M., Peng, A., Perelman, A., Peres, F.D., Petrov, M., Pinto, H.P., Pokorny, M., Pokrass, M., Pong, V. H., Powell, T., Power, A., Power, B., Proehl, E., Puri, R., Radford, A., Rae, J. W., Ramesh, A., Raymond, C., Real, F., Rimbach, K., Ross, C., Rotsted, B., Roussez, H., Ryder, N., Saltarelli, M. D., Sanders, T., Santurkar, S., Sastry, G., Schmidt, H., Schnurr, D., Schulman, J., Selsam, D., Sheppard, K., Sherbakov, T., Shieh, J., Shoker, S., Shyam, P., Sidor, S., Sigler, E., Simens, M., Sitkin, J., Slama, K., Sohl, I., Sokolowsky, B., Song, Y., Staudacher, N., Such, F. P., Summers, N., Sutskever, I., Tang, J., Tezak, N. A., Thompson, M., Tillet, P., Tootoonchian, A., Tseng, E., Tuggle, P., Turley, N., Tworek, J., Uribe, J. F., Vallone, A., Vijayvergiya, A., Voss, C., Wainwright, C. L., Wang, J. J., Wang, A., Wang, B., Ward, J., Wei, J., Weinmann, C., Welihinda, A., Welinder, P., Weng, J., Weng, L., Wiethoff, M., Willner, D., Winter, C., Wolrich, S., Wong, H., Workman, L., Wu, S., Wu, J., Wu, M., Xiao, K., Xu, T.,

Yoo, S., Yu, K., Yuan, Q., Zaremba, W., Zellers, R., Zhang, C., Zhang, M., Zhao, S., Zheng, T., Zhuang, J., Zhuk, W. and Zoph, B., (2023). Gpt-4 Technical Report. arXiv preprint arXiv:2303.08774. https://doi.org/10.48550/arXiv.2303.08774.

[28] Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N. M., Prabhakaran, V., Reif, E., Du, N., Hutchinson, B., Pope, R., Bradbury, J., Austin, J., Isard, M., Gur-Ari, G., Yin, P., Duke, T., Levskaya, A., Ghemawat, S., Dev, S., Michalewski, H., García, X., Misra, V., Robinson, K., Fedus, L., Zhou, D., Ippolito, D., Luan, D., Lim, H., Zoph, B., Spiridonov, A., Sepassi, R., Dohan, D., Agrawal, S., Omernick, M., Dai, A.M., Pillai, T. S., Pellat, M., Lewkowycz, A., Moreira, E., Child, R., Polozov, O., Lee, K., Zhou, Z., Wang, X., Saeta, B., Díaz, M., Firat, O., Catasta, M., Wei, J., Meier-Hellstern, K.S., Eck, D., Dean, J., Petrov, S. and Fiedel, N., (2023). Palm: Scaling Language Modeling with Pathways. *Journal of Machine Learning Research*, Vol. 24(240), 1-113. https://doi.org/10.48550/arXiv.2204.02311.

[29] Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J., Shafey, L. E., Huang, Y., Meier-Hellstern, K. S., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J. A., Bradbury, J., Brahma, S., Brooks, K. M., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crépy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., D'iaz, M.C., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., García, X., Gehrmann, S., González, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A. R., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W. H., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J.Y., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, O., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A., Roy, A., Saeta, B., Samuel, R., Shelby, R. M., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L. W., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S. and Wu, Y., (2023). Palm 2 Technical Report. arXiv preprint arXiv:2305.10403. https://doi.org/10.48550/arXiv.2305.10403.

[30] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E. and Lample, G., (2023). Llama: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*. https://doi.org/10.48550/arXiv.2302.13971

[31] Fergus, S., Botha, M. and Ostovar, M., (2023). Evaluating Academic Answers Generated using ChatGPT. *Journal of Chemical Education*, Vol. 100(4), 1672-1675. https://doi.org/10.1021/acs.jchemed.3c00087.

[32] Tan, Y., Min, D., Li, Y., Li, W., Chen, Y. and Qi, G., (2023). Evaluation of ChatGPT as a Question Answering System for Answering Complex Questions. *arXiv preprint arXiv:2303.07992*. https://doi.org/10.48550/arXiv.2303.07992.

[33] Elhafsi, A., Sinha, R., Agia, C., Schmerling, E., Nesnas, I. A. and Pavone, M., (2023). Semantic Anomaly Detection with Large Language Models. *Autonomous Robots,* Vol. 47(8), 1035-1055. https://doi.org/10.48550/arXiv.2305.11307.

[34] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M. and Sutskever, I., (2021). Zero-shot Text-to-image Generation. *International Conference on Machine Learning,* 8821-8831. https://doi.org/10.48550/arXiv.2102.12092.

[35] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. and Chen, M., (2022). Hierarchical Text-conditional Image Generation with Clip Latents. arXiv preprint arXiv:2204.06125, Vol. 1(2). https://doi.org/10.48550/arXiv.2204.06125.

[36] Hu, Y., Mai, G., Cundy, C., Choi, K., Lao, N., Liu, W., Lakhanpal, G., Ryan Zhenqi Zhou, R. Z. and Joseph, K. and Joseph, K., (2023). Geo-knowledge-guided GPT Models Improve the Extraction of Location Descriptions from Disaster-related Social Media Messages. *International Journal of Geographical Information Science*, Vol. 37(11), 2289-2318. https://doi.org/10.1080/13658816.2023.2266495.

[37] Manvi, R., Khanna, S., Mai, G., Burke, M., Lobell, D. and Ermon, S., (2023). Geollm:

Extracting Geospatial Knowledge from Large Language Models. *arXiv preprint arXiv:2310.06213*. https://doi.org/10.48550/arXiv.2310.06213.

[38] Roberts, J., Lüddecke, T., Das, S., Han, K. and Albanie, S., (2023). GPT4GEO: How a Language Model Sees the World's Geography. *arXiv preprint arXiv:2306.00020.* https://doi.org/10.48550/arXiv.2306.00020.

[39] Mai, G., Huang, W., Sun, J., Song, S., Mishra, D., Liu, N., Gao, S., Liu, T., Cong, G., Hu, Y., Cundy, C., Li, Z., Zhu, R. and Lao, N., (2024). On the Opportunities and Challenges of Foundation Models for GeoAI (Vision Paper). *ACM Transactions on Spatial Algorithms and Systems,* Vol. 10(2), 1-46. https://doi.org/10.1145/3653070.

[40] Zhang, Y., Wei, C., He, Z. and Yu, W., (2024). GeoGPT: An Assistant for Understanding and Processing Geospatial Tasks. *International Journal of Applied Earth Observation and Geoinformation*, Vol. 131. https://doi.org/10.1016/j.jag.2024.103976.

[41] Wainer, G. and St-Aubin, B., (2024). A Web-Based Architecture to Operationalize Geospatial Simulation Environments. *Advanced Theory and Simulations*, Vol. 7(9). https://doi.org/10.1002/adts.202400144.

[42] Jartarghar, H. A., Salanke, G. R., AR, A. K., Sharvani, G. S. and Dalali, S., (2022). React Apps with Server-Side Rendering: Next.js. *Journal of Telecommunication, Electronic and Computer Engineering* (JTEC), Vol. 14(4), 25–29. https://doi.org/10.54554/jtec.2022.14.04.005.

[43] Li, Z. and Ning, H., (2023). Autonomous GIS: The Next-Generation AI-Powered GIS. *International Journal of Digital Earth*, Vol. 16(2), 4668-4686. https://doi.org/10.48550/arXiv.2305.06453.

[44] Balsebre, P., Huang, W., Cong, G. and Li, Y., (2024). City Foundation Models for Learning General Purpose Representations from OpenStreetMap. Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 87-97. https://doi.org/10.48550/arXiv.2310.00583.

[45] Ferrari, E., Striewski, F., Tiefenbacher, F., Bereuter, P., Oesch, D. and Di Donato, P., (2024). Search Engine for Open Geospatial Consortium Web Services: Improving Discoverability through Natural Language Processing-Based Processing and Ranking. ISPRS International Journal of Geo-Information, Vol. 13(4). https://doi.org/10.3390/ijgi13040128.

[46] Huang, Z., Zhao, Y., Chen, W., Gao, S., Yu, K., Xu, W. and Xu, M., (2019). A Natural-Language-Based Visual Query Approach of Uncertain Human Trajectories. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 26(1), 1256–1266. https://doi.org/10.48550/arXiv.1908.00277.