

AI.MAP-Research Assist

Interactive Geospatial Mapping Platform for Researchers

Leandro Meneguelli Biondo - NOV.2024

Objective

Develop a platform that allows researchers from any field to:

1. Define their geospatial mapping needs through a conversational AI interface.
2. Identify relevant open datasets (e.g., from NASA, USGS, OpenStreetMap, or government portals).
3. Visualize findings with interactive maps that include layers, statistics, and temporal data comparisons.

Table of Contents

- AI.MAP-Research Assist
 - Interactive Geospatial Mapping Platform for Researchers
 - Objective
 - Key Components
 - Step-by-Step Guide
 - 1. Build the Foundation
 - 2. Develop AI for User Interaction
 - 3. Integrate Open Data APIs
 - 4. Implement Interactive Mapping
 - 5. Add Geospatial Analytics
 - Example Use Case
 - Learning Outcomes
 - System Workflow
 - Implementation Steps
 - 1. Local Model for Prompt Generation
 - 2. Handle API Response
 - 3. Build Feedback Loop
 - Advantages
 - Tech Stack
 - Future Enhancements
 - Funding
 - 1. Academic and Research Grants
 - 2. Climate and Environmental Grants
 - 3. Corporate Sponsorships
 - 4. Government Funding
 - 5. Partnerships
 - 6. Crowdfunding and Public Awareness
 - 7. Proposal Focus

Key Components

1. Conversational Interface:

- Use the GPT model for natural language interaction to gather requirements like:
 - Area of interest (e.g., country, region, custom bounding box).
 - Type of data (e.g., land use, climate, population density, biodiversity).
 - Analysis needed (e.g., time-series trends, spatial correlations, feature classification).
- Example:
User asks: "Show me deforestation trends in Brazil's Amazon from 2000 to 2020."
System responds with relevant datasets and maps.

2. Data Integration:

- Pull open geospatial datasets from:
 - *OpenStreetMap (OSM)* for infrastructure and land use.
 - *NASA EarthData* for climate and environmental data.
 - *Copernicus Open Access Hub* for satellite imagery.
 - *Local portals* for government and academic datasets.
- Automate fetching and preprocessing data with tools like Python (requests, GDAL, rasterio).

3. Interactive Maps:

- Use *Mapbox GL JS* or *Deck.gl* for rendering maps in React or Vue.js.
- Incorporate features like:
 - Layer controls (toggle between datasets).
 - Time sliders for temporal datasets.
 - Filters for user-selected attributes (e.g., vegetation type, elevation range).
- Highlight regions of interest dynamically based on user input.

4. Geospatial Analysis:

- Perform basic analyses using Python libraries like **GeoPandas*, ***Shapely*, or **PostGIS*:
 - Overlay analysis: Combine datasets to identify intersections (e.g., deforestation and protected areas).
 - Heatmaps: Visualize population density or climatic variations.
 - Classification: Group land types or analyze spatial clusters.
- Integrate pre-trained or fine-tuned AI models for advanced tasks like object detection in satellite imagery.

5. Report Generation:

- Allow users to export analysis as a PDF or interactive web link.
- Include annotated maps, charts, and dataset references.

Step-by-Step Guide

1. Build the Foundation

- **Backend:** Use Python (e.g., FastAPI) to handle data queries, processing, and serving AI interactions.

- **Frontend:** Use React for a responsive interface with mapping libraries like **Mapbox GL JS** or **Leaflet**.
- **Database:** Set up PostgreSQL with PostGIS for geospatial data storage and querying.

2. Develop AI for User Interaction

- Train or fine-tune a GPT-like model to interpret user queries and map them to geospatial needs:
 - Example: "I want flood-risk maps for Southern Brazil." or "My business supply natural food for schools and I want to expand in the Okanagan region in Canada."
 - Model response: "I recommend combining precipitation data from NASA with land elevation models of the states of Paraná, Santa Catarina and Rio Grande do Sul. Would you like to include urban areas and forest vegetation?" or "I suggest using a British Columbia school location map with total students labels and district management buildings with addresses to help searching."

3. Integrate Open Data APIs

- Use APIs to fetch live data:
 - *OpenStreetMap*: OSM Overpass API.
 - *EarthData*: NASA's APIs for environmental and climate datasets.
 - *Local Data Portals*: National/regional open data platforms.

4. Implement Interactive Mapping

- Visualize datasets dynamically:
 - Display boundaries, raster layers, and vector overlays.
 - Add interactivity (e.g., click-to-view metadata or stats).

5. Add Geospatial Analytics

- Integrate Python geospatial libraries to:
 - Calculate areas of change (e.g., urban sprawl).
 - Create heatmaps or contour maps.

Test and Iterate

- Collaborate with real researchers to refine the system based on feedback.
- Expand dataset compatibility as needed.

Example Use Case

1. A climate researcher asks: "Show temperature trends and deforestation in the Amazon from 2000 to 2020."
 2. The platform fetches relevant datasets from NASA (temperature) and Global Forest Watch (deforestation).
 3. It generates an interactive map with:
 - A time slider showing deforestation over the years.
 - Overlaid temperature trends as a heatmap.
 - Exportable charts and analysis reports.
-

Learning Outcomes

- Hands-on experience with AI for user input processing.
- Familiarity with geospatial data formats (GeoJSON, shapefiles, raster).
- Proficiency in interactive mapping tools and open data integration.
- Insight into researchers' geospatial needs and how to address them.

System Workflow

1. User Input:

- Researchers provide a query or description of their needs (e.g., "I need a heatmap of population density in coastal regions prone to flooding in Southeast Asia.").

2. Local Model:

- Your local model:
 - Parses the input and extracts key details (area of interest, type of data, type of analysis).
 - Generates a structured prompt for the OpenAI API, ensuring it includes necessary details.

3. API Interaction:

- The prompt is sent to OpenAI's GPT-4 or GPT-3.5-turbo via the API.
- The larger model generates refined outputs, such as:
 - A detailed analysis plan.
 - Suggested datasets and methods.
 - Additional clarifying questions for the user.

4. Result Processing:

- The response from OpenAI is processed and used to:
 - Fetch or suggest relevant datasets.
 - Configure the interactive map automatically.
 - Provide feedback or suggestions to the user.

5. Final Output:

- The platform delivers visualizations, analytics, or an exportable report based on the AI-driven suggestions.

Implementation Steps

1. Local Model for Prompt Generation

- Use a lightweight local model (e.g., GPT4All or LLaMA) to preprocess user input:
 - Example Prompt Output:
plaintext The user wants to analyze coastal regions in Southeast Asia prone to flooding.
Objectives:
 - Visualize population density.

- Identify flood-prone areas using elevation and rainfall data.
Datasets: OpenStreetMap, NASA EarthData. Questions:
 - What year range should the analysis cover?
 - Should it focus on urban or rural regions?
- Fine-tune the local model if necessary for better domain-specific responses.

API Call to OpenAI

- Use the OpenAI Python client to send structured prompts from your local model:

```
python import openai  
  
openai.api_key = "your-openai-api-key"  
  
prompt = "Analyze flooding risks in Southeast Asia with the following parameters: ..."  
  
response = openai.ChatCompletion.create( model="gpt-4", messages=[ {"role": "system", "content":  
"You are a geospatial analysis assistant."}, {"role": "user", "content": prompt} ] ) result =  
response['choices'][0]['message']['content'] print(result)
```

2. Handle API Response

- Parse the API's output and integrate it into your application:
 - Fetch datasets using the suggested sources.
 - Configure layers and visualizations on your map.
 - Generate an analysis report based on the recommendations.

3. Build Feedback Loop

- Use the OpenAI response to improve local prompt generation:
 - Refine the local model's outputs based on feedback from the API.
 - Implement logging to analyze API responses and enhance your system.

Advantages

- *Efficiency*: Offload complex reasoning to GPT-4 while keeping local tasks lightweight.
 - *Scalability*: Reduce hardware dependency for advanced tasks by leveraging cloud-based AI.
 - *Cost Optimization*: Use the local model for simpler tasks, limiting API usage to when it's most beneficial.
-

Tech Stack

A.I. section

- *Local Model*: GPT4All, LLaMA, or similar lightweight models.
- *OpenAI API*: For leveraging GPT-4 or GPT-3.5-turbo.
- *Backend*: Python (FastAPI or Flask) for managing input/output between the local model and API.
- *Frontend*: React + Mapbox for interactive mapping.
- *Database*: PostgreSQL + PostGIS for spatial data.

Interactive Map Section

- *Frontend*: React, Mapbox GL JS, Deck.gl.
 - *Backend*: Python (FastAPI), Flask, or Django.
 - *Database*: PostgreSQL + PostGIS.
 - *AI*: GPT-like models for query interpretation (e.g., GPT4All, OpenAI API).
 - *Geospatial Tools*: GDAL, Rasterio, GeoPandas.
 - *OS*: Ubuntu for better support with geospatial and AI libraries.
-

Future Enhancements

- *User Feedback Loop*: Train the local model with user interactions and refine its prompts over time.
- *Caching*: Store frequently used OpenAI responses locally to reduce costs.
- *Hybrid Analysis*: Combine outputs from the local model and OpenAI to create richer insights.

Funding

1. Academic and Research Grants

- *UBC Internal Grants*:
 - Apply for *Graduate Research Awards* or *Innovation Grants* at UBC Okanagan.
 - Collaborate with your department or research supervisor to access internal funding pools for technology and innovation projects.
 - *Canadian Research Programs*:
 - *NSERC (Natural Sciences and Engineering Research Council of Canada)*:
 - Apply for *Discovery Grants* or *Collaborative Research and Development Grants*.
 - Focus on the project's geospatial data processing and machine learning aspects.
 - *Canada Foundation for Innovation (CFI)*:
 - Propose the need for high-performance computing resources for geospatial analysis.
 - *Interdisciplinary Grants*:
 - Look into funding from *SSHRC (Social Sciences and Humanities Research Council)* if the project touches on societal impacts, like urban planning or disaster mitigation.
-

2. Climate and Environmental Grants

- *Canada's Climate Action and Clean Growth Programs*:
 - Apply for funding under initiatives like *Climate Action Fund* or *Impact Canada's Clean Technology Initiative*.
 - Highlight the potential use of the platform for addressing climate change-related challenges like deforestation, urban heat islands, or flood risk.
- *International Opportunities*:

- *UNDP Climate Fund*: Showcase how your platform can aid sustainable development in vulnerable regions like Brazil.
 - *Global Environment Facility (GEF)*: Propose the project as a tool to support biodiversity and land-use planning.
 - *Brazilian Funding*:
 - Apply to *CAPES* (Coordination for the Improvement of Higher Education Personnel) for funding joint research with Brazilian institutions.
 - Partner with researchers in Brazil to access *FAPESP* or *CNPq* funding for geospatial and environmental research.
-

3. Corporate Sponsorships

- *OpenAI*:
 - Propose a collaboration where you showcase how OpenAI models can support geospatial and environmental research. Highlight the academic and social impact.
 - *AWS and Google*:
 - *AWS Activate for Startups* or *Google Cloud Research Credits*:
 - Leverage cloud computing resources for processing and hosting your geospatial data and models.
 - Submit a proposal focusing on the scalability of your platform using their infrastructure.
 - *Microsoft AI for Earth*:
 - Apply for AI and cloud computing grants under Microsoft's *AI for Earth* program.
 - Emphasize how the platform aids environmental research and global collaboration.
-

4. Government Funding

- *Canadian Government*:
 - Apply for funding from *Canada's Digital Technology Supercluster* for projects aligning with technological innovation and societal benefit.
 - Tap into *Infrastructure Canada's Disaster Mitigation and Adaptation Fund* to demonstrate how your project addresses natural disaster preparedness.
 - *Brazilian Government*:
 - Collaborate with institutions like *IBGE* or *INPE* to support geospatial data-sharing initiatives.
 - Emphasize the use of the platform for monitoring environmental and social issues in Brazil, such as deforestation or urbanization.
-

5. Partnerships

- *Industry Partnerships*:

- Partner with companies in geospatial tech, like *Esri, **Mapbox, or *Hexagon Geospatial, for funding or technology support.
 - Collaborate with NGOs or think tanks focusing on environmental and urban challenges.
 - *Academic Collaborations:*
 - Form alliances with universities in Brazil or other countries to share resources and co-author grant applications.
-

6. Crowdfunding and Public Awareness

- Launch a *public crowdfunding campaign* to attract support from environmentally conscious individuals and organizations.
 - Use platforms like *GoFundMe* or *Experiment.com* to raise awareness and funding.
-

7. Proposal Focus

When applying for funding:

- *Highlight Societal Impact:* Showcase how your platform addresses real-world problems, such as disaster resilience, climate change, and urban planning.
- *Emphasize Innovation:* Stress the use of AI and open data for scalable, impactful solutions.
- *Offer Collaboration:* Propose partnerships with the funding organization to refine and test their tools (e.g., OpenAI models, cloud platforms).