Sustainable Smart City Assistant Using IBM Granite LLM

Project Documentation

1. Introduction

- Project Title : Sustainable Smart City Assistant Using IBM Granite LLM
- Team leader : Ishani S
- Team member : Sandhiya B
- Team member : Swathi E

2. Project overview

- Purpose :

The purpose of this application is to provide an AI-powered assistant that supports sustainability and policy understanding through two core functionalities: eco-advice generation and policy summarization. By leveraging IBM Granite's advanced language model, the system helps citizens adopt sustainable practices by offering practical and actionable eco-friendly tips tailored to specific environmental issues such as waste management, energy saving, and water conservation. At the same time, it assists policymakers, researchers, and the public by simplifying complex policy documents, extracting the most important points, provisions, and implications to make them easier to understand and apply. Through its simple Gradio-based interface, the application bridges the gap between citizens and sustainability knowledge, promoting informed decisionmaking and active participation in building smarter and greener communities.

- Features:

  1. Eco Tips Generator
  2. Policy Summarization
  3. AI-Powered Insights

4. User-Friendly Interface
5. Flexible Input Options
6. Accessible Deployment

1. Eco Tips Generator

- Key Point: Provides sustainability advice tailored to specific environmental issues.
- Functionality: Accepts problem keywords (e.g., plastic, energy, water waste) and generates practical, actionable eco-friendly tips.

2. **Policy Summarization**

- Key Point: Simplifies complex policy documents for easier understanding.
- Functionality: Extracts text from uploaded PDFs or pasted text and produces concise summaries highlighting key provisions and implications.

3. **AI-Powered Insights**

- Key Point: Leverages IBM Granite LLM for intelligent language processing.
- Functionality: Uses advanced NLP to generate high-quality tips, summaries, and contextual insights for sustainability and governance.

4. **User-Friendly Interface**

- Key Point: Provides an easy-to-use, interactive dashboard.
- Functionality: Built with Gradio, featuring tab-based navigation, text inputs, PDF uploads, and real-time output display

5. **Flexible Input Options**

- Key Point: Allows multiple ways to provide data.
- Functionality: Supports both text input and PDF uploads for document analysis, ensuring accessibility for all users.

6. **Accessible Deployment**
   - Key Point: Runs seamlessly across environments.
   - Functionality: Can be launched locally or shared publicly with a single click (share=True), making it easy to use and demonstrate.

3. Architecture

Frontend (Gradio)

The frontend is developed using Gradio, which provides a simple and interactive user interface. It is organized into two main tabs: Eco Tips Generator and Policy Summarization. Users can input environmental keywords, upload PDF documents, or paste text directly, and the results are displayed instantly in structured text boxes.

Backend (Python)

The backend is written in Python and contains the application logic. It manages functions such as PDF text extraction (via PyPDF2), prompt construction, and communication with the AI model. The backend ensures smooth handling of user input, processing requests, and returning appropriate outputs.

LLM Integration (IBM Granite LLM)

The application integrates with the IBM Granite LLM (granite- 3.2-2b-instruct) using Hugging Face Transformers. This layer is responsible for natural language understanding and generation. It tokenizes user input, processes it through the model, and decodes the results into meaningful responses. The integration is optimized to run on both CPU and GPU, improving performance and efficiency.

Deployment Layer

The deployment layer is powered by Gradio’s hosting mechanism. The application can be executed locally for development and testing or shared with others using the share=True option, which provides a public link for quick access. This makes the system flexible for demonstrations, collaborations, and real-world usage.

4. Setup Instructions

Pre-requisites:

- Python 3.9 or later
- pip and virtual environment tools
- API keys for IBM Watsonx and Pinecone
- Internet access to access cloud services

Installation Process:

- Clone the repository
- Install dependencies from requirements.txt
- Create a .env file and configure credentials (API keys, database URL, etc.)
- Run the backend server using FastAPI
- Launch the frontend via Streamlit
- Upload data and interact with the modules

5. Folder Structure

- app/ – Main application folder.
    - eco_tips.py – Module for generating eco-friendly tips using LLM.
    - policy_summarizer.py – Module for extracting and summarizing policy text from PDF or direct input.
    - utils.py – Helper functions (e.g., PDF text extraction, response generation).
    - model_loader.py – Handles loading of IBM Granite LLM and tokenizer.

  - interface.py – Defines the Gradio UI layout (tabs, inputs, outputs).

- main.py – Entry point script to launch the Gradio app.
- requirements.txt – List of dependencies (transformers, torch, gradio, PyPDF2, etc.).
- README.md – Documentation about project purpose, setup instructions, and usage.
- .env – (Optional) Configuration file for storing API keys or environment variables.
- data/ – Folder for sample PDFs or test documents.

- outputs/ – Folder for storing generated summaries, eco-tips, or logs.

6. Running the Application

To start the project:

- Run the Python script (python main.py).
- Gradio will automatically launch a local server and provide both a local URL and a public shareable URL (because of share=True).
- Open the URL in your browser.
- Navigate between the two tabs (Eco Tips Generator and Policy Summarization).
- Upload PDFs or enter text directly and receive AI-powered outputs in real time.

Frontend (Gradio)

The frontend is implemented with Gradio Blocks, providing a clean tab-based interface:

Eco Tips Generator Tab → Accepts keywords and generates actionable ecofriendly tips.

Policy Summarization Tab → Allows PDF upload or direct text input to generate concise summaries with key points.

Components like textboxes, file upload, and buttons are organized into columns and rows for user-friendly layout.

Backend (Model & Processing)

Model: Uses IBM's granite-3.2-2b-instruct model from Hugging Face, loaded via transformers.

Functions:

- generate_response() → Handles prompt creation and text generation.
- extract_text_from_pdf() → Reads uploaded PDFs with PyPDF2.
- eco_tips_generator() → Generates sustainability tips.
- policy_summarization() → Summarizes uploaded or pasted policy documents.

Optimized for GPU execution if available, otherwise falls back to CPU.

LLM Integration

Powered by IBM Granite LLM through Hugging Face transformers.

Provides context-aware responses with controlled sampling (temperature=0.7, do_sample=True).

Handles long inputs with truncation and safe max token lengths.

Deployment Layer

Local Deployment: Launch with python main.py.

Public Access: share=True generates a temporary public Gradio link for external users.

Lightweight: No need for a separate frontend/backend split — Gradio serves both UI & model inference seamlessly

7. API Documentation

Backend APIs available include:

- POST /generate-eco-tips – Accepts environmental problem keywords and responds with AI-generated actionable sustainability tips.
- POST /summarize-policy – Accepts a PDF file or policy text and returns a summarized document highlighting key points, provisions, and implications.
- POST /chat/ask – Accepts a user query and responds with an AI generated message (for future chat integration).
- POST /upload-doc – Uploads documents and stores embeddings in Pinecone for semantic search.
- GET /search-docs – Returns semantically similar policies based on user natural language queries.
- POST /submit-feedback – Stores citizen feedback for review or analytics.

8. Authentication

Each endpoint is tested and documented in Swagger UI for quick inspection and trial during development.

This version of the project runs in an open environment for demonstration purposes.

However, secure deployments can integrate:

- Token-based authentication (JWT or API keys)
- OAuth2 with IBM Cloud credentials

- Role-based access (admin, citizen, researcher)
- Planned enhancements include user sessions and history tracking

9. User Interface

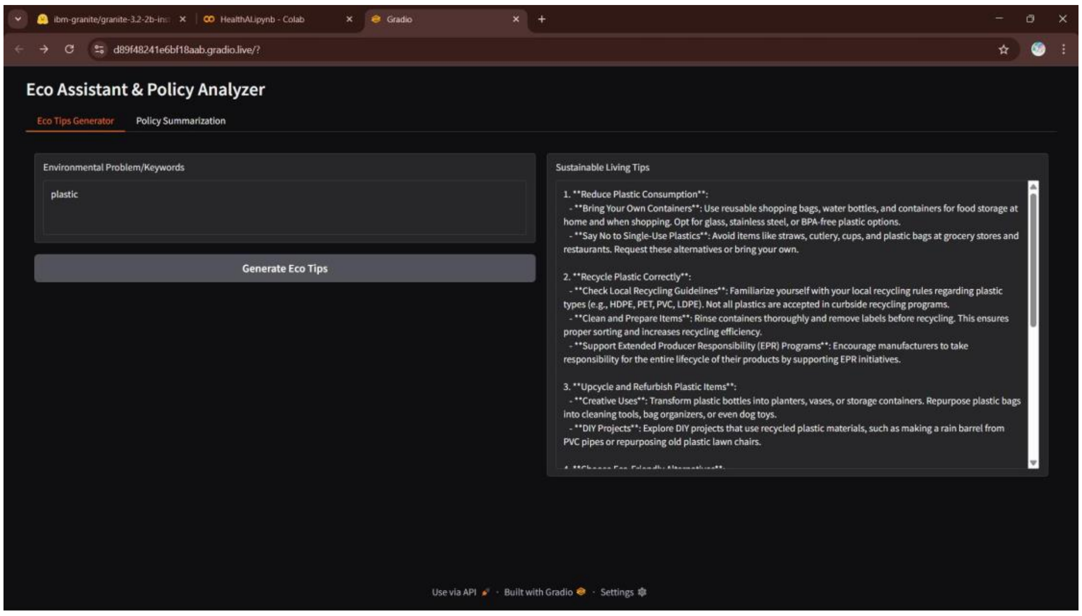
- The user interface is built using Gradio, providing a clean and interactive web platform.
- Organized into two main tabs: Eco Tips Generator and Policy Summarization.
- Eco Tips Generator:
  - Users input environmental problems or keywords.
  - Click Generate Eco Tips to get practical and actionable sustainability suggestions.
  - Suggestions are displayed in a dedicated output textbox.
- Policy Summarization:
  - Users can upload a PDF or paste policy text directly.
  - Click Summarize Policy to generate a concise summary highlighting key points, provisions, and implications.
- Tabbed layout allows seamless switching between tools.
- Flexible input options: text or PDF for policy summarization.
- Real-time dynamic outputs powered by the AI model.
- Designed to be user-friendly and accessible for non-technical users.

10. Testing

- Unit Testing: Conducted for all core functions, including prompt generation, PDF text extraction, and response formatting to ensure each module works correctly in isolation.

- API Testing: Performed using Swagger UI, Postman, and custom test scripts to verify endpoints, request/response accuracy, and error handling.
- Manual Testing: Executed on the Gradio interface to validate file uploads, keyword inputs, chat interactions, and output consistency.
- Edge Case Handling: Tested scenarios with malformed inputs, large PDF files, empty text fields, and invalid API keys to ensure graceful failure and informative error messages.
- Validation: Each module and function was confirmed to work reliably in both offline mode (local testing) and API-connected mode (real-time responses)

11. Screenshots


ibm-granite/granite-3.2-2b-ins
HealthAI.ipynb - Colab
Gradio
d89f48241e6bf18aab.gradio.live/?
Eco Assistant & Policy Analyzer
Eco Tips Generator
Policy Summarization
Environmental Problem/Keywords
Generate Eco Tips
Sustainable Living Tips
Use via API · Built with Gradio · Settings

plastic

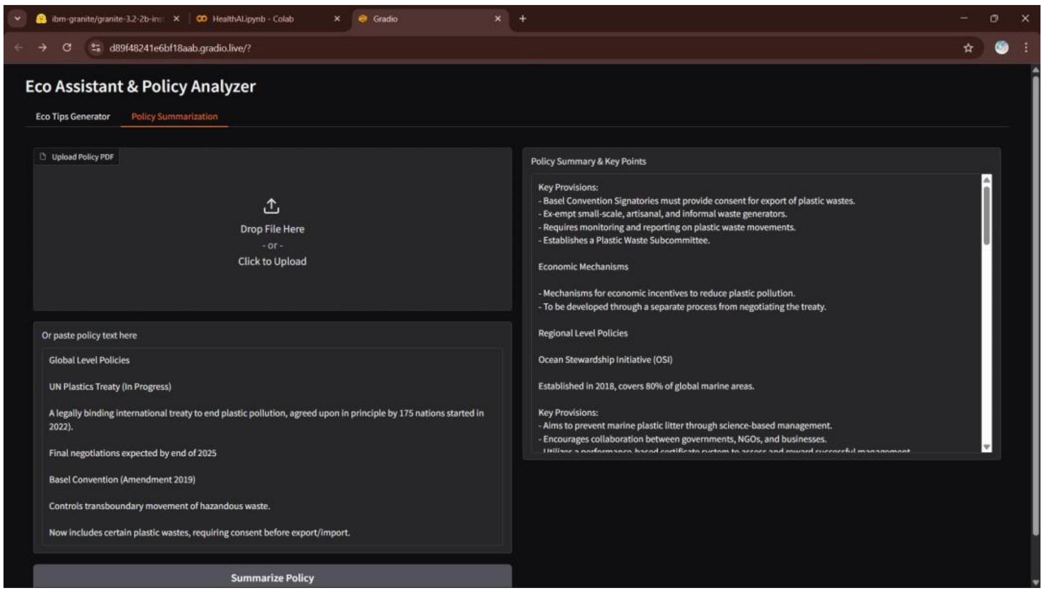
```
1. **Reduce Plastic Consumption**:
   - **Bring Your Own Containers**: Use reusable shopping bags, water bottles, and containers for food storage at home and when shopping. Opt for glass, stainless steel, or BPA-free plastic options.
   - **Say No to Single-Use Plastics**: Avoid items like straws, cutlery, cups, and plastic bags at grocery stores and restaurants. Request these alternatives or bring your own.

2. **Recycle Plastic Correctly**:
   - **Check Local Recycling Guidelines**: Familiarize yourself with your local recycling rules regarding plastic types (e.g., HDPE, PET, PVC, LDPE). Not all plastics are accepted in curbside recycling programs.
   - **Clean and Prepare Items**: Rinse containers thoroughly and remove labels before recycling. This ensures proper sorting and increases recycling efficiency.
   - **Support Extended Producer Responsibility (EPR) Programs**: Encourage manufacturers to take responsibility for the entire lifecycle of their products by supporting EPR initiatives.

3. **Upcycle and Refurbish Plastic Items**:
   - **Creative Uses**: Transform plastic bottles into planters, vases, or storage containers. Repurpose plastic bags into cleaning tools, bag organizers, or even dog toys.
   - **DIY Projects**: Explore DIY projects that use recycled plastic materials, such as making a rain barrel from PVC pipes or repurposing old plastic lawn chairs.

4. **Choose Eco-Friendly Alternatives**:
```

Fig 1.Eco tip


ibm-granite/granite-3.2-2b-in
HealthAI.ipynb - Colab
Gradio
d89f48241e6bf18aab.gradio.live/?

Eco Assistant & Policy Analyzer

Or paste policy text here

Global Level Policies

UN Plastics Treaty (In Progress)

A legally binding international treaty to end plastic pollution, agreed upon in principle by 175 nations started in 2022).

Final negotiations expected by end of 2025

Basel Convention (Amendment 2019)

Controls transboundary movement of hazandous waste.

Now includes certain plastic wastes, requiring consent before export/import.

Policy Summary & Key Points

Key Provisions:
- Basel Convention Signatories must provide consent for export of plastic wastes.
- Ex-empt small-scale, artisanal, and informal waste generators.
- Requires monitoring and reporting on plastic waste movements.
- Establishes a Plastic Waste Subcommittee.

Economic Mechanisms

- Mechanisms for economic incentives to reduce plastic pollution.
- To be developed through a separate process from negotiating the treaty.

Regional Level Policies

Ocean Stewardship Initiative (OSI)

Established in 2018, covers 80% of global marine areas.

Key Provisions:
- Aims to prevent marine plastic litter through science-based management.
- Encourages collaboration between governments, NGOs, and businesses.

Fig 2.Policy Summarization

12. Known Issues

- Large PDF Handling: Very large PDF files may take longer to process or cause memory issues in low-resource environments.
- Incomplete Text Extraction**:** Some PDFs with complex formatting or scanned images may not extract text accurately using PyPDF2.
- Response Truncation: Generated outputs may be truncated if the response exceeds the model's maximum token limit.
- Latency on First Run: Model loading on the first run can be slow, especially without GPU acceleration.
- No Authentication in Demo: The current demo runs in an open environment; endpoints are not secured by default.
- Limited Error Feedback: Some unexpected inputs may return generic error messages rather than detailed diagnostics.
- Browser Compatibility: Certain Gradio UI elements may not render consistently on older browsers.
- Resource Usage: High CPU/GPU usage during model inference may affect other processes on the host machine.

13. Future enhancement

Future enhancements for the Eco Assistant & Policy Analyzer include implementing robust authentication and authorization mechanisms, such as JWT-based token systems, OAuth2 integration, and role-based access control to ensure secure usage. The platform could also incorporate user session tracking and history logs, enabling personalized recommendations and analytics. Improved PDF and document handling using OCR for scanned documents and advanced parsing techniques would enhance accuracy. Integration of multi-language support and expanded AI capabilities, such as predictive sustainability insights, real-time data visualization, and interactive dashboards, is planned. Additionally, cloud deployment optimizations, scalable API infrastructure, and mobile-friendly interfaces are envisaged to broaden accessibility and performance.