citizen_ai.py Documentation

Overview

This project provides an AI-powered assistant for analyzing city safety data and assisting citizens with queries related to public services, government policies, and civic issues.

It leverages Hugging Face's `transformers` library with the Granite model and provides a user interface through Gradio.

Dependencies

- transformers
- torch
- gradio

Components

1. Model Initialization

- Loads `ibm-granite/granite-3.2-2b-instruct` model and tokenizer.
- Handles GPU/CPU device compatibility.
- Ensures proper padding token setup.

2. Function: `generate_response(prompt, max_length=1024)`

- Tokenizes input prompt.
- Moves tensors to GPU if available.
- Uses `model.generate` with temperature sampling to produce responses.
- Decodes and cleans the output.

3. Function: `city_analysis(city_name)`

- Constructs a detailed prompt for analyzing a city.
- Provides information on:
- Crime index & safety statistics
- Accident rates & traffic safety
- Overall safety assessment
- Calls `generate_response`.

4. Function: `citizen_interaction(query)`

- Constructs a prompt for government-related citizen queries.

- Calls `generate_response`.

5. Gradio Interface

- Built using `gr.Blocks`.

- **City Analysis Tab**: Accepts city name, outputs safety analysis.

- **Citizen Services Tab**: Accepts user queries, outputs government response.

- Launched with `app.launch(share=True)`.

Usage

1. Run the script in Google Colab (preferably with GPU runtime).

2. Install dependencies:

Command Line Example:

pip install transformers torch gradio

3. Launch the Gradio app.

4. Access city analysis or citizen services through the interface.

Notes

- Designed for public service applications.

- AI-generated responses should be validated against official government data before use in decision-making.