Tabwise Chrome Extension

Tabwise is a Chrome extension designed to enhance your browsing experience by categorizing tabs into common categories (e.g., e-commerce, sports, social) and generating summaries using AI models. The extension leverages local AI inference as well as Gemini Nano's experimental summarization capabilities.

Features

- **Tab Categorization:** Classifies websites into predefined categories.
- **Summarization API:** Provides AI-generated summaries for each open tab using the experimental Gemini Nano API in Chrome.
- **Local Inference:** Runs AI models locally for privacy-first categorization.

System Requirements

Tabwise General Requirements

- **Operating System:** macOS (tested on Mac M1/M2 silicon), Windows 10/11, or Linux.
- **Python Version:** Python 3.8 or newer.
- **Browser:** Google Chrome version 129.0.6639.0 or newer (Canary recommended).

Gemini Nano Requirements (for Summarization)

- **Operating Systems:**
  - macOS: Version ≥ 13 (Ventura).
  - Windows: 10 or 11.
  - Linux: Requirements not specified.
- **Storage:**
  - At least 22 GB of free storage on the Chrome profile volume.
  - Note: After the download, Chrome deletes the model if storage drops below 10 GB.
- **GPU:**
  - Integrated or discrete GPU with 4 GB minimum VRAM.
- **Network Connection:** Non-metered internet connection.
- **Important Notes:**
  - Gemini Nano is currently not supported on Chrome for Android, iOS, or ChromeOS.
  - The requirements may change as Gemini Nano is under active development.

Installation

Step 1: Set Up Chrome with Gemini Nano

1. **Download and Install Chrome Canary:**
   - Visit the Chrome Canary download page and install the latest version.

- Confirm your version is **129.0.6639.0 or newer**.

2. **Check System Requirements:**
   - Ensure your device meets the requirements for Gemini Nano, including storage and GPU capabilities.
3. **Enable Gemini Nano:**
   - Open a new tab and navigate to:

     ```
     chrome://flags/#optimization-guide-on-device-model
     ```

   - Set the flag **"BypassPerfRequirement"** to **Enabled**.
   - Relaunch Chrome.
4. **Enable the Summarization API:**
   - Open a new tab and navigate to:

     ```
     chrome://flags/#summarization-api-for-gemini-nano
     ```

   - Set the flag to **Enabled**.
   - Relaunch Chrome.
5. **Initialize Gemini Nano:**
   - Open Chrome DevTools (F12) and run the following commands in the console:

     ```
     await ai.summarizer.create();
     ```

     - This forces Chrome to schedule a model download.
   - Wait 3–5 minutes for the download to complete, then run:

     ```
     await ai.summarizer.capabilities();
     ```

     - Wait until the response changes to `"readily"`.
   - **Troubleshooting:**
     - If you encounter `"The model was available but there was not an execution config available for the feature."`, wait for 24 hours and try again.

Step 2: Install Tabwise Backend

1. **Create a Python Virtual Environment:**

```
python3 -m venv env
source env/bin/activate
```

2. **Install Dependencies:**

```
pip install -r requirements.txt
```

3. **Install ctransformers for Mac M1/M2 (if applicable):**

```
CT_METAL=1 pip install ctransformers --no-binary ctransformers
```

4. **Download Required Models:** Navigate to the backend folder and run the Model Manager to download TinyLlama:

```
cd backend
python model_manager.py ensure
```

The Model Manager CLI supports several commands:

- `ensure`: Downloads the required models(including TinyLlama).
- `download` --model MODEL_NAME: Download a specific model
- `verify`: Check if all models are valid
- `list`: Show all available models
- `info --model MODEL_NAME`: Show detailed information about a specific model Optional arguments:
- `--force`: Force download even if model exists
- `--models-dir PATH`: Custom directory for storing models (default: backend/models)
- `--cache-dir PATH`: Custom directory for model cache (default: backend/cache)
- `--max-cache-size GB`: Maximum cache size in GB (default: 4.0)

5. **Run the Backend Server:**

```
python server.py
```

The server should start on `http://localhost:8000`.
