API User and Deployment Manual

Team Scrumptious - Project Symmetry

DESCRIPTION:

Project Symmetry is a semantic comparator tool which will take any Wikipedia article from a user, and translate it to whatever language the user desires. The tool utilizes an ML powered backend to provide comparisons between the two articles regarding informational inconsistency in a diff view, allowing the user to examine what information is extra/missing in their native language. The purpose of this app is to combat open source misinformation for under-represented languages.

I. RUNNING APP

Installation:

Install the following tools prior to dependency configuration:

- Node.js: Latest version (e.g., 20.11.0, to run UI)
- Python: Version 3.8 - 3.11 (NLP library requirements prevent 3.12)
- npm: Version 8.19.4
- Electron: Version v26.2.4 (to run UI)

Next, clone the repo:

```
git clone https://github.com/grey-box/Project-Symmetry-AI.git
```

Running API:

1. Ensure that your current working directory is fastapi (i.e. run `cd fastapi`)
2. If you have not already, set up your virtual environment (`python3 -m venv venv`)

3. Switch to the virtual environment (`source venv/bin/activate` or `venv\bin\activate.bat` depending on OS)

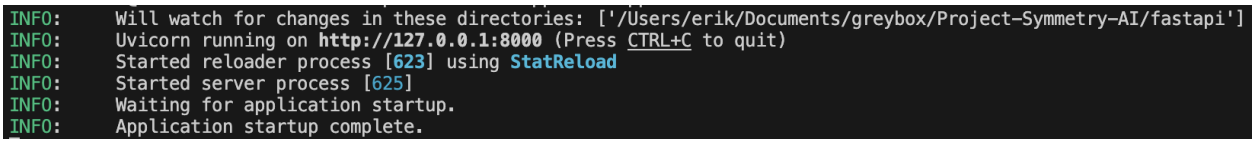
4. If you have not already, install dependencies (`pip install -r requirements.txt`)

5. Consider enabling debug logging:

    1. Create `.env` inside the `fastapi` folder
    2. Edit it and properties to change the logging level to debug and set the debug flag for FastAPI properties: `LOG_LEVEL=DEBUG FASTAPI_DEBUG=True`

6. Finally, run the app with Uvicorn (`uvicorn app.main:app --reload`) The app runs on port 8000 by default. To use another port, add `--port <number>` to the command.

You should see logs similar to the following:



To test the endpoints, use a tool like POSTMAN or curl in order to make GET or POST requests and receive formatted JSON responses.

II. FILE STRUCTURE

`app` is the root package of the python project.

`app/ai` contains the backend ML functionality.

`app/api` contains the API endpoints exposed by the web server.

`app/model` contains request and response structures for communication between the front and back end.

III. CURRENT FUNCTIONALITY

`/symmetry/v1/wiki/articles`

Protocol: `GET`

Parameters:

- `query`: A Wikipedia article URL or title
- `lang`: Optional. A language short code (i.e. "en" or "fr") defaulting to "en" for English.

Response:

```
{
    "sourceArticle": "<article content>",
    "articleLanguages": [
        // Article language data
    ]
}
```

`/symmetry/v1/articles/compare`

This endpoint is a work in progress and currently only returns dummy data.

Protocol: `POST`

Parameters:

```
{
    "article_text_blob_1": "<article 1 content>",
    "article_text_blob_2": "<article 2 content>",
    "article_text_blob_1_language": "<article 1 short language code>",
    "article_text_blob_2_language": "<article 2 short language code>",
    // Float ranging from 0-1 representing similarity percentage.
    "comparison_threshold": 0.8,
    "model_name": "<model name string>"
}
```

Response:

```
{
    "comparisons": [
        // Array of comparison objects.
        {
            "left_article_array": [
                // Array of article 1 content divided into subsections.
            ],
            "right_article_array": [
                // Array of article 2 content divided into subsections.
            ],
            "left_article_missing_info_index": [
                // Array of indices of article 1 subsections that are not present in 2.
            ],
            "right_article_extra_info_index": [
                // Array of indices of article 2 subsections that are not present in 1.
            ]
        }
    ]
}
```

IV. CONCEPTS & RESEARCH

Simple UI

The UI is supposed to be as simple as possible - the UI team is supposed to focus their efforts on improving the user experience, not covering all possible cases. That means that the API/middleware is responsible for the bulk of logic related to input validation.

For example, the dynamic search box allowing queries by URL or article title is handled by the wiki articles endpoint rather than the UI.

Semantic Comparison Without Translation

The language comparison model in use does not require that the input texts be in the same language.

JSON vs XML

We were tasked with investigating the benefits of using XML over JSON for communications as it might be easier for some ML models to work with.

Our findings were that the primary benefit of using XML for some ML models is that it can be used to more easily guard against prompt injection. It is otherwise largely harder to work with, and is a model-specific detail.

CORS

Cross-origin resource sharing is useful for protecting against certain attacks. By restricting resources to whitelisted domains, we can prevent this.

However, Symmetry currently operates as a local server. CORS will not be important until a future state where a standalone Symmetry server exists.

Endpoint Naming Conventions

A quick read on RESTful resource naming.

The current format is `/symmetry/v1/<path>/<to>/<resource>`

For a more dense and in-depth view of the RESTful philosophy, the above article links Roy Fielding's dissertation.