User Documentation

Requirements

- Python3

Install the requirements by running the following command from the root file

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

Creating YAML landscape file

To create a yaml file with the CNCF Lanscape follow this instructions:

1. Poblate with repository data

    1. You will need a Github token to access the API. Refer to Creating a personal access token. Copy and paste it in the appropiate location in the script landscape_explorer.py replacing "test_token".

    2. Go to the folder src/scripts

        ```
        cd src/scripts
        ```

    3. Execute landscape_explorer.py

        ```
        python landscape_explorer.py
        ```

2. Poblate with scraped data from websites

    1. Go to the folder src/landscape_scraper and execute

    ```
    scrapy crawl docs -O output.json
    ```

    2. Go to the folder src/scripts and execute:

    ```
    python augment_landscape.py
    ```

    3. The desired landscape_augmented_repos_websites.yml will be in the sources folder

Running Entire ETL and QA Processes (Tested on Ubuntu 20.04, Compatible with Linux and macOS)

The 'run_all.sh' script automates environment setup, ETL processes, and Q&A generation tasks.

Prerequisites

1. **Environment Variables**: Create a `.env` file in the root directory with the following content:

```
GITHUB_TOKEN=<YOUR_GITHUB_TOKEN>
HF_TOKEN=<YOUR_HUGGING_FACE_TOKEN>
```

Replace '<YOUR_GITHUB_TOKEN>' with your GitHub token obtained as described earlier, and '<YOUR_HUGGING_FACE_TOKEN>' with your Hugging Face token, which can be found at (https://huggingface.co/settings/tokens)

2. **Execute from Root Directory**: Run the script from the root directory of your project.

Usage

```
./script.sh [etl] [qa] <data_set_id>
```

Example

This command executes the ETL process, uploading the output to the specified dataset:

```
./script.sh SuperOrganization/WorldDataset
```

Training

You can find a jupyter notebook that you can use to train using Google Colab or, if you have the resources, locally, in `src/scripts/training/initial_colab_training.ipynb`. Additionally, if you want to train on a server, you can find necessary scripts in `src/hpc_scripts`. Copy this directory and then follow the instructions below.

To execute an example training script, run

```
./training_job.sbatch
```

in

```
src/hpc_scripts/training
```

This will start

```
src/hpc_scripts/training/model_training.py.
```

The hyperparameters were found using hyperparameter tuning, they might need to get changed to your specific use case.

Local-ai support

If you want to use the model with Local-ai, run local-ai in a docker container, using a docker image provided by local-ai from docker hub. You also need to pass a model configuration file to the docker container to tell local-ai which model to implement. All necessary commands are provided in

```
src/scripts/GUI/preparation_scripts.sh
```

> [!NOTE] If you want to use a GPU with local-ai, you need to:

1. Install Nvidia driver and cuda toolkit.
2. Install Nvidia container toolkit.
3. Pull and run local-ai image from docker hub. You can find all necessary commands in

```
src/scripts/GUI/preparation_scripts.sh
```

aswell.

Accessing the Model

The CNCFLLM model can be accessed through a CLI or local-ai as described above, providing an interface similar to ChatGPT. Follow the steps below to interact with the model:

Step 1: Open the Chat Interface

1. Go to the local-ai ChatUI web page.
2. You will be presented with a chat window where you can type your queries.

Step 2: Ask Questions

1. In the chat window, type your question related to CNCF projects and press Enter.
2. The CNCFLLM will process your question and provide an elaborated answer.

Step 3: Review Responses

1. Review the response given by the model.

2. If needed, you can ask follow-up questions or request more details for better clarification.

Example Queries

Here are some examples of the types of questions you can ask the CNCFLLM:

- "What is Kubernetes?"
- "How do I set up a CI/CD pipeline using Jenkins?"
- "What are the key features of Prometheus?"

Tips for Best Results

- **Be Specific**: The more specific your question, the more accurate the response.
- **Use Clear Language**: Avoid using slang or overly complex sentences.
- **Ask One Question at a Time**: This helps the model to provide focused and detailed answers.

Troubleshooting

If you encounter any issues while using the ChatUI, here are some common troubleshooting steps:

- **No Response**: Refresh the page and try asking your question again.
- **Inaccurate Answers**: Rephrase your question for clarity or provide more context.
- **Technical Issues**: Ensure you have a stable internet connection. If the problem persists, check the HuggingFace support page for assistance.

Contact and Support

For further assistance, you can visit our GitHub repository for more information and updates.