Usage Tutorial:

Enhance MeTA Toolkit and Metapy usability

David McGuire and Jose Cols
{dmcguire,josec5}@illinois.edu

Overview

MeTA and Metapy are unmaintained

Because of the complexity of compiling their source code on different platforms, these projects have become difficult to use.

MeTA

- Correct the CMake compilation of C++ source code.
- Add support for containerized workflows based on Docker with CI builds.

Metapy

- Update CI builds to include support for currently maintained versions of Python.
- Add Google Colaboratory support to tutorials based on Jupyter Notebook.
- Add support for containerized workflows based on Docker with CI builds.

Prerequisites

Docker

- Docker makes it significantly simpler to deliver software packages in containers using OS virtualization.
- Installation: https://docs.docker.com/get-docker/

Google Colaboratory

- Sign in to Google Apps account through Illinois with `@illinois.edu` email and password
- See Welcome to Colaboratory for usage guide

Installation instructions

Metapy Python Wheel

- A Python Wheel is a built-package format for distributing Python software that can be easily installed.
- To install the **metapy** wheel, select the package corresponding to your Python and OS versions from the releases page.
- Install it using the PIP command. For example:

  -
    ```
    pip install https://github.com/illinois/metapy/releases/download/v0.2.14/metapy-0.2.14-cp38-cp38-manylinux_2_24_x86_64.whl
    ```

Installation instructions

MeTA Docker image

- Once Docker is installed, installing any image is quite simple using its Command-line interface. To install MeTA's Docker image, run the following command:

```
docker pull josecols/meta:3.0.2
```

Metapy Docker image

- To install Metapy's Docker image, run the following command:

```
docker pull josecols/metapy:0.2.14
```

Example use cases

Running metapy in Colab

- Running Jupyter Notebooks in `tutorials` folder locally requires substantial expertise
- By comparison, running in Google Colaboratory (Colab) is streamlined
- Promotes experimentation, since the hosted site allows 5 concurrent environments via Illinois account

Running metapy on macOS

- Hang defect in `pybind11 < v2.3.0` for `Python >= 3.7` on macOS
- Stock `metapy` build for `Python 3.7` does not work with MPs
- Patched rebuild for `Python 3.7` does work with MPs

Example use cases

Running MeTA commands with Docker

- Any MeTA command can be run by opening a bash session in the Docker container. To do so, execute the following command:
    ```
    docker run -it --rm --name meta --entrypoint bash josecols/meta:3.0.2
    ```

- Because the MeTA toolkit executables are in the PATH, they can be run from any directory. All the available tools can be found on MeTA's website.

- If you wish to use files on your host machine as inputs for MeTA commands, you can use a Docker Volume. For example, to run the "profile" program on a specific "doc.txt" file on your machine, use the following commands:
    ```
    docker run -it --rm --name meta --mount type=bind,source=$(pwd),target=/app
    --entrypoint bash josecols/meta:3.0.2
    ```
    ```
    profile /meta/config.toml doc.txt --stop
    ```

Example use cases

Running metapy Python scripts with Docker

- Run the Docker container using the directory containing the script as the working directory.

  -
    ```
    docker run -it --rm --name metapy --mount type=bind,source=$(pwd),target=/app
    --entrypoint bash josecols/metapy:0.2.14
    ```
- Install other dependencies, if needed.

  -
    ```
    pip install -r requirements.txt
    ```
- Run the Python script inside the docker container:

  -
    ```
    python script.py
    ```

Other resources

MeTA repository

MeTA: Modern TExt Analysis

Metapy repository

metapy: (experimental) Python bindings for MeTA

Meta-Toolkit web site

MeTA: A Modern C++ Data Sciences Toolkit