Lumu test

Configuration

About dynaconf

This script uses dynaconf to manage settings and secrets keys. Because of this, it is necessary that you set
the queries file path and the Lumu secret key before of running the programm

1. Open the `settings.toml` file, and set the QUERIES_FILE_PATH variable to the path where the queries file is. By default, it is set to 'queries', which is the example file provided by Lumu
2. Open the `.secrets.toml` file and set the LUMU_CLIENT_KEY variable with the unique Lumu API Key from your account

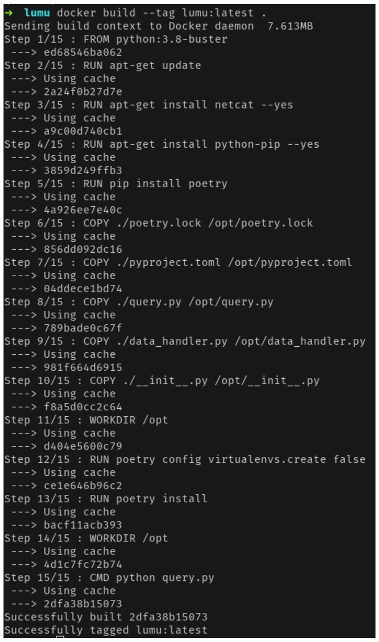
About docker

I have used docker to create a container with the necessary python packages to run the program. Please install docker following the instructions at https://docs.docker.com/engine/install/ubuntu/

How to run the program

1. Go to the path where the Dockerfile is located
2. Run

```
docker build --tag lumu:latest .
```

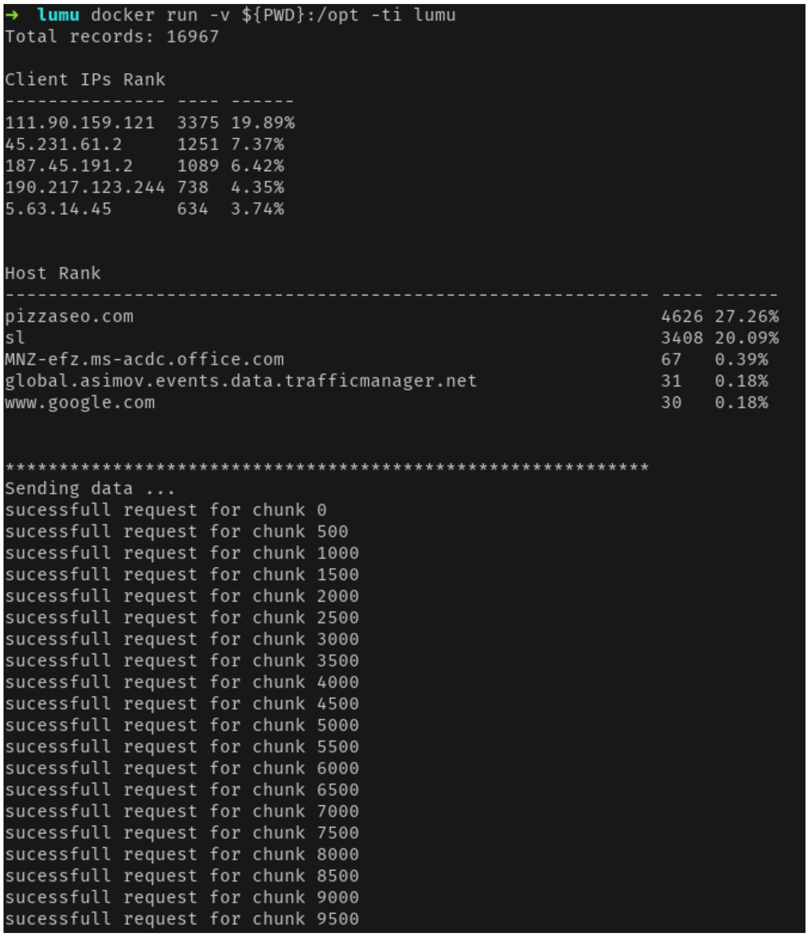


```
➜  lumu docker build --tag lumu:latest .
Sending build context to Docker daemon  7.613MB
Step 1/15 : FROM python:3.8-buster
 ---> ed68546ba062
Step 2/15 : RUN apt-get update
 ---> Using cache
 ---> 2a24f0b27d7e
Step 3/15 : RUN apt-get install netcat --yes
 ---> Using cache
 ---> a9c00d740cb1
Step 4/15 : RUN apt-get install python-pip --yes
 ---> Using cache
 ---> 3859d249ffb3
Step 5/15 : RUN pip install poetry
 ---> Using cache
 ---> 4a926ee7e40c
Step 6/15 : COPY ./poetry.lock /opt/poetry.lock
 ---> Using cache
 ---> 856dd092dc16
Step 7/15 : COPY ./pyproject.toml /opt/pyproject.toml
 ---> Using cache
 ---> 04ddece1bd74
Step 8/15 : COPY ./query.py /opt/query.py
 ---> Using cache
 ---> 789bade0c67f
Step 9/15 : COPY ./data_handler.py /opt/data_handler.py
 ---> Using cache
 ---> 981f664d6915
Step 10/15 : COPY ./__init__.py /opt/__init__.py
 ---> Using cache
 ---> f8a5d0cc2c64
Step 11/15 : WORKDIR /opt
 ---> Using cache
 ---> d404e5600c79
Step 12/15 : RUN poetry config virtualenvs.create false
 ---> Using cache
 ---> ce1e646b96c2
Step 13/15 : RUN poetry install
 ---> Using cache
 ---> bacf11acb393
Step 14/15 : WORKDIR /opt
 ---> Using cache
 ---> 4d1c7fc72b74
Step 15/15 : CMD python query.py
 ---> Using cache
 ---> 2dfa38b15073
Successfully built 2dfa38b15073
Successfully tagged lumu:latest
```

3. Run

```
docker run -v ${PWD}:/opt -ti lumu
```

This command will sent to stdout the ip and hosts rank and the POST requests results



```
➜  lumu docker run -v ${PWD}:/opt -ti lumu
Total records: 16967

Client IPs Rank
--------------- ---- ------
111.90.159.121  3375 19.89%
45.231.61.2     1251 7.37%
187.45.191.2    1089 6.42%
190.217.123.244 738  4.35%
5.63.14.45      634  3.74%


Host Rank
------------------------------------------------------------ ---- ------
pizzaseo.com                                                 4626 27.26%
sl                                                           3408 20.09%
MNZ-efz.ms-acdc.office.com                                   67   0.39%
global.asimov.events.data.trafficmanager.net                 31   0.18%
www.google.com                                               30   0.18%


************************************************************
Sending data ...
sucessfull request for chunk 0
sucessfull request for chunk 500
sucessfull request for chunk 1000
sucessfull request for chunk 1500
sucessfull request for chunk 2000
sucessfull request for chunk 2500
sucessfull request for chunk 3000
sucessfull request for chunk 3500
sucessfull request for chunk 4000
sucessfull request for chunk 4500
sucessfull request for chunk 5000
sucessfull request for chunk 5500
sucessfull request for chunk 6000
sucessfull request for chunk 6500
sucessfull request for chunk 7000
sucessfull request for chunk 7500
sucessfull request for chunk 8000
sucessfull request for chunk 8500
sucessfull request for chunk 9000
sucessfull request for chunk 9500
```