README.md

- Clone https://github.com/TranslatorSRI/Plater

Copy the `.env-template` file to `Plater/PLATER/.env` and edit as follows:

```
NEO4J_USERNAME=neo4j
NEO4J_PASSWORD=test
```

If using Docker to run Plater (note, we didn't ultimately run Plater in Docker):

Update L10 of the Dockerfile:

```
COPY .env .
```

We ended up running Neo4j in a Docker container:

```
docker pull neo4j:4.4.26
docker run --publish=7474:7474 \
           --publish=7687:7687 \
           --volume=$HOME/neo4j/data:/data \
           neo4j:4.4.26
```

Upload the KG (in JSON-lines format, one nodes file and one edges file) into Neo4j:

```
kgx neo4j-upload \
           --uri bolt://localhost:7687 \
           --username neo4j \
           --password test \
           --input-format jsonl \
           ./kg2c-test-nodes.jsonl \
           ./kg2c-test-edges.jsonl
```

Do this only if we resort to using Docker to run Plater (which will require special network bridging):

```
cd PLATER
docker build --tag plater .
docker run --env-file .env --name plater -p 8080:8080 plater
```

This is our preferred option for running Plater

- Run plater in the host OS:

```
./main.sh
```

- Then navigate to:

```
localhost:8080/docs
```

To Stop the running Neo4j and then run a new neo4j container with `-it`:

```
docker run -i -t --rm \
    -v $PWD/data:/data \
    -v $PWD/backups:/backups \
    --entrypoint /bin/bash \
    neo4j:4.2
```

RENCI wants a Neo4j "dump" file of KG2c. So, run this inside the Neo4j container:

```
neo4j-admin dump --to=/foo/kg2.db.dump
```

Give the `kg2.db.dump` file to Evan Morris who can generate the other needed artifacts from the db dump file.