

首先再次感谢贡献开源项目的作者！

一、搭建环境

1.1、下载项目，项目预览

https://github.com/qg547276542/Agriculture_KnowledgeGraph

1.2、docker 安装

Ubuntu 系统下：sudo apt install -y docker.io

1.3、neo4j 的介绍和启动

1.3.1、拉取 neo4j 镜像

\$sudo docker pull neo4j: 3.1.2

1.3.2、启动 neo4j 容器

\$docker run -it -d -p 7474:7474 -p 7687:7687 neo4j: 3.1.2

#启动镜像，其中 7474 用于管理界面，7687 用于连接 neo4j 服务端

-v 的参数，冒号前的是本机参数，后是镜像参数

1.3.3、查看启动的 neo4j 镜像

\$docker ps

1.4、打开 Neo4j 的 web 界面

<http://localhost:7474>

其中默认密码是 neo4j，第一次登陆后，记得修改。

在调试该项目的代码时，如果遇到 neo4j 相关的错误，可能是代码里写了密码，没有修改成自己的

1.5、安装环境

\$pip3 install -r requirement.txt

遇到错误：No module name Cython 解决：pip3 install Cython

注意，Neo4j 的 python 驱动，需要安装 2、3 版的，4 版要改代码：

\$ pip install py2neo==3.1.2

二、导入数据

这部分主要照着 GitHub 做就可以了。

需要注意的是，第一，要先把 csv 文件放到了 Neo4j 的 import 目录下

第二，会遇到内存问题。

2.1、导入数据

2.1.1、导入 hudong_pedia.csv 到 neo4j 的 import 目录下

```
$sudo docker cp hudong_pedia.csv {CID}:/var/lib/neo4j/import
```

{CID}换成你的 neo4j 容器的 ID，可以用 `sudo docker ps` 查看，只需要输入 3、4 位就行

2.1.1、在 neo4j 的 web 控制台，执行导入命令：

```
LOAD CSV WITH HEADERS FROM "file:///hudong_pedia.csv" AS line
```

```
CREATE
```

```
(p:HudongItem{title:line.title,image:line.image,detail:line.detail,url:line.url,openTypeList:line.openTypeList,baseInfoKeyList:line.baseInfoKeyList,baseInfoValueList:line.baseInfoValueList})
```

【遇到问题】：Please try increasing ‘dbms.memory.heap.max_size’ in the neo4j configuration(normally in ‘conf/neo4j.conf’

(解决方案一) docker 启动的时候加上参数：

```
--env=NEO4J_dbms_memory_heap_maxSize=2048M
```

但是有的时候不会生效。

(解决方案二) 注意，遇到这个问题，是因为一次读入的文件比较大，而 neo4j 默认运行内存是 500M，neo4j 可以执行命令的时候，每读几行就执行，而不是读完整个文件再执行，所以执行导入数据在命令前边加上“USING PERIODIC COMMIT 300”，即每 300 行提交一次即可，完整命令如下：

```
USING PERIODIC COMMIT 300
```

```
LOAD CSV WITH HEADERS FROM "file:///hudong_pedia.csv" AS line
```

```
CREATE
```

```
(p:HudongItem{title:line.title,image:line.image,detail:line.detail,url:line.url,openTypeList:line.openTypeList,baseInfoKeyList:line.baseInfoKeyList,baseInfoValueList:line.baseInfoValueList})
```

2.2、建立索引，按照 GitHub 说明执行即可

2.3、继续生成数据

进入 `wikidataSpider/wikidataProcessing`

运行 `relationDataProcessing.py`，注意修改密码 neo4j 的密码，在第 10 行

然后按照 GitHub 说明继续重复导入

三、启动服务：

3.1、进入 demo/Model/neo_models.py,修改第 9 行的 neo4j 账号密码，改成你自己的

3.2、在 demo 目录下，执行 `sudo sh django_server_start.sh`

注意在 demo 下的 setting，关于 allow ip 的

目前涉及图的展示，会有一个小小的彩蛋，需要调试一下。相关的数据查询可以在 neo4j 自己查数据验证一下。Neo4j 的语法网上教程很多，请自行查看。

四、爬虫相关：

主要看 wikidataSpider 目录下的 readme，建议先自学一下 scrapy。

五、实体分类：

KNN 的分类模型，用了 fasttext 的模型，现在过期下载不到了，可以尝试自己训一个。重点看他是如何构造特征工程的。

六、其他：

实体识别和该项目的对话，请看它在 GitHub 首页的介绍，具体代码可以在 demo 下找到，

对话看 demo/demo/question_answering.py

关系抽取下次再讲，可以预习