

用户手册

代码结构

- `data/`
 - 存放了数据处理后的所有文件, 包括分词后的文本, 各个字典, 图的邻接矩阵以及导入Neo4j时需要使用的csv文件
- `utils.py`
 - 封装了各个实用函数
- `preprocess.ipynb`
 - 生成 `data/` 下各个文件时使用的各个函数, 由于 `data/` 已经准备完全, **用户无需运行该文件**
- `import.neo4j`
 - 将节点和关系导入Neo4j数据库中使用的终端命令
- `main.py`
 - 主程序

Prerequisite

- 安装 Neo4j
- 安装 py2neo 包

构建图

- 创建新的空数据库名为 `NGM`, 密码 `123`
- 将 `data/Nodes_whole.csv` 和 `data/Relationship_whole.csv` 复制到 `NGM` 数据库对应的 `<neo4j-home>/import` 目录下
- 打开 `NGM` 数据库的终端, 运行 `import.neo4j` 中的内容, 即

```
bin\neo4j-admin import --nodes=import/Nodes_whole.csv --relationships=import/Relationships_
```

- 启动 `NGM` 数据库

运行主程序

- 更改 `main.py` 中图数据库的端口为自己的 (默认是7687)

- `cd Project/`
`python main.py`

- 提供七个功能, 输入**功能编号**即可

- **#0:基本信息查看**

- **输出**热门人物, 热门机构, 图的节点数, 边数, 连通分量个数, 最大连通分量节点数

- **#1:查询邻居**

- **输入**一个实体名称
 - **输出**与之相邻的共现次数最大的10个实体及其共现次数

- **#2:计算PageRank**

- **输出**PageRank最大的20个实体及其PageRank值

- **#3:最短路径计算**

- **输入**source实体节点名称, target实体节点名称
 - **输出**两者之间最短路径长度

- **#4:社区挖掘**

- 提供**流**和**统计量**两种模式

- **流模式**

- **输入**1

- **输出**所有社区及其包含的节点到json文件 `data/community.json`

- **统计量模式**

- **输入**2

- **输出**所有社区的个数, 以及平均modularity

- **#5:中介中心性计算**

- **输入**实行计算的网络编号: **#1:人物共现网络, #2:机构共现网络, #3:实体共现网络**
 - **输出**对应网络中中介中心性最高的前10个节点及其中介中心性的值

- **#6:聚集系数计算**

- **输入**实行计算的网络编号: **#1:人物共现网络, #2:机构共现网络, #3:实体共现网络**
 - **输出**对应网络中聚集系数最高的前10个节点及其据聚集系数的值

- 除计算节点中介中心性, 计算节点聚集系数外, 其余功能默认在**实体共现网络**中运行

- 所有功能在主程序交互页面均有详细显示, 如果输入不存在的节点, 则会返

回 `Node Not Valid! / You are KIDDING ME!`