

Đại học Công Nghệ Thông Tin
Đại Học Quốc Gia Tp. Hồ Chí Minh

Visualization of Graph Database

với Python và Neo4j



Thầy hướng dẫn: Phó Giáo Sư Đỗ Phúc
Sinh viên: Lê Hưng

Year of 2018

Giới thiệu

Chương được xây dựng để đọc dữ liệu về các bài báo (Paper), tác giả (Author), hội thảo (Venue), chủ đề (Topic), từ (Words) và mối quan hệ giữa chúng để lưu vào Graph Database. Các mối quan hệ bao gồm:

- + Author - wrote -> Paper
- + Paper - published -> Venue
- + Paper - cites -> Paper
- + Paper - belongsTo -> Topic
- + Topic - includes -> Word

Sau khi đã lưu dữ liệu vào database, chương trình có khả năng thực hiện trực quan hoá dữ liệu và tính toán Centrality.

Copyright

Các bạn sinh viên có thể sử dụng source code cho mục đích học tập và nghiên cứu của mình.

Các bạn sinh viên không được sử dụng hoàn toàn source code để nộp bài tập đồ án môn học cá nhân/nhóm.

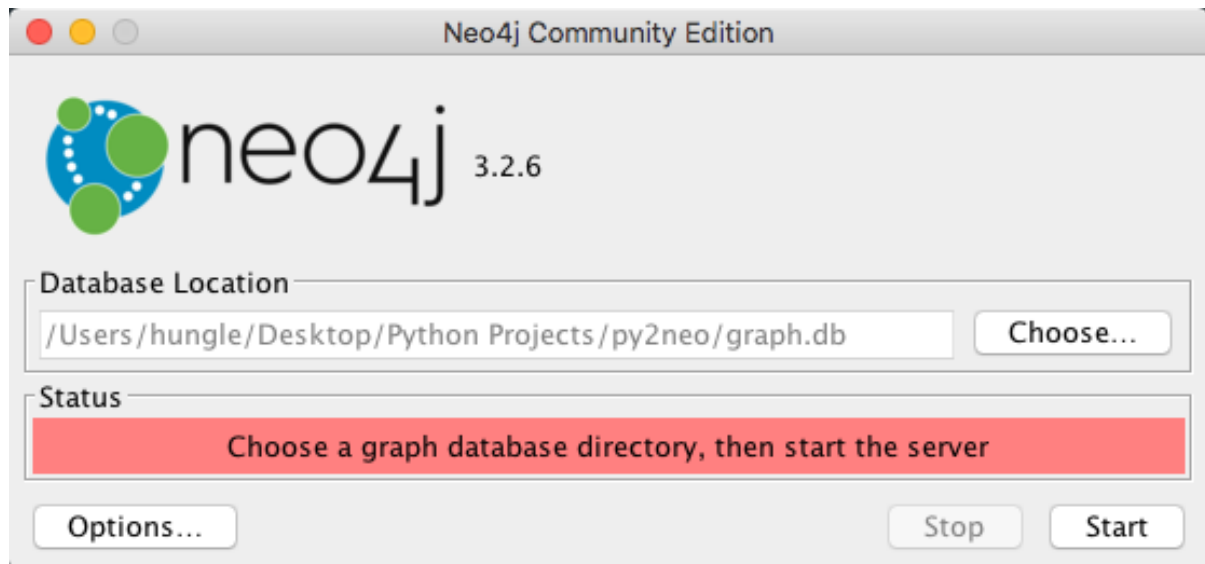
I. Các phần mềm cần thiết để có thể chạy được chương trình:

- Python 3.6
- Neo4j 3.2: <https://neo4j.com> (Để lưu trữ cơ sở dữ liệu dạng Graph)
- NetworkX
- Matplotlib
- py2neo
- IDE có thể chạy được python (Có thể dùng Pycharm hoặc Sublime Text + Anaconda)

II. Hướng dẫn cài đặt:

- Tùy vào hệ điều hành, các bạn có thể xem cách cài đặt python 3.6 vào máy của mình. NetworkX, Matplotlib, py2neo là các thư viện của Python cũng đã có documentation rất cụ thể, cách cài đặt cũng rất dễ dàng.

- Đối với Neo4j, sau khi cài đặt và khởi động lên các bạn sẽ thấy một cửa sổ như sau:



Các bạn có thể nhấn vào Choose... để chọn vị trí lưu database. (trong hình đã chọn vị trí lưu database vào folder có tên graph.db) Nhấn Start để bắt đầu khởi động. Sau khi khởi động thành công Neo4j sẽ thông báo, các bạn có thể vào <http://127.0.0.1:7474/> để kiểm tra.

* Lưu ý: Lần đầu khởi tạo và truy cập, Neo4j sẽ yêu cầu nhập mật khẩu. Mật khẩu mặc định ban đầu là neo4j. Sau đó các bạn sẽ phải reset mật khẩu để có thể tiếp tục sử dụng.

III. Một số câu lệnh cơ bản để tương tác với database bằng Python:

Để kết nối với database:

```
graph = Graph("http://neo4j:neo4j2@localhost:7474/db/data/")
```

Với neo4j là username, neo4j2 là password

Để tạo 1 node mới:

```
newNode = Node(*args, **kwargs)
create(newNode)
```

Để run 1 Cypher query và return 1 node:

```
graph.evaluate('MATCH...WHERE id=$id', id=4, **kwargsparameters)
```

Để tìm một node dựa vào label và các properties:

```
a = graph.find_one('Paper', 'id', 4) #find_one(*args, **kwargs)
```

Để tạo 1 relationship giữa 2 node:

```
rel = Relationship(node1, 'RelType', node2)
create(rel)
```

Để có thể tìm hiểu thêm về những câu lệnh tương tác với neo4j, các bạn có thể tham khảo documentation của py2neo: <http://py2neo.org/v3/>

IV. Về source code:

Source code gồm có 2 phần:

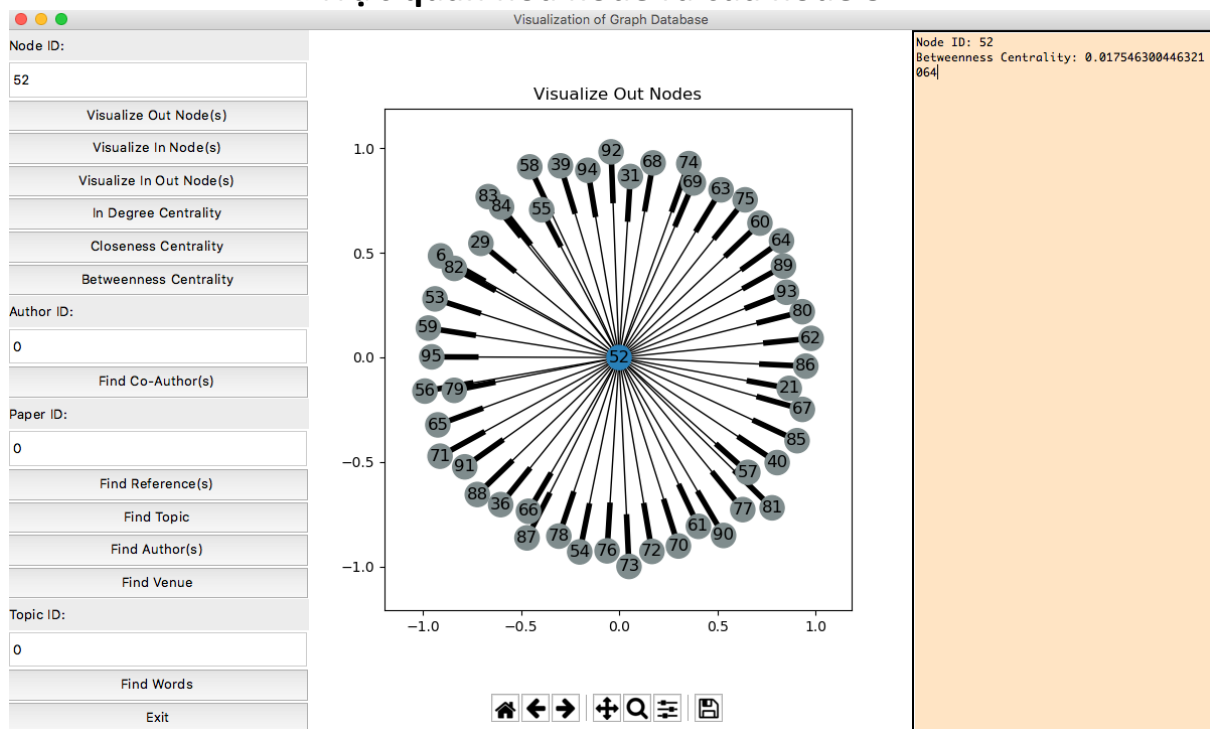
- Phần 1: file importToDatabase.py sẽ xóa toàn bộ dữ liệu cũ trong database, đọc dữ liệu từ file data.txt và lưu vào Neo4j dưới dạng các Nodes và Relationships. Phần này không có GUI.

- Phần 2: gồm 3 pyController.py, pyModel.py, pyView.py. Phần này các bạn chỉ cần chạy file pyController.py, file này sẽ gọi đến 2 file pyModel.py và pyView.py để hiện lên GUI. Lúc này các bạn có thể tương tác với database qua giao diện này.

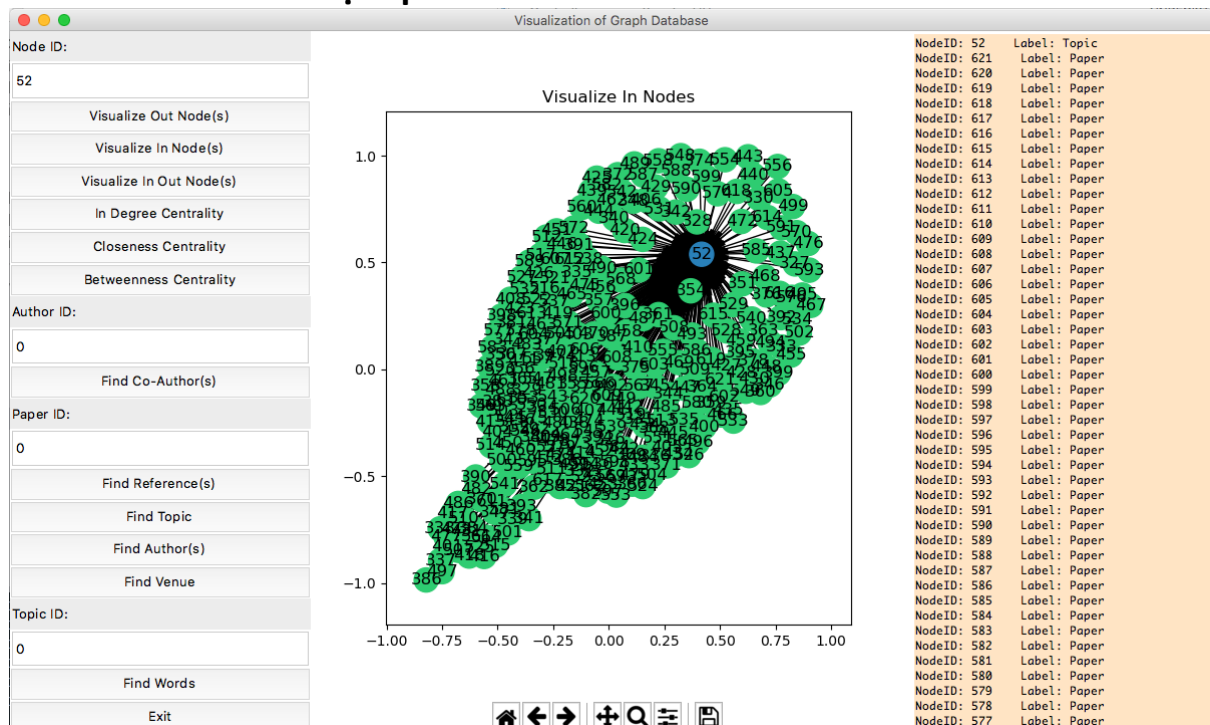
*Lưu ý: trước khi thực chạy file trong phần 1 hay phần 2 các bạn đều phải khởi động Neo4j database. Với phần 1, các bạn cần sửa địa chỉ đường dẫn đến file test phù hợp với máy của bạn.

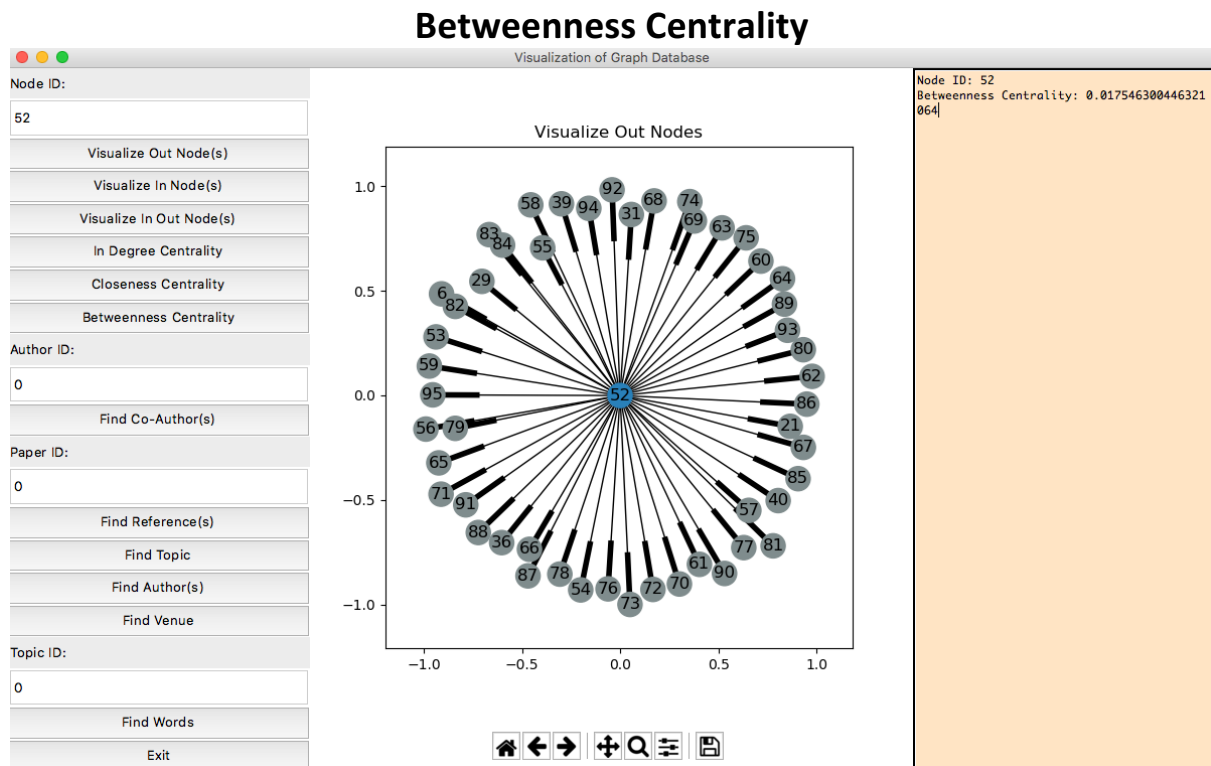
V. Một số hình ảnh của chương trình:

Thực quan hoá node ra của node 52



Trực quan hoá node vào của node 52





VI. Liên hệ:

- Các bạn có thắc mắc hay góp ý gì có thể liên hệ theo địa chỉ email:
hungle1abc@gmail.com

Cảm ơn Thầy Đỗ Phúc đã giúp em tìm hiểu về Graph Database
 và viết chương trình này!