

Graph based search

Documentation



July 16, 2019

Andy Ang Yong Kiat

Contents

[File Directory Explanation 2](#_Toc14337280)

[How to Run the Program 2](#_Toc14337281)

[How to Store Data into Neo4j 2](#_Toc14337282)

# File Directory Explanation

\***Bold** words are folders\*

**app**

* \_\_init\_\_.py - Contains all the methods
* requirements.txt  - Contains all our venv dependencies
* run.py - Where our flask app will be run from, the main method
* clustering.py and k\_means.py - Perform document clustering and output processeddf dataframe
* processeddf - Contains clustering results in dataframe
* neo4j.py - Inserting clustering results into Neo4j
* collaborativefiltering.py – To test out collaborative filtering
* searchesdata.xlsx – Used to generate searches.csv
* usersdata.xlsx – Used to generate users.csv
* searches.csv – Contains user searches
* users.csv – Sontains user data
* **documents/documents2** - Contains documents scraped for clustering
* **scrapyapp** - Contains scrapyapp to scrape HR policy website
* **static**
  + **node\_modules** - Contains external modules, installed using *npm install*
  + **css/vendor** - Contains our CSS files
  + **img** - Contains images used in app
  + html files - Contains all the templates
  + package.json - Contains all our npm dependencies
  + package-lock.json - Auto generated file from package.json
* **Database Files**
  + databases.zip/data.zip - Contains data used for Neo4j
* **Documentations**
  + Word documents and PowerPoint slides

# How to Run the Program

**Assuming Python Version**: Python 3.7.3

**Steps to Follow – Initialisation:**

1. **Open command prompt and install Virtual Environment:** pip install virtualenv
2. **Creates folder:** virtualenv venv
3. **Git Clone inside venv folder:** https://github.com/Astarcorp/app.git
4. **Activate Virtual Environment:** venv\Scripts\activate
5. **Go inside app folder:** cd app
6. **Install Requirements:** pip install -r requirements.txt
7. **Add Neo4j Graph:** Create Local Graph - Graph Name: Graph, Password: 1234
8. **Turn on Neo4j Server**
9. **Run Flask:** Python run.py
10. **You can access our webpage here:** http://localhost:5000/
11. **Download node.js:** https://nodejs.org/en/ the one on the left side (LTS)
12. **Open a second command prompt**
13. **Go inside app folder:** cd app\static
14. **Install dependencies:** npm install

# How to Store Data into Neo4j

**Neo4j Cypher Language** <https://neo4j.com/docs/cypher-manual/current/>

1. To load CSV file into Neo4j database
   1. Turn on Neo4J and run scrapyapp to scrape HR, may need to update policyId in searchesdata.xlsx and generate a new search.csv

scrapy crawl hrpolicy -o data.csv

* 1. Place users.csv into Neo4j import folder

e.g. C:\Users\Andy\.Neo4jDesktop\neo4jDatabases\database-ae68c79c-4579-433f-8233-a2437cb860e9\installation-4.0.4\import

LOAD CSV WITH HEADERS FROM 'file:///users.csv' AS line

CREATE (:User { username: line.username, email: line.email, dateofbirth: line.dateofbirth, jobtype: line.jobtype, dateofhire: line.dateofhire, password: line.password})

* 1. Place searches.csv into Neo4j import folder

e.g. C:\Users\Andy\.Neo4jDesktop\neo4jDatabases\database-ae68c79c-4579-433f-8233-a2437cb860e9\installation-4.0.4\import

LOAD CSV WITH HEADERS FROM 'file:///searches.csv' AS line

MATCH (node:User {username: line.username})

MATCH (p:Policy) WHERE ID(p) = toInteger(line.policyId)

MERGE (node)-[s:SEARCH {numsearch: toInteger(line.numsearch)}]->(p)

* 1. Run neo4j.py to merge relationships from clustering

Or shortcut

1. Extract database or data zip and replace database or data folder in the respective Neo4j database

e.g. C:\Users\Andy\.Neo4jDesktop\neo4jDatabases\database-ae68c79c-4579-433f-8233-a2437cb860e9\installation-4.0.4\data

1. Creating a node

CREATE (node:Image {image: './static/img/charts.png'})

1. Searching for a node
   1. MATCH (ee:Person) WHERE ee.name = "Emil" RETURN ee
   2. MATCH (ee:Person)-[:KNOWS]-(friends) WHERE ee.name = "Emil" RETURN ee, friends
2. Deleting nodes

MATCH (n) DETACH DELETE n

**Py2Neo Python Codes** <https://py2neo.org/v4/>

1. Creating a node

relationshiptext = 'RELATED'

relatedterm = Node("RelatedTerm", name=relatedtermname, text=text) #creating a node

relationship = Relationship.type(relationshiptext) #changing it into a relationship type

graph.merge(relationship(relatedterm, policy), "Node", "name") #merging nodes with relationship