Didi Yang

Elhadi Elhadi

Modern DB Project

12/17/2020

User Documentation

Link to Presentation & Demo:

https://iowa-my.sharepoint.com/:f:/g/personal/elhadi_uiowa_edu/EmCeaANAZxBKma1eqkOVH9sBG8RJhpdrFwmUL7OD8ycX_w?email=guadalupe-canahuate%40uiowa.edu&e=JcqGPN

Kaggle Datasets: books.csv & ratings.csv

https://www.kaggle.com/hoshi7/goodreads-analysis-and-recommending-books

Instructions to Run Program:

MongoDB:

- Open terminal/command prompt
- Type **mongod** to start the mongoDB server
- Open another terminal, type **mongo** to start the mongoDB client
- Follow MongoDB Data Import section below to import books.csv into mongodb

Neo4j:

- To start Neo4j server, in terminal go to Neo4j directory
- Type "**bin/neo4j start**" to start Neo4j server
- To go http://localhost:7474/ to browse Neo4j database
- Follow Neo4J Data Import section below to import ratings.csv into Neo4j

Python:

- To run Flask server, go to the **UsingPython** directory, and run **python3 app.py**
- In browser go to – **localhost:5000**
- **Explore our application!**

MongoDB Data Import:

- **Step1: To import books.csv table:**

mongoimport --type csv -d finalProjectDb -c Books --headerline --drop --file ~/Downloads/book.csv

This will import the data to finalProjectDB database in the Books collection.

Given that your books.csv file is under your Downloads directory.

- **Step2: Create mongodb index:**
  After creating the database, add text index on Books collections using:

```
db.Books.createIndex({ "original_title": "text"})
```

This will create an index on original title field which can later do a text index on

Neo4J Data Import:

- **Step1: To import ratings.csv table:**

```
auto USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS FROM"file:///ratings.csv" AS row
MERGE (m:User {userId: toInteger(row.user_id)})
WITH m, row
MERGE (n:Book{bookId:toInteger(row.book_id)})
WITH m, row, n MERGE (m)-[r:RATED{rating:tofloat(row.rating)}]->(n)
```

- **Step2: To create a similarity relationship:**

```
MATCH (b1:Book)<-[x:RATED]-(u:User)-[y:RATED]->(b2:Book)
WITH SUM(x.rating * y.rating) AS xyDotproduct,
SQRT(REDUCE(xDot = 0.0, a IN COLLECT(x.rating) | xDot + a^2)) AS xLength,
SQRT(REDUCE(yDot = 0.0, b IN COLLECT(y.rating) | yDot + b^2)) AS yLength, b1, b2
MERGE (b1)-[s:SIMILARITY]-(b2)
SET s.similarity = xyDotproduct / (xLength * yLength)
```