

Neo4J lab: Network of Thrones

LSDM

1 Acknowledgments

The initial inspiration for this lab is a publication by Andrew Beveridge and Jie Shan titled “*Network of Thrones*”. It also borrows ideas from labs proposed by Vincent Leroy, Natasha Tagasovska, and Francieli Zanon Boito.

2 Summary

The authors of the original paper have created a dataset describing all the interactions between the characters of Game of Thrones. The dataset is available here: <https://raw.githubusercontent.com/pupimvictor/NetworkOfThrones/master/stormofwords.csv>.

Each row in the dataset defines a relation between two characters and associate a weight to the relation: higher weights correspond to stronger relationships between those characters.

In this lab, we are going to make our first steps with the Neo4j graph database (<https://neo4j.com/>), a graph database.

Using the relations between Game of Thrones characters as an example, we are going to learn about the features provided by Neo4j. More specifically, we are going to learn how to use the Cypher declarative graph query language to manipulate graph data: <https://neo4j.com/docs/cypher-manual/current/>.

3 Getting Started

The first step to run this lab is to install Neo4j. To this end, we propose two options. We recommend you to use Option 3. Note that we assume that you are using a Linux-enabled system. MacOS users can use the commands directly. Windows users can use the Windows Subsystem for Linux (WSL).

- Option 1: download a Neo4j tarball from the official repository. To this end, access <https://neo4j.com/download-center/> and click at the "Community Server" tab to download a free version.
- Option 2: download a Neo4j tarball using the following link (attention, this is certainly an old version): <https://filesender.renater.fr/?s=download&token=d74ce3bd-90f8-0eb2-12f7-aca3fa929583>. To start Neo4j for Options 1 and 2, run the following commands:

```
tar xzf neo4j-community-3.5.0-unix.tar.gz
cd neo4j-community-3.5.0/
./bin/neo4j start
```

- Option 3 (recommended): use Docker to run Neo4j. To this end, you need to have Docker installed on your machine. If you do not have it yet, you can follow the instructions at <https://docs.docker.com/get-docker/>. Once you have Docker installed, you can run Neo4j using the following command:

```
docker run \
  --name neo4j-lab \
  -p7474:7474 -p7687:7687 \
  -d \
  neo4j
```

Once Neo4j is running, you can open a browser and go to <http://localhost:7474> to interact with it. To login, both username and password are “neo4j” (you will be asked to change the password at the first login).

With this setup you can use the browser for importing data, executing Cypher queries, and getting preview of results in tabular or graph formats. You can also execute the available tutorials to get familiarized with Neo4j and Cypher.

4 Importing Data

Before importing data, we will define a constraint to ensure that our database will never contain more than one node with a given label and one property value. To this end, run the following command:

```
CREATE CONSTRAINT FOR (c:Character) REQUIRE c.name IS UNIQUE;
```

Adding such a constraint will assert the integrity of our schema. It will also improve performance for lookup queries.

In Neo4j, the commands should be entered in the dialog box at the top of the graphical interface.

To import the data in Neo4j, execute the following command:

```
LOAD CSV WITH HEADERS FROM
"https://raw.githubusercontent.com/pupimvictor/NetworkOfThrones/master/stormofswords.csv"
AS row
MERGE (src:Character {name: row.Source})
MERGE (tgt:Character {name: row.Target})
MERGE (src)-[r:INTERACTS]->(tgt)
ON CREATE SET r.weight = toInteger(row.Weight)
```

This command creates nodes with the label **Character**. Each node also has a property **name**. Furthermore, it creates relationships between nodes. Each relationship is of type **INTERACTS**. Furthermore, a property **weight** is set for each relationship.

5 Basic Data Manipulation

In the following, we suggest operations to discover Neo4j and the Cypher query language. To find more information about the language, we refer you to:

- The Cypher manual (you can choose the neo4j version from the menu, the current version as of October 2025 is 5): <https://neo4j.com/docs/cypher-manual/current/introduction/>
- The Cypher reference card: <https://neo4j.com/docs/cypher-refcard/current/>

Basic operations

- To count the total number of characters in the graph, run:

```
> MATCH (c:Character) RETURN count(c)
```

The MATCH clause is used to search for the pattern described in it. Here, we look for all nodes with the label `Character`.

- Since all nodes have the label `Character`, we can even simplify the request. How?
- We can also display the whole graph:

```
> MATCH (c:Character) RETURN c
```

- To find the node corresponding to the character `Sansa`, run:

```
> MATCH (c:Character {name: 'Sansa'}) RETURN c
```

An alternative command is:

```
> MATCH (c:Character) WHERE c.name= 'Sansa' RETURN c
```

Here we use WHERE to add constraints to the described patterns.

- To display all the relations to `Sansa`, run:

```
> MATCH (c:Character {name: 'Sansa'})--(c1) RETURN c,c1
```

- Does it change the results if the property “name: 'Sansa'” is required for `c1` instead of `c`?

- To display the weight of the relationship between `Sansa` and `Arya`

```
MATCH (c {name: 'Sansa'})-[r]-(c1 {name: 'Arya'}) RETURN r.weight
```

Advanced operations

- Find the shortest path between `Arya` and `Ramsay`

- To answer this question, you can use the *shortestpath* function, see <https://neo4j.com/docs/cypher-manual/current/patterns/shortest-paths/>.

- Find the character the most distant to `Arya`

- To answer this question, you can use the *WITH* clause to filter the results of the call to *shortestpath*, see <https://neo4j.com/docs/cypher-manual/current/clauses/with/>.

- How many characters are there with distance at most 4 to `Arya`?

- Information about how to work with relationships can be found here: <https://neo4j.com/docs/cypher-manual/current/relationships/>.

`com/docs/cypher-manual/current/clauses/match/#find-relationships.`

- How many characters are there with a distance of 2 to Arya?
 - *You can include the characters that also have a smaller distance.*
- Find all the paths of size at most 3 from Arya to Orell.

6 To Go Further

If you would like to go further in the analysis of this dataset using Neo4j, we suggest you to follow the tutorial presented here: <https://www.lyonwj.com/2016/06/26/graph-of-thrones-neo4j-social-network-analysis/>.