

Introduction to Neo4j - Lab exercise

Vincent Leroy, Natasha Tagasovska

2017*

1 Introduction

Nowadays, the links between the data are of high importance to enterprises. Many applications need real-time insights into these relationships, and not data alone. Graph databases use graph structures with nodes, edges, and properties to store and retrieve data along with the relationships. In this exercise we will explore graph databases. As in the lectures, the database covered is Neo4j. To get some hands on experience with this NoSQL database, we'll start by installing Neo4j and then use its query language Cypher to explore an interesting dataset.

2 Dataset

Let's explore how Neo4j capabilities can help in analyzing relationships between "Game of Thrones" characters. Andrew Beveridge and Jie Shan published "Network of Thrones" where they analyze a network of character interactions from the novel "A Storm of Swords", the third book in the "A Song of Ice and Fire" saga that is the basis for the "Game of Thrones" TV series. In their paper¹ they detail how they constructed the network of character interactions by using text analysis and entity extraction to find characters mentioned together in the text. The csv file we will use has the following structure: "Source, Target, Weight" where Source and Target represent characters from the books, while the weight quantifies the number of interactions between the them two.

3 Getting started

Start by downloading a Neo4j tarball from <https://neo4j.com/download-center/> (first click at the "Community Server" tab to download a free version). Notice that, as usual, we assume that you are using Linux.

*Updated in 2018 by Francieli Zanon Boito

¹<https://www.maa.org/sites/default/files/pdf/Mathhorizons/NetworkofThrones%20%281%29.pdf>

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

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 follow the option “Write Code” that takes you to some sample datasets to play with.

You can start by getting familiar with Cypher exploring the movie dataset.

4 Analyzing the Graph of Thrones

4.1 Import dataset

The first part of the data for this exercise will be directly imported from the authors’ website. We will convert the adjacency list from the file into a simple data model (Figure 1):

```
(:Character {name})-[:INTERACTS {weight}]->(:Character {name})
```

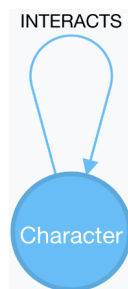


Figure 1: Data model for Network of thrones

Nodes with label “Character” represent characters from the text, and we have a single relationship type “INTERACTS” which connects characters who have interacted in the text. We’ll store the character’s name as *property-name* on the node and the number of interactions between two characters as a *property-weight* on the relationships.

To assure integrity of our schema, we first create a constraint for uniqueness of the character names:

```
CREATE CONSTRAINT ON (c:Character) ASSERT c.name IS UNIQUE;
```

Question: In your opinion, does imposing this constrain improve the performance in a way other than integrity?

Once the constraint is created we can use the Cypher *LOAD CSV* statement to import the data:

```
LOAD CSV WITH HEADERS FROM
"https://www.macalester.edu/~abeverid/data/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 = toInt(row.Weight)
```

4.2 Analyzing the network

With the dataset loaded we can start exploring the graph. Start by visualizing the dataset. Next, write Cypher queries to answer the following questions (you can use the Cypher Reference Card² for quick help on syntax). Notice that for "INTERACTS" relationships the directions are not important.

- How many characters appear in the graph?
- Get summary statistics for the minimum, maximum and average number of characters each character has interacted with.
- Find the shortest path from Arya to Ramsay. Consider the possibility of having few paths of same length. List all of them in the result. You can use the functions "shortestPath" and "allShortestPaths" provided in Cypher.
- What is the longest shortest-path distance between any two characters?
- How many characters are there with interaction distance 4 to Cersei?

4.3 Expanding the graph

Next, we'll enrich our graph by adding information on the family ties between the characters. Download the `family_ties.csv` file from https://francielizanon.github.io/teaching/TP/family_ties.csv and place it in the "import" folder inside the "neo4j-community-3.5.0" folder.

Import the second data set to your Neo4j instance using the following command:

```
LOAD CSV WITH HEADERS FROM "file:///family_ties.csv" AS row
MERGE (src:Character {name: row.character1})
MERGE (tgt:Character {name: row.character2})
MERGE (tgt)-[r:RELATIONSHIP]->(src)
ON CREATE SET r.tie = row.tie
```

With the new dataset included we extend our model to the following (Figure 2):

```
(:Character {name})-[:RELATIONSHIP {tie}]->(:Character {name})
```



Figure 2: Data model for Network of thrones and Family ties

If you visualize the graph, when you hover over links named “RELATIONSHIPS”, you should be able to see the ties. These relationships are directed from parents to their children. Write Cypher queries to answer the following questions:

- Who are Sansa’s parents?
- You can notice that sibling relationships are missing. Try to create them by finding common parents for the ties we have already in the graph.
- Are there any children of incestuous relationships?

²<https://neo4j.com/docs/cypher-refcard/current/>