Skeleton of final report

COMS3
Group 9
Tracking Interconnected Twitter Links
Using Graph Database Neo4j

Lindiwe, Clifford, Thomas

2017

Contents

1 Introduction

1.1 Overview

This purpose of this project is to graph Twitter data so that hidden trends and patterns may be revealed.

1.2 Neo4j

Neo4j is a relatively new database management system, that distinguishes itself from other systems through its ease of use and its speed. Its was initially launched in 2010, however the stable release was only in June 2017. Its fundamental design is to store data as nodes, edges and attributes. Nodes are connected by edges, both can have any number of attributes.

1.3 requirements

To run the code as desired the following programs or packages are required Neo4j v3.2.4
Python v2.7.13
PyCharm 2017.2.3
TextBlob v0.13.0
Tweepy v3.6.0
py2neo v3.1.2
$CipherQueries(GitHub)$
$Twitterneo(GitHub)$

1.4 Running the code

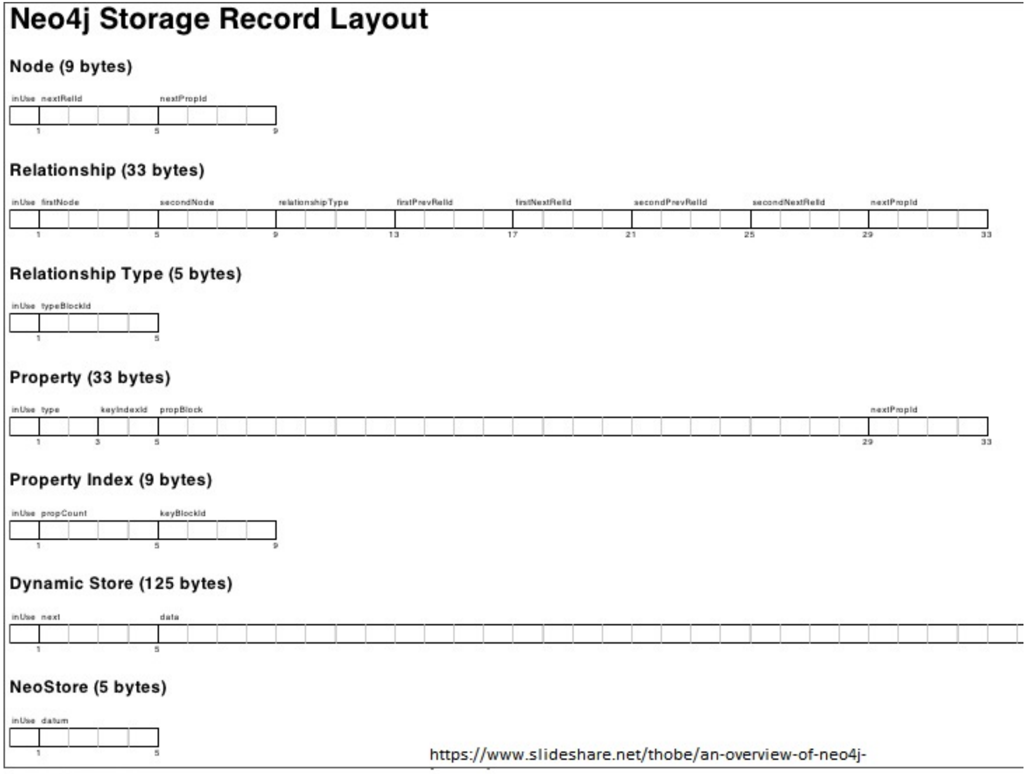
The specifics of running the code as desired can be found on `GitHub in the ReadMe` document

2 Description

We import public data from twitter and store it in a Neo4j database. This database is graphically represented through the Neo4j Browser, which shows nodes and connections, in different colours and sizes to highlight and differentiate various things. Filters can be applied to customise or limit what the graph shows. For example only tweets with a certain phrase or hashtag, "#neo4j", can be shown. This can give an idea about the age, location or gender of people tweeting about Neo4j.

2.1 Design

Figure 1 shows a demo of the final look. Big green nodes are people on twitter, labelled by their twitter handle, they share an edge labelled ”tweeted” to red nodes, labelled ”T” with anything they tweet. Should anyone retweet this tweet, then a ”RT” node is created in purple, sharing ”retweeted” edges with the original tweet and the retweter. Another edge is created from the retweeter to the original tweeter if they are a follower.
Figure 2 shows the data process, from Twitter to the final front-end interface.


Neo4j Storage Record Layout

Node (9 bytes)

Relationship (33 bytes)

Relationship Type (5 bytes)

Property (33 bytes)

Property Index (9 bytes)

Dynamic Store (125 bytes)

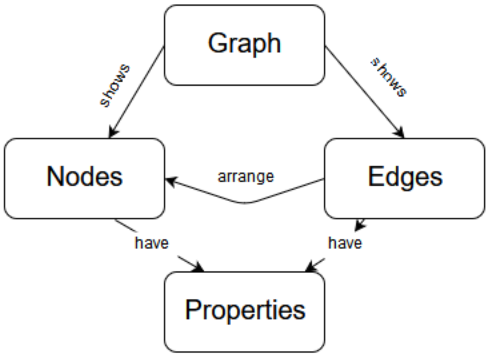
NeoStore (5 bytes)

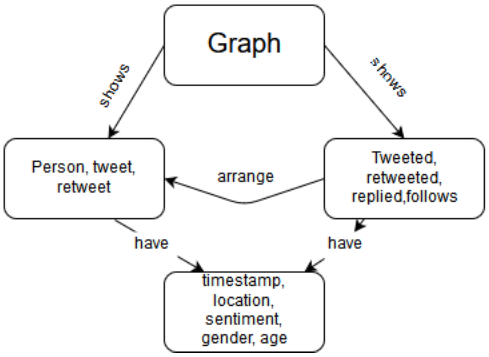
https://www.slideshare.net/thobe/an-overview-of-neo4j-

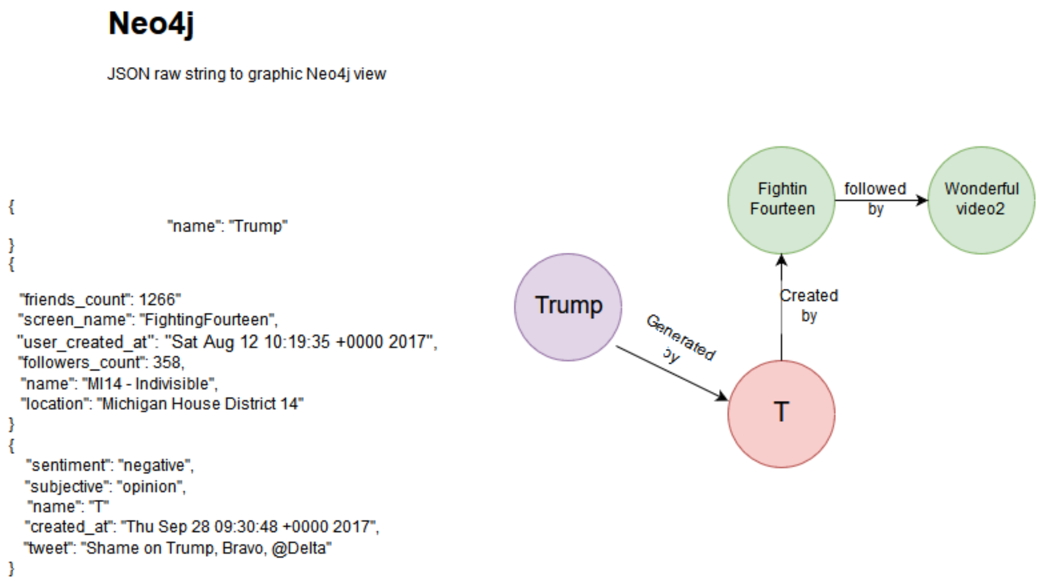
Figure 3 shows the basic idea behind the set up of the database.

Database overview

How Neo4j stores data




Graph
shows
shows
Person, tweet, retweet
arrange
Tweeted, retweeted, replied,follows
have
have
timestamp, location, sentiment, gender, age



Neo4j

JSON raw string to graphic Neo4j view

```
{
                         "name": "Trump"
}
{

  "friends_count": 1266"
  "screen_name": "FightingFourteen",
 "user_created_at": "Sat Aug 12 10:19:35 +0000 2017",
  "followers_count": 358,
  "name": "MI14 - Indivisible",
  "location": "Michigan House District 14"
}
{
   "sentiment": "negative",
   "subjective": "opinion",
    "name": "T"
   "created_at": "Thu Sep 28 09:30:48 +0000 2017",
  "tweet": "Shame on Trump, Bravo, @Delta"
}
```

We discussed including the location of where the tweet was made, however this is problematic for several reasons

- Not all users share their location with twitter
- Different providers share locations differently (3 locations from our testing include, "#GameofThrones" vs "Basin, WY, United States" vs "Kenya")

Discuss sentiment.

2.2 Back-end

simple description on what the back-end does and why

2.3 Front-end

Front end stuff

2.4 application

A tool like this has many applications, certain tweet patters could signal to an account being bot operated, it can be used to find out what a certain age, race, gender, location are tweeting about a certain topic. The first application that pops to mind is targeted advertising, but many more exist.

2.5 looking forward

The first improvement we would like to make would be to host the Neo4j database on a server, so that visual designs and templates can be stored for a uniform look. The next would be a script, either through a website or a program to automate the initial set up in Python and pyCharm, so that the user just needs to select his preferences or enter a topic into a search field and the script does the rest and displays the results in the browser.

3 Members

who did what