Software Requirement Analysis

COMS3
Group 9
Tracking Interconnected Facebook Links
Using Graph Database Neo4j

Lindiwe, Clifford, Thomas

Aug 2017

1 Introduction

1.1 Purpose

Social media plays a vital role in everyday people lives. There is enormous information about individuals and how they relate to one another. This information is useful to individuals, advertisers, politicians and many other organisations. The purpose of our software is to provide means to get links between individuals in social media. Our first focus will be analysing Twitter links using Neo4j database via the website.

1.2 Scope

- Create a database using Neo4j to store our social media data (Clifford)
- Find out about the legal implications of using Twitterdata (Lindiwe)
- Register with Twitter to enable them to give us access to their data (Lindiwe)
- Create a beautiful easy to use web interface (Thomas/Lindiwe)
- Testing of the software (Clifford/Thomas/Lindiwe)

1.2.1 Definitions

- SDLC: Software Development Life Cycle
- UC: use case
- GD: graph database(Neo4j)

2 Overall Description

2.1 Product Perspective

Since this is the first software we are producing, there no other programs to interface with. With this program you will be able to understand your interaction in the Twitter world, to analyse behaviour of tweeters The program will be expected to deliver on the following:

- Show all people using twitter
- Show all tweets of each person
- show retweets, distinguish between followers and non followers
- show replies/mentions

2.2 product functions

- This programwill be designed in Client-Server model
- The front end will be through the Neo4j browser and the back end will be done in Neo4j
- Using Python to handle the data import from twitter to Neo4j
- The program is expected to have a fast response
- The program is expected to do at least three different data analysis

2.3 User characteristics

- The user will have to be willing to share their Twitter activity with our program(this was a concern for Facebook data)
- Twitter is more public than Facebook, the import copies all publicly available data
- The program should be able to handle at least five users without affecting the user experience

2.4 General Constraint

Obtaining permission from Facebook to use their data, else acquire similar data from another source, otherwise switching to use twitter data.

2.5 Assumptions and dependencies

- Facebook will grant us access to their data
- There will be a server where we can run the program

3 Detailed Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces (Thomas)

The Neo4j browser offers easy to use manipulation tools and a nice graphic visualisation, it's the best choice among many similar tools due the compatibility with the Neo4j database.

3.1.2 Hardware Interfaces (Clifford)

The only Hardware interface that will be required for this project is the computer that will be used to program and save the database for this project

3.1.3 Software Interfaces (Clifford)

Neo4j will be used as the database for this project and the browser afforded by Neo4j will be the first choice if feasible as the user interface. At the moment Java or Python are the front runners to be used for the back-end programming. Java because Neo4j supports native Java and Python because many data science libraries are written in Python

3.2 Function Requirements

3.2.1 Back-end requirements

UC1: Show people as big nodes

UC2: Bridge all tweets to their creators

- Primary actor: The tweeter
- Track the time of the actions, to compare to given period

UC3: Bridge all retweets of a comment

- Primary actor: The retweeter
- Secondary actor: The original tweeter
- Track the time of the actions, to compare to given period
- Check if the retweeter is a follower of the original tweeter

UC4: Replies and Mentions

- Primary actor: The tweeter
- Secondary actor: tweet/person mentioned/replied to

3.2.2 Front-end requirements

Represent different actions by different connects between nodes

UC1: People
Biggest nodes to show people(green)

UC2: Tweets
Smaller nodes(red) bridges(brown) to people

UC3: Retweets
Nodes(purple) for a retweet, bridge(pink) to the tweet, bridge(blue) to retweeter, and same colour bridge to original tweeter, if the retweeter is a follower

UC4: Mentions
Little nodes(light blue), bridges(light blue) to tweet/person mentions/replied to

3.3 Performance Requirements

3.3.1 Intuitiveness

The database and representation should be as logical as possible, and as easy to use, maintain and modify as manageable. A simple design will decrease database queries to improve response time

3.3.2 Speed

The speed of execution is important. Neo4j is faster than MySql Relational Database Management System. Expected response time of 15 seconds 90% of the time.

3.3.3 Agility

The system is to be naturally adaptive, with the flexibility to add and remove data from the database

3.4 Software System Attributes

3.4.1 Reliability

- Elimination of duplicated data
- Ease of use
- Consistency

3.4.2 Security

~~As the data is personal it should be secured, so no other can view it, and anonymised.~~
Twitter data is publicly available, anonymity isn't an issue anymore

3.4.3 Maintainability

Maintenance should be possible and easy and allow for the implementation of new functions

3.5 Design Constraints

3.5.1 Hardware

Our software uses Neo4j database,Neo4js storage is organized in record-based files per data structure nodes, relationships, properties, labels, and so on. Each node and relationship record block is directly addressable by its id.Nodes occupy 15B of space, relationships occupy 31B of space and properties occupy 41B of space.The low level cache requirements are the same as for the disk space.The system will be integrated with a website. To use recommendation system,user should enter from a personal computer,mobile device with internet connection.

3.5.2 User direct interaction constraints

This project is expected to use the graph database to manage large samples of database from Facebook of nodes (members) and links (connections between nodes.The softwaare will be pretty static software,preventing users from enacting personalized websites layout.It will only use the data from facebook.This does not allow much room for creative customizability of a user's profile.