Your name
Date

McKesson Final Project Deep Azure

Topic: Case Study of Social Network Analysis using MS SQL Server & R

Problem/: To measure the influence of twitter users by studying patterns in retweets of a Presidential candidate using RESTFul APIs, R for visualization graphics and SQL Server 2016 for ingest and storage of tweets.

Big Data Set:
Streaming data from Twitter API focused on tweets and retweets from Bernie Sanders (@SenSanders and @BernieSanders) and which twitter users retweeters follow. This user has approximately 2 million twitter followers. The data is quite challenging because twitter's default API severely reduces how quickly you can capture data from it. However, I did manage to collect 80% of the 250,000 retweets of the Senator's recent 150 tweets and follower relationships of 5 million twitter users and loaded data into a SQL Server database which contained 1 GB of tweets.

Hardware/OS:

- Intel 2-core i5-5200 CPU 2.2Ghz, 8 GB RAM, 30Mbit/sec fiber Internet connection, 64 bit Windows OS

Software:

Technology/tools	Description
Twitter REST API	Twitter's Public REST API to its data
Revolution R 3.x	R used for creating data set, displaying graphs
HTTR library	Library to request data from REST APIs and to authenticate access from OAuth APIs
SQL Server 2016	SQL Server used for storage & manipulation of data

Overview of steps:

1. Install SQL Server 2016 including its R Services (In-Database) component.
2. Attach the sample database, or create a new empty one (by running ddl.sql) in Microsoft SQL Server Management Studio or whatever GUI or command line tool you like.
3. Install the HTTR package for R and possible two subsidiary R packages
4. Learn the Twitter API. Get Twitter app keys if you don't have any. Place keys in Twitter_Account_List.r
5. Run Get_Retweets.r to generate data of tweets and retweets
6. Run Get_Friends.r and Get_Followers.r to generate data of followers of retweeters
7. Run various graphical data summaries via R script Display_Analysis.r.

Summary:
I considered the graph-oriented DBMS Neo4J as the data store, but after looking at it's simple indexing technology I could not see a data retrieval architecture designed to increase network data retrieval by an order of magnitude vs the architectures employed by modern mainstream DBMSs.

The much larger issue was finding a way around the default Twitter API's throttle on capturing significant amount of **useful related** data. In particular, according to all public sources, it is impossible to collect a non-trivial amount of retweet data. Due to the importance of this info to answering the problem statement after days I figured out an undocumented loophole.

YouTube URLs
Short: https://youtu.be/shorthere

Long: https://youtu.be/longhere

References: Put your URLs here