Certificate

This is to certify that the project entitled **"Twitter Data Analytics on Women Education and Employment"** is being submitted at IGDTUW, Delhi for the award of **Bachelor of Technology** in **Computer Science & Engineering** degree. It contains the record of bonafide work carried out by **NAME OF STUDENT** under my supervision and guidance. It is further certified that the work presented here has reached the standard of B.Tech and to the best of my knowledge has not been submitted anywhere else for the award of any other degree or diploma.

Prof. Ela Kumar
(Dean of Academic Welfare)
Computer Science Department
IGDTUW, Delhi

Date:
Place: New Delhi

Student's Declaration

We, Group Id-1 hereby declared that the work being presented in this report entitled "**Twitter Data Analytics on Women Education and Employment"** submitted to INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN, New Delhi, for the award of degree of B.Tech CSE is an authentic record of our work carried out under the guidance of **Prof. Ela Kumar** (Dean of Academic Affairs), IGDTUW, Delhi.

The matter embodied in this report has not been submitted by us for the award of any other degree.

Dated:

GROUP MEMBERS :

Sonal	**06713502711**
Manali Verma	**06813502711**
Sonam Pal	**07013502711**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Under the Guidance of

Prof. Ela Kumar
(Dean of Academic Affairs)
Computer Science Department
IGDTUW, Delhi

Acknowledgement

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. First and foremost, we would like to express infinite thanks to our Mentor **Prof. Ela Kumar (Dean, Academic Affairs), IGDTUW, Delhi** for provision of expertise, and technical support in the implementation. Without their superior knowledge and experience, the Project would like in quality of outcomes, and thus their support has been essential. We would like to express our sincere thanks towards volunteer researchers who devoted their time and knowledge in the implementation of this project.

Nevertheless, we express our gratitude toward our families and colleagues for their kind co-operation and encouragement which help us in completion of this project.

Date:
Place: New Delhi

ABSTRACT

This Document is submitted for the purpose of fulfillment of Degree of B. Tech in Computers Science and Engineering. The objective of this project is to develop an **Analysis on Twitter data on women education and employment**. In this project, our aim is to provide an enhancement in the women empowerment by performing extraction, transformation and standardization of the tweets from the Twitter. Twitter makes it easy to engage users and communicate directly with them, and in turn, users can provide word-of-mouth marketing for companies by discussing the products. The tweets are converted from JSON format into excel sheet. Nodexl is a free and open-source network analysis and visualization software package for Microsoft Excel 2007/2010. It is a popular package similar to other network visualization tools such as Pajek, UCINet, and Gephi. NodeXL imports UCINet and GraphML files, as well as Excel spreadsheets containing edge lists or adjacency matrices, into NodeXL workbooks. NodeXL also allows for quick collection of social media data via a set of import tools which can collect network data. NodeXL contains a library of commonly used graph metrics: centrality, clustering coefficient, diameter. NodeXL differentiates between directed and undirected networks. NodeXL implements a variety of community detection algorithms to allow the user to automatically discover clusters in their social networks.

This project is to provide an estimate and statistics of the education and employment of women all across the world especially in India. This project implements all the concepts learned in different subjects such as DBMS and ERP and provides a valuable experience to gain a sound understanding as well as utilize creativity to develop this solution. If successful, this project has the potential to be implemented within the organization as well.

CONTENTS

CHAPTER 1

Introduction

Overview

The term Big Data is used almost anywhere these days. Big Data can be seen in the social media, finance and business where enormous amount of stock exchange, banking, online and onsite purchasing data flows through computerized systems every day and are then captured and stored for inventory monitoring, customer behavior, human relationship and market behavior. It can also be seen in the life sciences where big sets of data such as genome sequencing, clinical data and patient data are analyzed and used to advance breakthroughs in science in research.

Project Background

Our aim for developing this project is to provide an estimate and statistics of the education and employment of women all across the world especially in India. In India, the women literacy rate is very low and the educated women are paid less than women. Worth of the work done or services rendered by women has not been recognized. We by this project are trying to give a hand in support for women empowerment. Since Social media has gained immense popularity with marketing teams, and Twitter is an effective tool for a company to get people excited about its products we are using Twitter's Data for our analytics. Twitter makes it easy to engage users and communicate directly with them, and in turn, users can provide word-of-mouth marketing for companies by discussing the products. Given limited resources, and knowing we may not be able to talk to everyone we want to target directly, marketing departments can be more efficient by being selective about whom we reach out to.

After extracting the number of tweets, we can perform analytics by using a query based interface tool and then can produce results

Thesis Outline

This thesis report consists of seven chapters, References and Appendix. Rest of the thesis is organized as following:

Chapter 2- Various techniques and methodologies are using for extracting the data from social media. Here, we discuss the recent extraction details that are used for collecting, transforming and visualizing the data. Different organizations have different techniques to update the big data from social media.

Chapter 3-This chapter discusses the problem in extracting the huge amount of data from social media with seed words. We discuss the problem in transforming and standardize the data of the women' education and employment taken directly from twitter. This chapter also shows the position and implications of women in India.

Chapter 4- The proposed approach algorithm is described in this chapter followed by the system Requirements, Specifications, hardware architecture, software architecture, System integration and layered structure of the system. This chapter gives the details about the hardware components used for the system along with their specifications.

Chapter 5-The software implementation and schematic of the system and the integration of both hardware and software is explained in this section. The working of this project is explained with the help of a sequence diagram and the flow of software code is also explained in the form of a flowchart.

Chapter 6- Finally thesis ends with the conclusion and contributions of the system. The limitations of the system are also given in this chapter and at the end the scope for future work is given.

CHAPTER 2

Literature Review

To develop an **Analysis on Twitter data on women education and employment**. Our aim is to provide an enhancement in the women empowerment by performing extraction, transformation and standardization of the tweets from the Twitter.

2.1 General

Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but unregistered users can only read them. Users access Twitter through the website interface, SMS, or mobile device app. Twitter Inc. is based in San Francisco and has more than 25 offices around the world.

In order to retrieve tweets from Twitter in real time, querying Twitter data in a traditional RDBMS is inconvenient, since the Twitter Streaming API outputs tweets in a JSON format which can be arbitrarily complex. A Python library called Tweepy is used to connect to Twitter Streaming API and downloading the data. Tweepy is a Python 2.6 and 2.7 library for accessing Twitter. It provides access to all twitter RESTful API methods, including reading and posting of tweets. Tweepy supports OAuth authentication, as BasicAuth is no longer supported by the Twitter API.

NodeXL is a free and open-source network analysis and visualization software package for Microsoft Excel 2007/2010. It is a popular package similar to other network visualization tools such as Pajek, UCINet, and Gephi.

NodeXL is a set of prebuilt class libraries using a custom Windows Presentation Foundation control. Additional .NET assemblies can be developed as "plug-ins" to import data from outside data providers. Currently-implemented data providers for NodeXL include Facebook, Twitter, Wikipedia (theMediaWiki understructure), web hyperlinks, Microsoft Exchange Server.

NodeXL is intended for users with little or no programming experience to allow them to collect, analyze, and visualize a variety of networks. NodeXL integrates into Microsoft Excel 2007 and 2010 and opens as a workbook with a variety of worksheets containing the elements of a graph structure such as edges and nodes. NodeXL can also import a variety of graph formats such as edgelists, adjacency matrices, GraphML, UCINet .dl, and Pajek .net.

NodeXL imports UCINet and GraphML files, as well as Excel spreadsheets containing edge lists or adjacency matrices, into NodeXL workbooks. NodeXL also allows for quick collection of social media data via a set of import tools which can collect network data from e-mail, Twitter, YouTube, and Flickr. NodeXL requests the user's permission before collecting any personal data and focuses on the collection of publicly available data, such as Twitter statuses and follows relationships for users who have made their accounts public. These features allow NodeXL users to instantly get working on relevant social media data and integrate aspects of social media data collection and analysis into one tool.

NodeXL workbooks contain four worksheets: Edges, Vertices, Groups, and Overall Metrics. The relevant data about entities in the graph and relationships between them are located in the appropriate worksheet in row format. For example, the edges worksheet contains a minimum of two columns, and each row has a minimum of two elements corresponding to the two vertices that make up an edge in the graph. Graph metrics and edge and vertex visual properties appear as additional columns in the respective worksheets. This representation allows the user to leverage the Excel spreadsheet to quickly edit existing node properties and to generate new ones, for instance by applying Excel formulas to existing columns.

NodeXL contains a library of commonly used graph metrics: centrality, clustering coefficient, diameter. NodeXL differentiates between directed and undirected networks. NodeXL implements a variety of community detection algorithms to allow the user to automatically discover clusters in their social networks.

NodeXL generates an interactive canvas for visualizing graphs. The project allows users to pick from several well-known Force-directed graph drawing layout algorithms such as Fruchterman-Reingold and Harel-Koren. NodeXL allows the user to multi-select, drag and drop nodes on the canvas and to manually edit their visual properties (size, color, and opacity). In addition, NodeXL allows users to map the visual properties of nodes and edges to metrics it calculates, and in general to any column in the edges and vertices worksheet.

The software architecture comprises three extendable layers:

Data Import Features: NodeXL stores data in a pre-defined Excel template that contains the information needed for generating network charts. Data can be imported from existing Pajek files,other spreadsheets, comma separated value (CSV) files, or incidence matrices. NodeXL also extracts networks from a small but extensible set of data sources that includes email stored in the Windows Search Index and the Twitter micro-blogging network. Email reply-to information from personal e-mail messages is extracted from the Microsoft Windows Desktop Search index. Data can also be imported about which user subscribes to one another's updates in Twitter, a micro-blogging social network system. NodeXL has a modular architecture that allows for the integration of new components to extract and import network data from additional resources, services, and applications. The open source access to the NodeXL code allows for a community of programmers to extend the code and provide interfaces to data repositories, analysis libraries, and layout methods. Spreadsheets can then be used in a uniform way to exchange network data sets by a wide community of users.

Network Analysis Module: NodeXL represents a network in the form of edge lists, i.e., pairs of vertices which are also referred to as nodes. Each vertex is a representation of an entity in the network. Each edge, or link, connecting two vertices is a representation of a relationship that exists between them. This relationship may be directed or not. Some relationships are bidirectional (like marriage); others can be uni-directional (like lending money). the presence of a relationship. These lists can be extended with additional columns that can contain data about the relationship. NodeXL includes a number of software routines for calculating statistics about

individual vertices including in-degree, out-degree, clustering coefficient, and closeness, betweenness, and eigenvector centrality. Additional analyses features can beintegrated by advanced users. The results of the network metric calculations are added to the spreadsheet as additional columns that can be further combined and reused in Excel formula during analysis and visualization. Spreadsheet features like data sorting, calculated formulae, and filters can be applied to network data sets directly.

Graph Layout Engine: NodeXL provides a canvas for displaying and manipulating network charts and data. Users can apply a range of controls to convert an edge list into a useful node-link chart. These include display options that specify the appearance of individual edges and nodes as well as the overall layout of the network. The lines between nodes that represent edges can have different thickness, color, and level of transparency depending on the attributes of the data or parameters specified by the user. Similarly, each node representing a vertex can be set to have a different location, size, color, transparency, or shape. Optionally, the user can specify images to replace the node shapes.

2.2.1 SYSTEM DESCRIPTION AND WORKFLOW

The core of NodeXL is a special Excel workbook template that structures data for network analysis and visualization. Six main worksheets currently form the template. There are worksheets for “Edges”, “Vertices”, and “Images” in addition to worksheets for “Clusters,” mappings of nodes to clusters (“Cluster Vertices”), and a global overview of the network’s metrics (“Overall Metrics”). NodeXL workflow typically moves from data import through processing, calculation and refinement before resulting in a network graph that tells a useful story. These steps include:

Step 1: Import data. Network data can be imported from one or more network data sources. Users may have data in files, e.g. in text format, separated by delimiters, or stored in relational databases. Wherever the data originates, it is entered into the NodeXL template in the “Edges” worksheet in the form of pairs of names, along with any additional attributes about relationship between them. Multiple edge lists can be stored in the same spreadsheet, expressing different relationships among a set of nodes or the same relationships at different times. Relationships can

be annotated with multiple additional columns, which can be used to set values for display attributes. Data about the "strength" of the relationship can be included, or edges can be annotated with the time slice or date range in which they occurred, allowing a single dataset to contain edges over multiple time periods.

Step 2: Clean the data. This typically involves eliminating duplicate edges when appropriate. Data sets can often be noisy and contain redundant data. In some cases network measures cannot be calculated correctly if multiple edges between the same pair of entities exist in a single data set. In these cases redundant edges may be aggregated into a single edge with a weighting that reflects the number of original instances.

Step 3: Calculate graph metrics-A range of measurements exist that capture the size and internal connectivity of a network as well as attributes of each node. NodeXL supports a minimal set of the most crucial network measures for individual nodes: in- and outdegree, clustering coefficient, and betweenness, closeness, and eigenvector centrality. This operation also populates the Vertex worksheet with a unique list of nodes and their network measures. In some cases multiple edges between nodes are part of what is interesting in a data set and should be retained despite the fact that some metrics will be inaccurate and should be ignored. NodeXL marks these network metrics with the Excel "Bad" format if asked to calculate some metrics with duplicate edges in the data set.

Step 4: Create clusters-Network nodes may share a variety of attributes. It is often useful to group and analyze them together. NodeXL has a clustering algorithm and allows users to create clusters and map nodes to them by editing the Clusters and Cluster Vertices worksheets. Each cluster can have its own display attributes with a distinctive shape, color, size, transparency or image. Users can toggle the display of clusters so that the display features for each node is replaced by the display features for its cluster (if any).

Step 5: Create sub-graph images-Whole graph images are often too large or dense to reveal details about individual nodes or clusters. Sub-graph images produce a local network that centers on each node at a time and encompasses the nodes to which it is immediately connected. These

extracted images can be useful representations of the range of variation in the local network structures of the population in the network.

Step 6: Prepare edge lists- Nodes can have a "Layout Order" value that governs the presentation of nodes in the graph display. Nodes and edges have attributes that can be used to order the data, for example, ordering nodes by their date of first appearance or rate of connection to other nodes. The value found in "Layout Order" governs the order in which nodes are laid out in the whole graph visualization.

Step 7: Expand worksheet with graphing attributes-Columns can be auto-filled to map data to display attributes. Graphical attributes of nodes and edges, their shape, color, opacity, size, label, and tooltip can be altered to convey additional information in the network visualization. Images listed in the "Images" worksheet can replace the shapes used to represent nodes. Users can insert additional numerical attributes about each node in adjacent columns. These attributes can be automatically scaled to display characteristics. For example, each node may have data about the income of the person it represents or the number of employees in an enterprise which could be mapped to the size, shape, or color of a node. Once set, these mappings apply to all networks created from that point on with NodeXL until reset "sticky" layout feature simplifies the creation of multiple networks while maintaining consistency of display mappings.

Step 8: Show graph-This opens or hides the graphical display pane in which NodeXL will render a visualization of the network. At each stage of the NodeXL workflow, the toolkit provides a number of options; the workflow is not rigidly prescribed. The user can iteratively refine any stage of the analysis and visualization. This may involve operations like: Read workbook. Load the current state of the network as stored in the spreadsheet and render it according to the selected layout. Adjust layout. Select among a number of automated layout options that govern where each node in the network will be located. Layouts include a force-directed Fruchterman-Reingold layout [10], that attempts to dynamically find a layout that clusters tightly connected nodes near one another as well as simple geometric layouts like circles or grids. Apply dynamic filters. Selectively hide edges and nodes, depending on the attributes of

the network. For example, a filter could hide all but the most connected nodes, or show only the edges that are ‘stronger’ than a selected threshold. Filtering may involve “trimming” parts of the network and then recalculating network metrics and layout based on the remaining population of nodes and edges. Re-render the graph. Redraw the network based on remaining nodes and edges and their changed display attributes. Finalize the network analysis

CHAPTER 3

Problem Identification

There are several technologies used for extracting the data from social media. Also several machine learning, open source, freeware and licensed software are available in the market. We have to select one among this wide range of components which appropriately suits the design and helps in achieving the goals and objectives set for this project. This chapter gives details about the problem arise in extracting the big data from social media directly and gives inefficient results.

3.1 Background

Analyzing information involves examining it in ways that reveal the relationships, patterns, trends, etc. that can be found within it. That may mean subjecting it to statistical operations that can tell you not only what kinds of relationships seem to exist among variables, but also to what level you can trust the answers you're getting. It may mean comparing your information to that from other groups (a control or comparison group, statewide figures, etc.), to help draw some conclusions from the data. The point, in terms of your evaluation, is to get an accurate assessment in order to better understand your work and its effects on those you're concerned with, or in order to better understand the overall situation.

There are two kinds of data you're apt to be working with, although not all evaluations will necessarily include both. **Quantitative data** refer to the information that is collected as, or can be translated into, numbers, which can then be displayed and analyzed mathematically. **Qualitative data** are collected as descriptions, anecdotes, opinions, quotes, interpretations, etc., and are generally either not able to be reduced to numbers, or are considered more valuable or informative if left as narratives. As you might expect, quantitative and qualitative information needs to be analyzed differently.

The problem of taking the project on women's education and employment not only to show the impact on one side but also emphasis to each organization. There are some questions arise when we extract and transform the data sets on women's education and employment. To what extent legislative measures have been able to raise the status of women in India? Are women now feel empowered in the sense that they are being equally treated by men in all spheres of life and are able to express one's true feminine urges and energies? What initiatives have proven successful in helping women and girls transition from school/training to work? What about institutional efforts on gender equality and employment? What types of non-formal education and training do women and girls participate in? What about women employment after marriage? These are the important questions to be investigated with regard to women's empowerment in India.

3.2 Problem in Data Extraction, Transformation and Standardization

The problems start right away during data acquisition, when the great volume of data flooded beyond the memory and requires us to make decisions. This continuously creates an problem to the data analyst, business intelligence manager and data scientist to take strategic decision to accommodate the petabytes of data. Currently there is an ad hoc manner, about what data to keep and what to discard, and how to store what we keep reliably with the right metadata. Another problem arises when the extracted data is not in the format that we use in our system. Then we manually manage the structured format of data. Naive user approximately analyzes the structured and standardized the data. But, if we have a petabytes of data in our data-sets, then there would be a problem to standardize the data. As we know that the data in the blogs, tweets, facebook, google hangout are not so much structured and have multiple format of texts, images audio and video for display and storage. So, we need to transform the data according to the user perspective into the standardized and structured way. The data should be platform independent and open source to everyone. This becomes a great challenge to manage the huge amount of data in the memory for collecting and transforming to conclude the effective results. Data should be integrated and credible to every data analyst for performing efficient analysis on seed words from social media. Today, mostly the data is directly extracted from the social media and stored for analysis, which gives inappropriate conclusion and percentage of growth. We have the opportunity and the challenge to facilitate the seed words to justify the queries of analyst. Data

analysis, organization, retrieval, and modeling are other foundational challenges. Algorithm design to get effectively retrieval of data is another increases the complexity at social media.

Even after data cleaning and error correction, some incompleteness and some errors in data are likely to remain. This incompleteness and these errors must be managed during data analysis.

CHAPTER 4

Proposed Approach

This chapter describes the proposed approach and conceptual approach of an application regarding social media. The approach related with the concepts of extracting the tweets from twitter for data analysts. The conceptual design of slow growth in extracting the huge amount of data is a concern for organizations. It comprises system design and methodology related with the requirement. It is an approach to clarify the techniques to extract the data effectively from social media.

4.1 Motivation

Basically, we should take a stand to empower the women for education and employment. This should help the country to visualize the perspective related with women and girls. The subject of empowerment of women has becoming a burning issue all over the world including India since last few decades. They have demanded equality with men in matters of education, employment, inheritance, marriage, and politics and recently in the field of religion also to serve as cleric (in Hinduism and Islam). Women want to have for themselves the same strategies of change which men-folk have had over the centuries such as equal pay for equal work. Their quest for equality has given birth to the Big Data Analytics from Social Media on Women's education and employment.

4.2 Proposed Solution

We designed the algorithm for keywords cleaning and eliminating the repeated tweets from a twitter user by the following mechanism:

1. Eliminating the non- English tweets.
2. Counting the retweets from the data sets, this refers to tweets from somebody and shared by any other person because this may conclude the enrichment in women's education and employment.
3. Elimination of URLs from the tweets.

4. Eliminating the hashtag (#) from the tweets because great volumes of data are attached with seed words, which tend to magnify the vocabulary size inadvertently.
5. Tackle the problem of repetitions of unnecessary letters in the tweets eg. Wwwooommmeeennn to women.
6. Eliminating the format of @username from the tweets.
7. Condense each 2 repetitive letters into single letter if the number of such sequences is over a threshold. For example, ookk will be reduced to ok, while book remains intact.
8. Removing all the punctuation marks and stopword.
9. Eliminating the additional English grammar sentences.
10. Eliminating the non seed words. For example, Womania, educationaism, emplymentnews and others.
11. Eliminating the name of the same twitterer who shared his/her tweets with others because they unnecessarily increase the volume of data.

4.2.1 Assumption

The algorithm of Fruchterman and Reingold added "even vertex distribution" to the earlier two criteria and treats vertices in the graph as "atomic particles or celestial bodies, exerting attractive and repulsive forces from one another." The attractive and repulsive forces are redefined to

$$fa(d) = d2/k \quad , \quad fr(d) = -k2/d$$

in terms of the distance d between two vertices and the optimal distance between vertices k defined as

$$k = C (\text{area} / \text{number of vertices})^{1/2}$$

The algorithm of Fruchterman and Reingold adds the notion of "temperature" which could be used as follows: "the temperature could start at an initial value (say one tenth the width of the frame) and decay to 0 in an inverse linear fashion." The temperature controls the displacement of vertices so that as the layout becomes better, the adjustments become smaller. The use of temperature here is a special case of a general technique called simulated annealing.

Pseudo code :

```
while temp > 0.5 and passes < maxIterations (500)
//calculate repulsive forces between each node
for v = 0 to numberOfNodes
for u = v+1 to numberOfNodes
```

calculate the distance vector between the positions of v and u
calculate a displacement displaceVec = (distVec/|distVec|) * repulsion(|distVec|)
add displaceVec vector to v's displacement vector
subtract displaceVec from u's displacement vector
end
end
//calculate attractive forces
for e = 0 to numberOfEdges
get the nodes attached to the edge (v and u)
calculate the distance vector between the positions of v and u
calculate a displacement displaceVec = (distVec/|distVec|) * attraction(|distVec|)
subtract displaceVec vector from v's displacement vector
add displaceVec to u's displacement vector
end

calculate each nodes's displacement, but limit max displacement to temp

//decrease temperature parameter acording to cooling schedule

if "Show updates" is true and this is an Nth pass, update the layout on screen
at the end, go over all the nodes to find the max and min of the coords, rescale all coords so that network will fill the display end while

//repulsion function
repulsion(distance) = (optDist^2*arcWeight^2) / layoutdistance

//attraction function attraction(distance) =
layoutDistance^2 /(optDist*arcWeight))

//cooling function
coolTemp(temp) = querys the CoolingSchedule for the the appropriate value for the iteration, as set by the user with the cooling function.

//optimal distance optimalDistance = layout parameter in apply layout settings

Asymptotic Notation and Complexity Calculation:

Each iteration the basic algorithm computes O(|E|) attractive forces and O(|V |2) repulsive forces. To reduce the quadratic complexity of the repulsive forces, Fruchterman and Reingold suggest using a grid variant of their basic algorithm, where the repulsive forces between distant vertices are ignored. For sparse graphs, and with uniform distribution of the vertices, this method allows

a O(|V |) time approximation to the repulsive forces calculation. This approach can be though of as a special case of the multi-pole technique introduced in n-body simulations

4.2.2 Algorithm

Multi-level Graph algorithms are designed to make all the edges about the same length and to minimize line crossings, which can make for a more aesthetically pleasing and readable graph.

Notation and Definition:

Let G = G(V,E) be an undirected graph of vertices V , with edges E and which we will assume is connected. For any vertex v let Γv be the neighbourhood of, or set of vertices adjacent to, v, i.e., Γv = {u ∈ V : (u, v) ∈ E}. We use the |.| operator to denote the size of a set so that |V | is the number of vertices in the graph and |Γv| is the number of vertices adjacent to v (the degree of v). We also use |.| to denote the weight of a vertex; since weighted vertices in the coarsened graphs represent sets of vertices from the original graph, the weight of a coarsened vertex is just equivalent to the number of original vertices in the set it represents. We then use ||.|| to denote Euclidean distance in either 2D or 3D.

Algorithm:

```
{ initialisation }
function fr(x, w) := begin return −Cwk2/x end
function fa(x) := begin return x2/k end
t := t0;
Posn := NewPosn;
while (converged = 1) begin
        converged := 1;
        for v ∈ V begin
           OldPosn[v] = NewPosn[v]
        end
        for v ∈ V begin
           { initialise Θ, the vector of displacements of v }
           Θ := 0;
           { calculate (global) repulsive forces }
        for u ∈ V , u = v begin
```

```
            Δ := Posn[u] − Posn[v];
            Θ := Θ + (Δ/||Δ||) · fr(||Δ||, |u|);
        end
        { calculate (local) attractive/spring forces }
        for u ∈ Γv begin
            Δ := Posn[u] − Posn[v];
            Θ := Θ + (Δ/||Δ||) · fa(||Δ||);
            end
        reposition v }
    NewPosn[v] = NewPosn[v] + (Θ/||Θ||) · min(t, ||Θ||);
    Δ := NewPosn[v] − OldPosn[v];
    if (||Δ|| > k·tol) converged := 0;
end
{ reduce the temperature to reduce the maximum movement }
t := cool(t);
end
```

Complexity Calculation:

If we set R to be the maximum distance over which repulsive forces will act we can then modify the algorithm by changing the global force calculation to:

```
function fr(x, w) :=
    begin
        if (x ≤ R) return −Cwk2/x;
        else return 0.0;
    end
```

In itself this modification will do little or nothing to speed up the calculation as the complexity is still O(|Vl|. However Fruchterman & Reingold, [7], showed that if the domain is divided up into regular square cells (or cube shaped cells in 3D) of size R2 (or R3 in 3D) then each vertex will only be affected by repulsive forces from vertices in its own and adjacent cells (including those diagonally adjacent). To implement this efficiently we simply visit every vertex at the start of each outer loop and add each to a linked list of vertices for the cell to which it belongs. Repulsive

forces can then be calculated for each vertex by using the linked lists of their own and adjacent cells.

4.3 Comparison of NodeXL and HootSuite

In this section we compare our analysis with HootSuite.

HootSuite has confusing interface because there is so much activity taking place within any one tab. It has longer learning curve. Hootsuite can incorporate analytics from Twitter, Facebook, LinkedIn, Google Analytics and its own Ow.ly shortened URL metrics. Costs can mount up. Although the Pro plan costs just $9.99 per month, adding more than one team member increases the cost — adding a second team member doubles the price, to almost $20 per month.

CHAPTER 5

IMPLEMENTATION

The people are responsible for extracting and transforming the huge amount of data to analyze various types of queries, so it starts by thinking about its structure. This describes the implementation details of the tweets from the twitter. What types of tweets are extracted from the twitter? What should each one do while transforming the data? And how should analyze the keywords of the twitter? Once code actually exists, they ask more questions. What does this class look like? What other classes is it related to? What's the sequence of calls from this method?
All of these questions lend themselves to visual answers. In every case, creating diagrams that show what's going on can be the clearest path to understanding.

5.1 Getting the Twitter API Keys

1. Create account on Twitter Developer : ***https://dev.twitter.com/apps*** and log in with your twitter credentials.

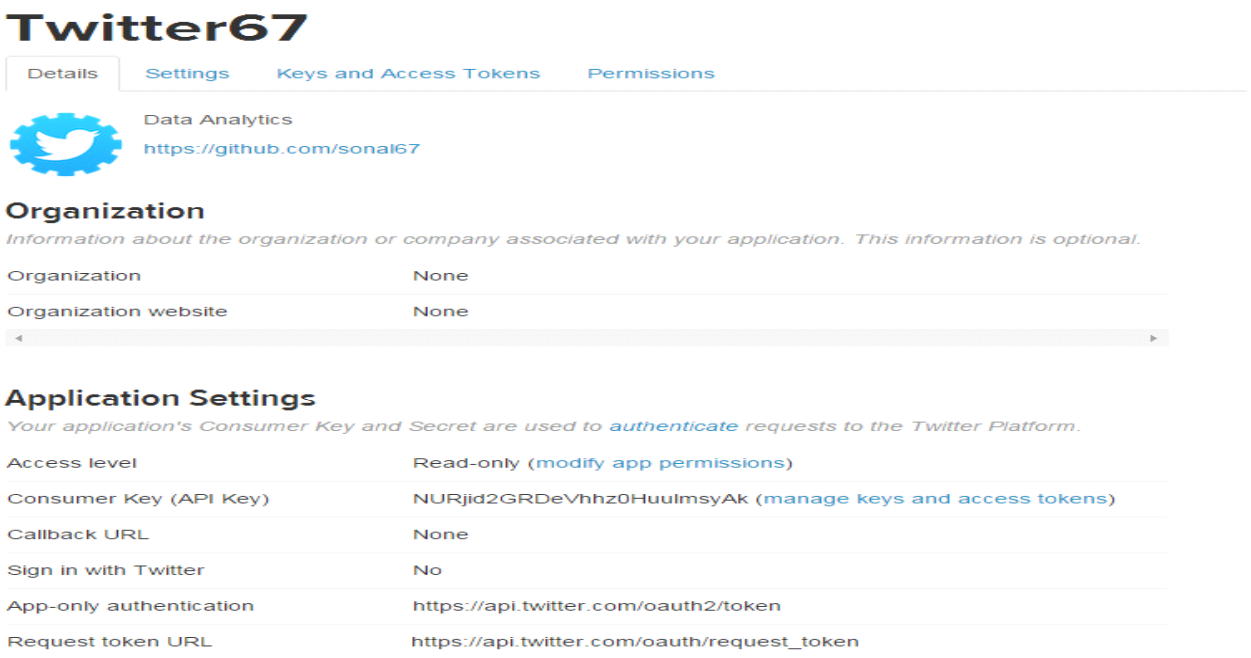
2. Click "Create New App"

3. Fill out the form, agree to the terms, and click "Create your Twitter application"

4. In the next page, click on "API keys" tab, and copy your "API key" and "API secret".

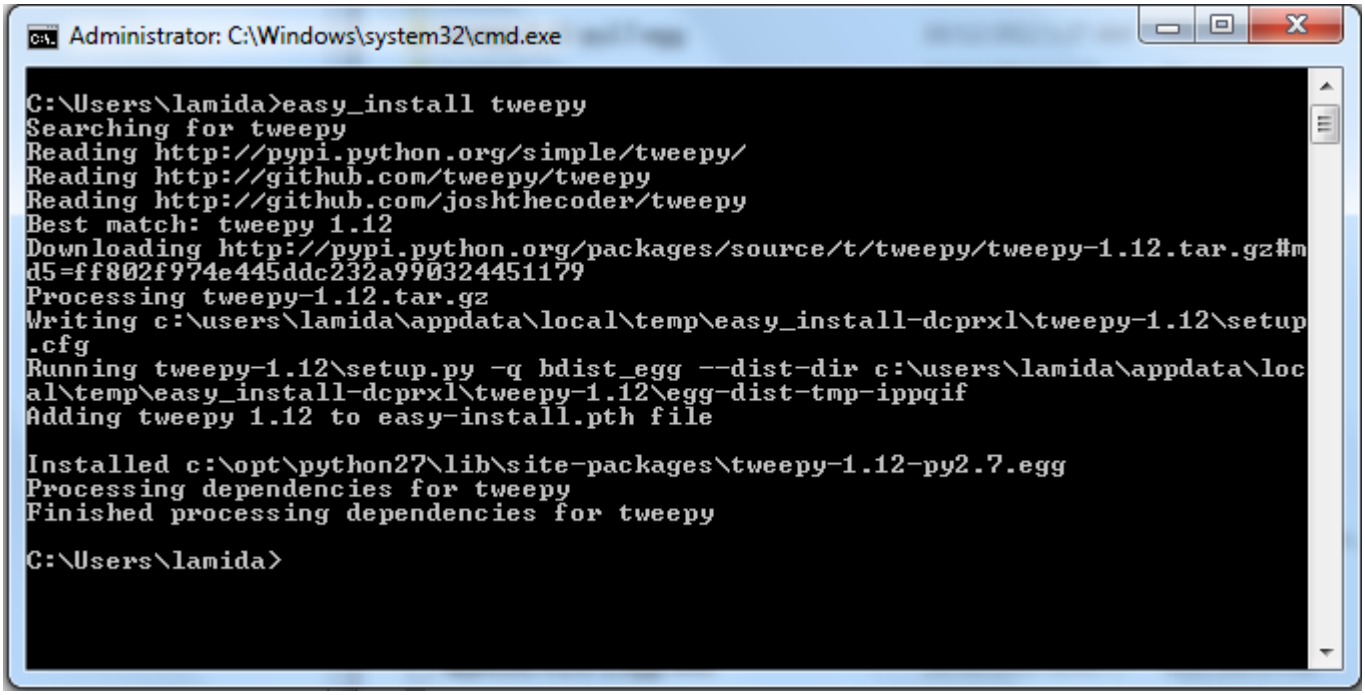
5. Click "Create my access token", and copy your "Access token" and "Access token secret".

```
TwitterAgent.sources = Twitter

TwitterAgent.sinks = HDFS

TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource

TwitterAgent.sources.Twitter.channels = MemChannel

TwitterAgent.sources.Twitter.accessToken = <accessToken>

TwitterAgent.sources.Twitter.keywords=women,education, employment
```

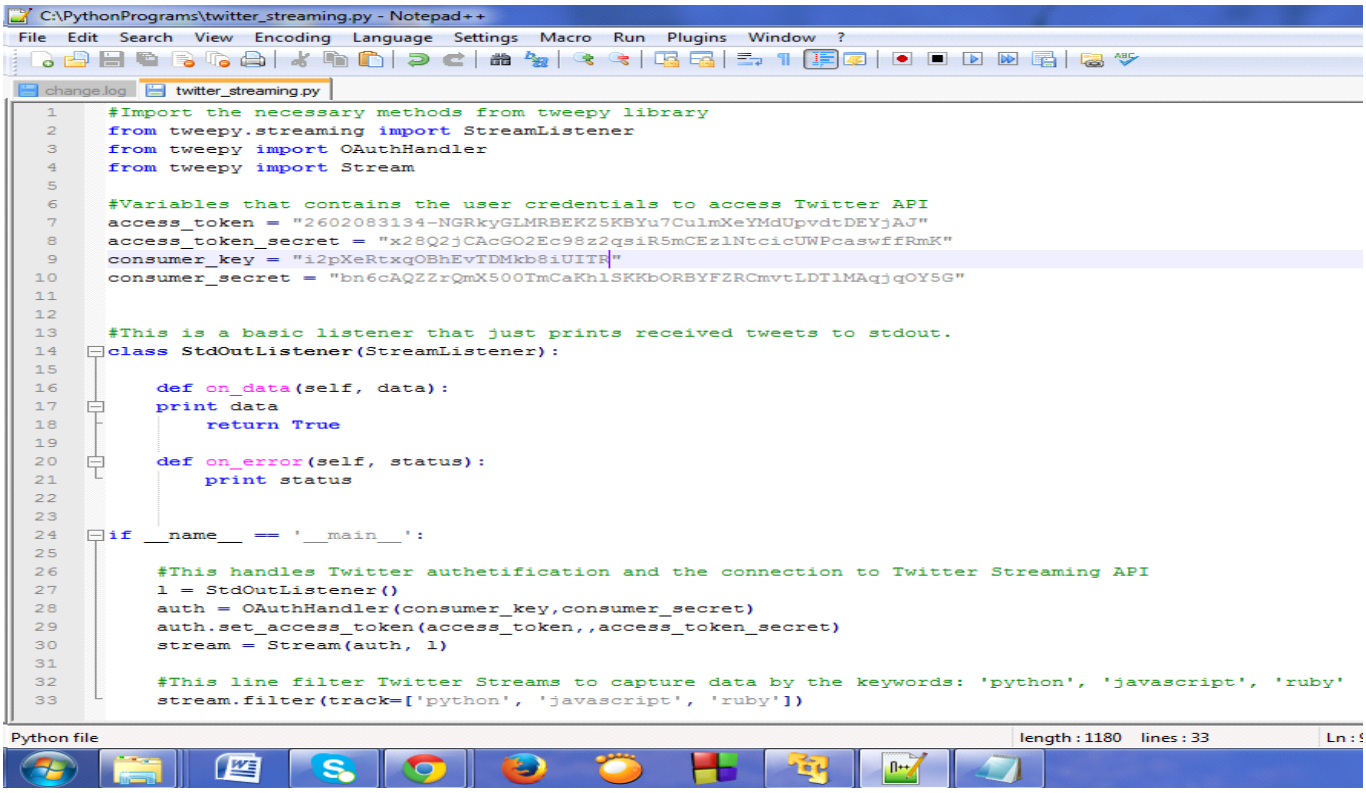


5.2 Connecting to Twitter Streaming API and downloading data

1. Install the tweepy python for importing the necessary methods from twitter library.



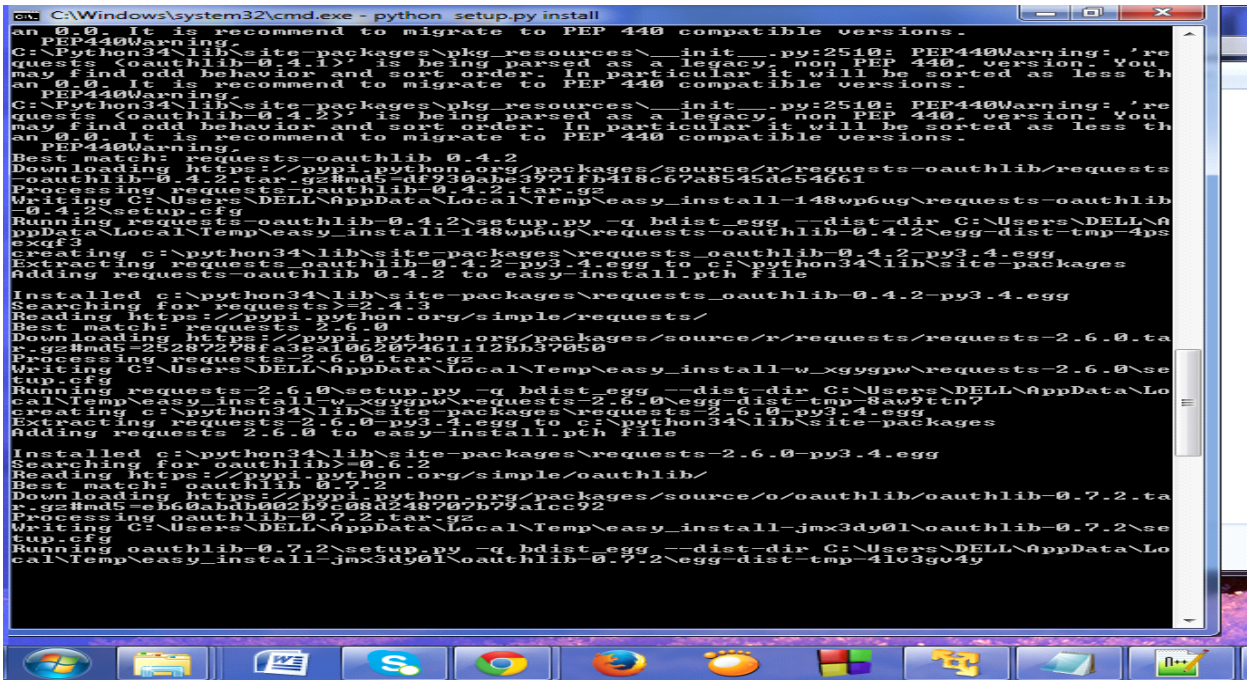
2. Implement the configuration keys on the notepad and save the file in the config folder:



3. Execute the following command on the terminal and saving the extracted tweets on data.txt :

“ python twitterstreamingapi.py > data.txt “

4. Getting the extracted data in json format



5. Run the program for 2 days (from 2015/01/15 till 2015/01/17) to get a meaningful data sample. This file size is 242 MB.

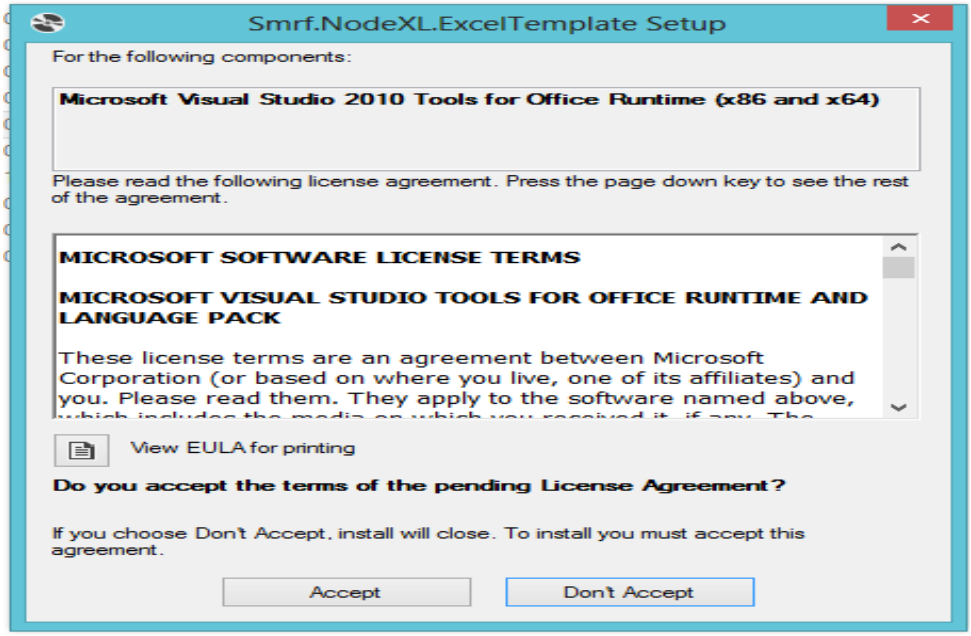
6. Standarize the extracted data in excel sheets format from www.json-xls.com/.

5.3 NodeXL Installation to Import Tweets

NodeXL is a general purpose network analysis application that supports network overview, discovery and exploration. The tool enables the automation of a data flow that starts with the collection of network data and moves through multiple steps until final processed network visualizations and reports are generated.

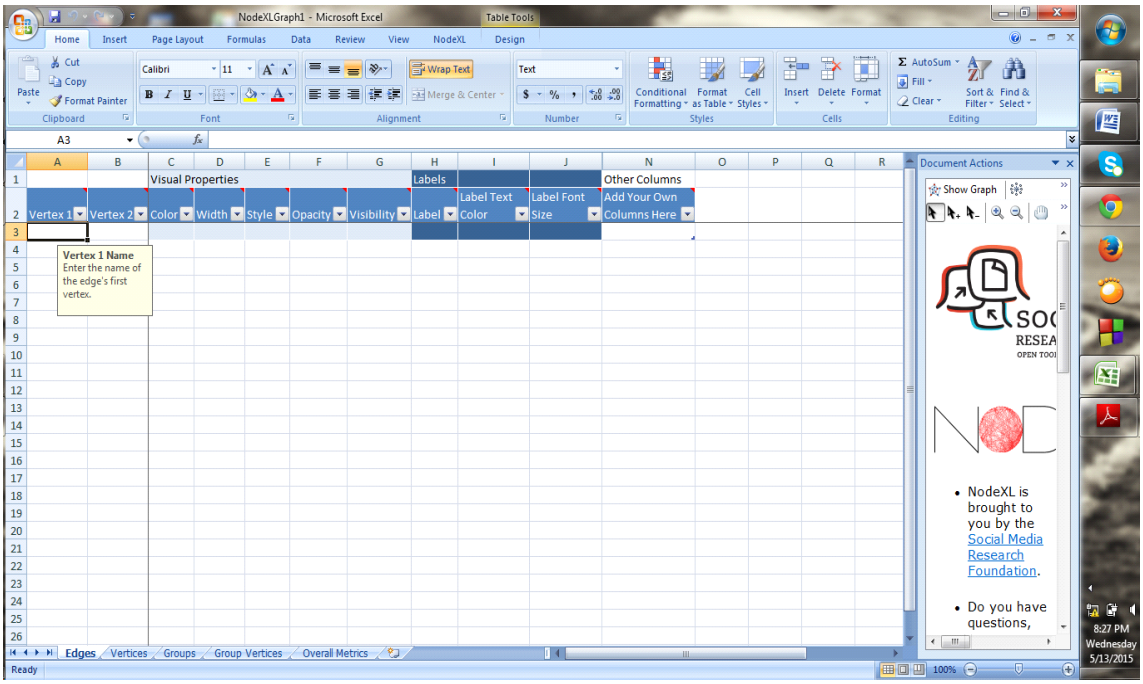
1. Download the NodeXL setup from the

2. Run the NodeXL set up and accept the liecense.



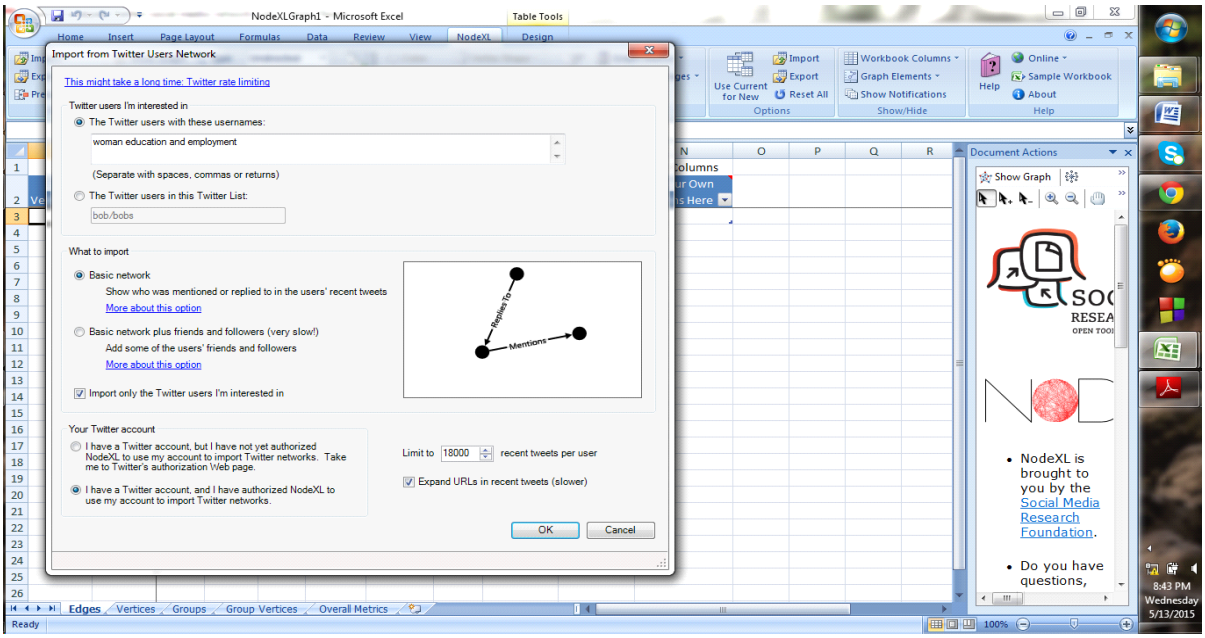
5.4 Mining Tweets in NodeXL

1. Open the NodeXL Sheets.

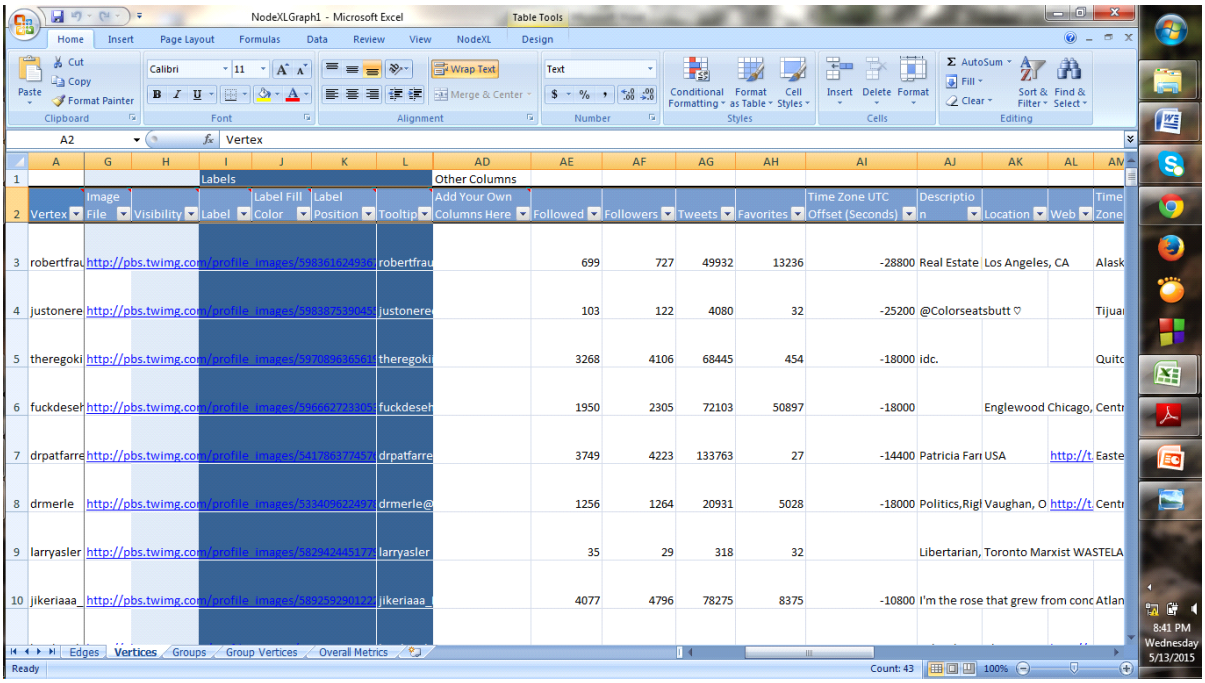


2. Import the standarized tweets in NodeXL excel sheet.

We used the NodeXL Twitter data import feature to extract networks. We configured our query using settings . We requested that NodeXL also add an “edge” which describes the connection between two twitter users that is formed when they follow, reply or mention one another. Data about each user along with the contents of their latest tweet were also selected to be added to the data set.



3. Getting tweets on NodeXL sheets.



Each "**edge**" represents a connection event between two people who tweeted within the data sample period. NodeXL constructs four different types of Twitter edges from the data it collects: follows, replies, mentions and tweet.

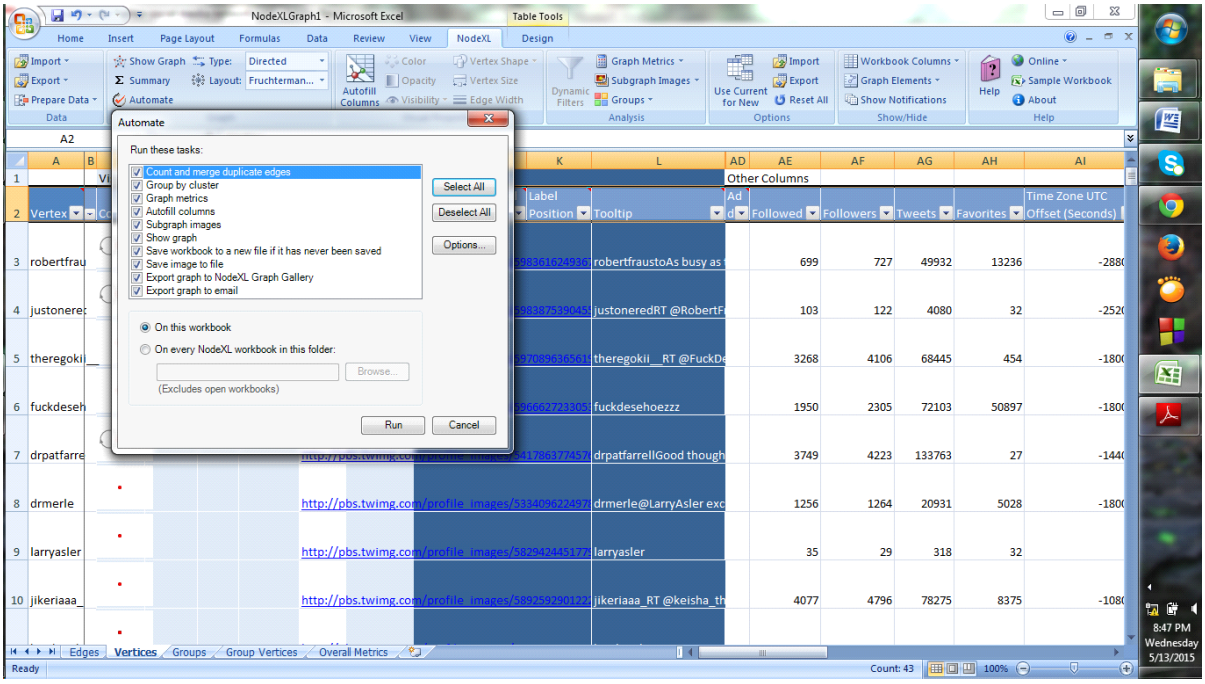
A "**follows**" edge is created if one author follows another who also tweeted in the sample dataset (the time stamp for a follows edge is the date of the query rather than the time when one user followed another user, which is information that is not available from Twitter).

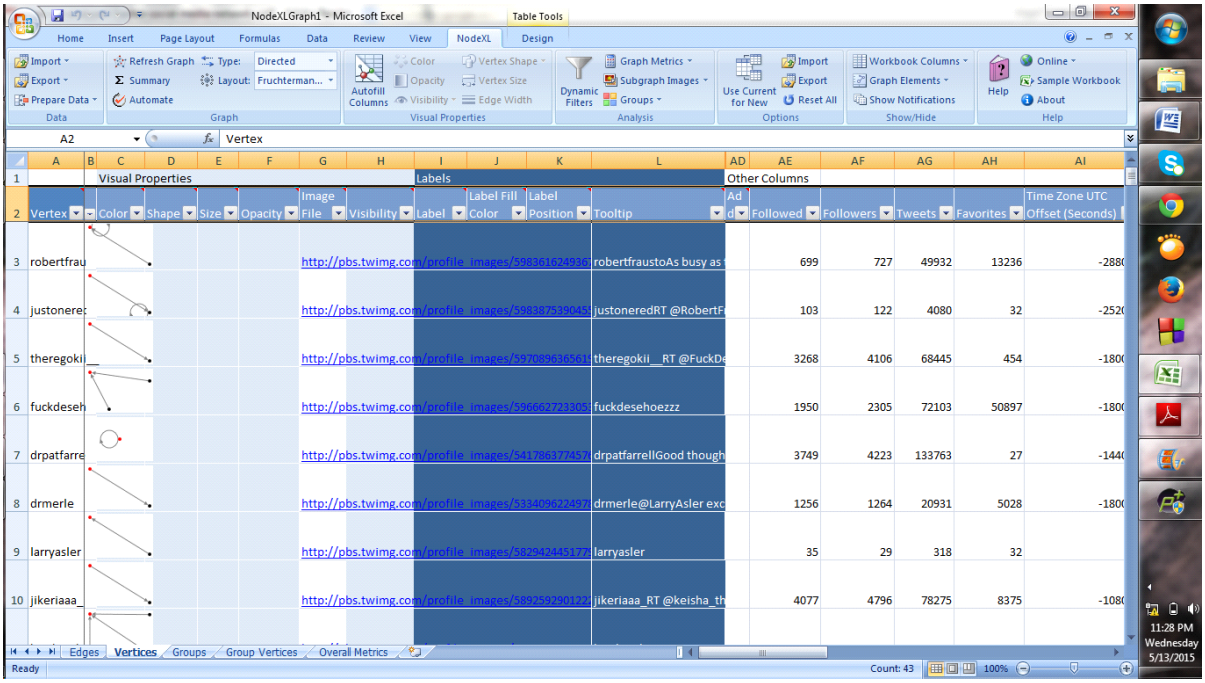
A "**mentions**" edge is created when one user creates a tweet that contains the name of another user (indicated with a preceeding "@" character, ex: "just spoke about social media with @marc_smith").

A "**reply**" relationship is a special form of "mention" that occurs when the user's name is at the very start of a tweet (ex: "@itaih just spoke about social media").

A tweet is a message that does not contain a reply or mention.

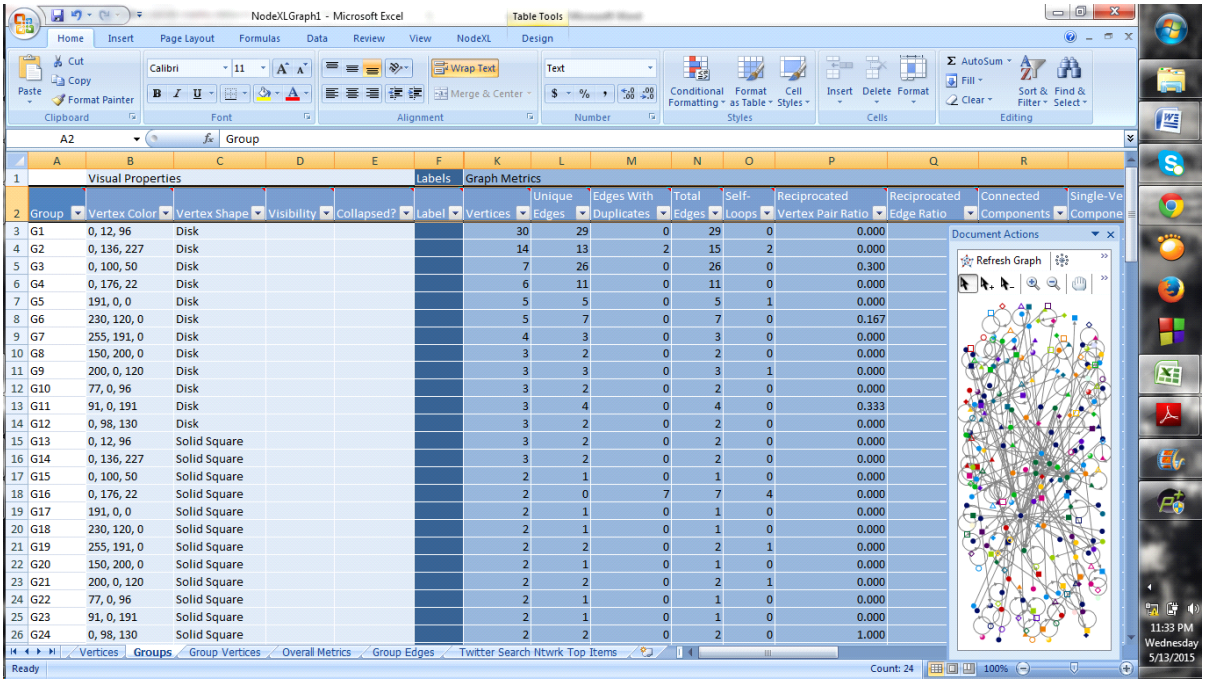
4. **Automate the NodeXL** : Using "Automate" NodeXL executes a series of user configured operations on the network without direct user control. The Automate dialog provides a good summary of the steps and operations applied to each network graph.





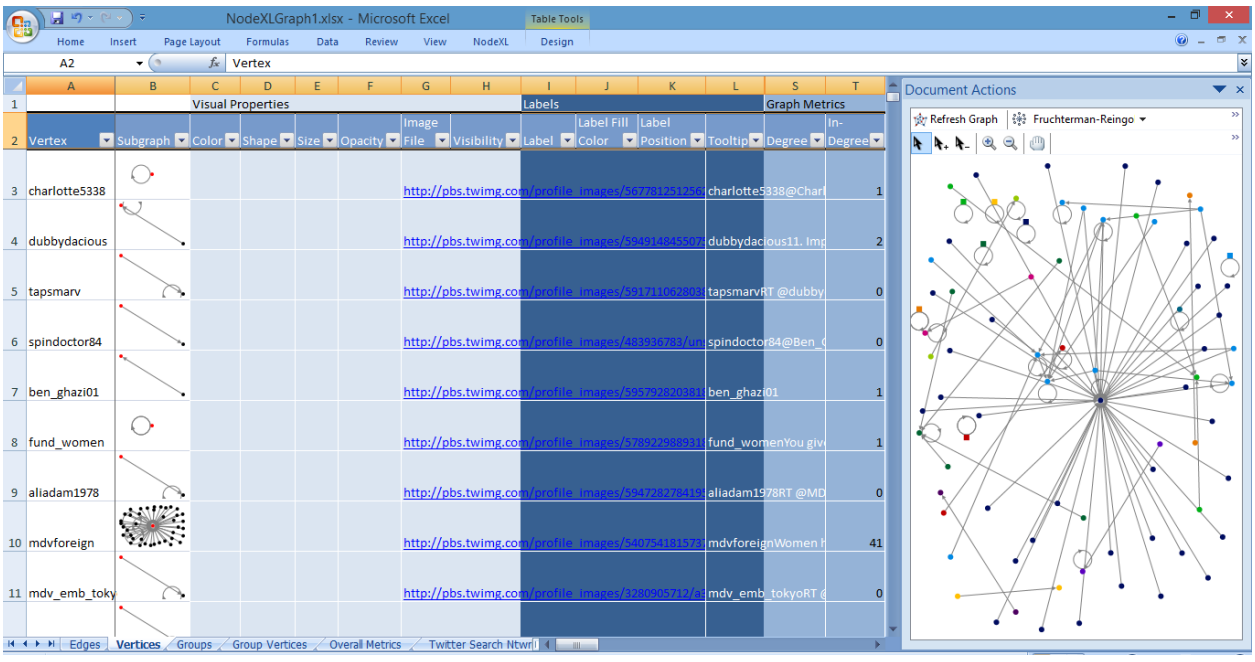
5. **Graph Metrics** : Each of the network metrics captures a different dimension of the size and shape of the graph as a whole and the location and connection properties of each person or entity in the network graph. We selected the creation all of the network metrics available through NodeXL.

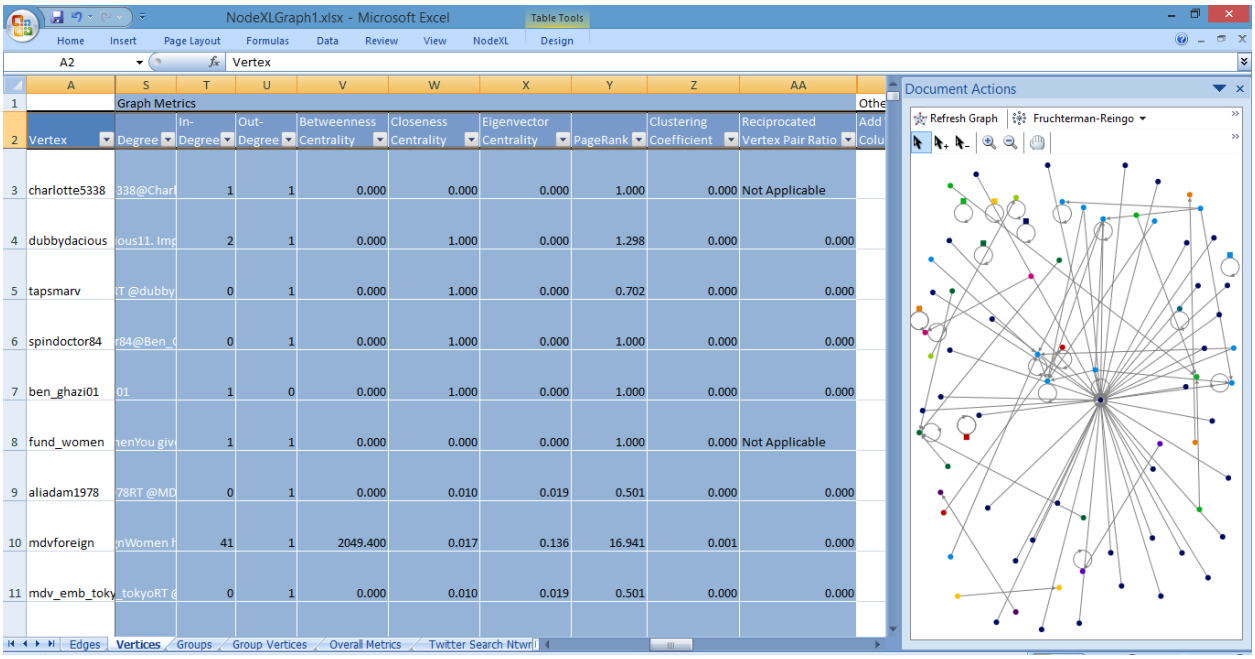
Many of these metrics can be mapped to various network display attributes. For example, the size of a vertex representing a Twitter user can be scaled to represent the number of users who have chosen to Follow each user.



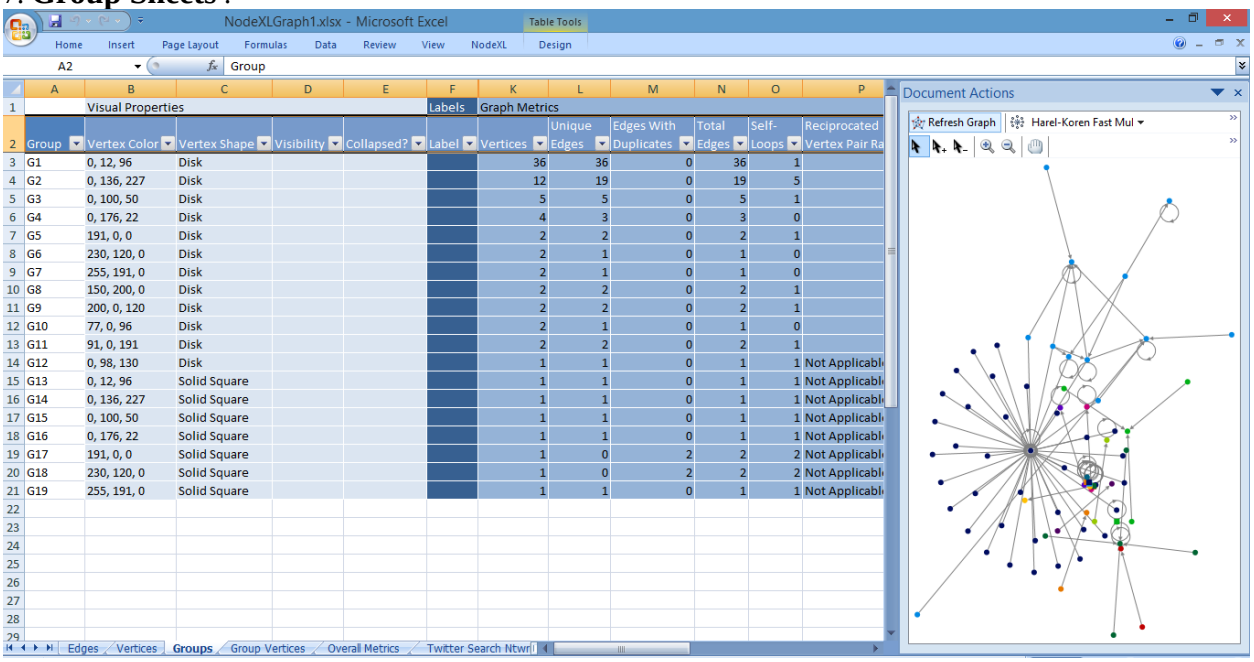
6. **Vertices**:

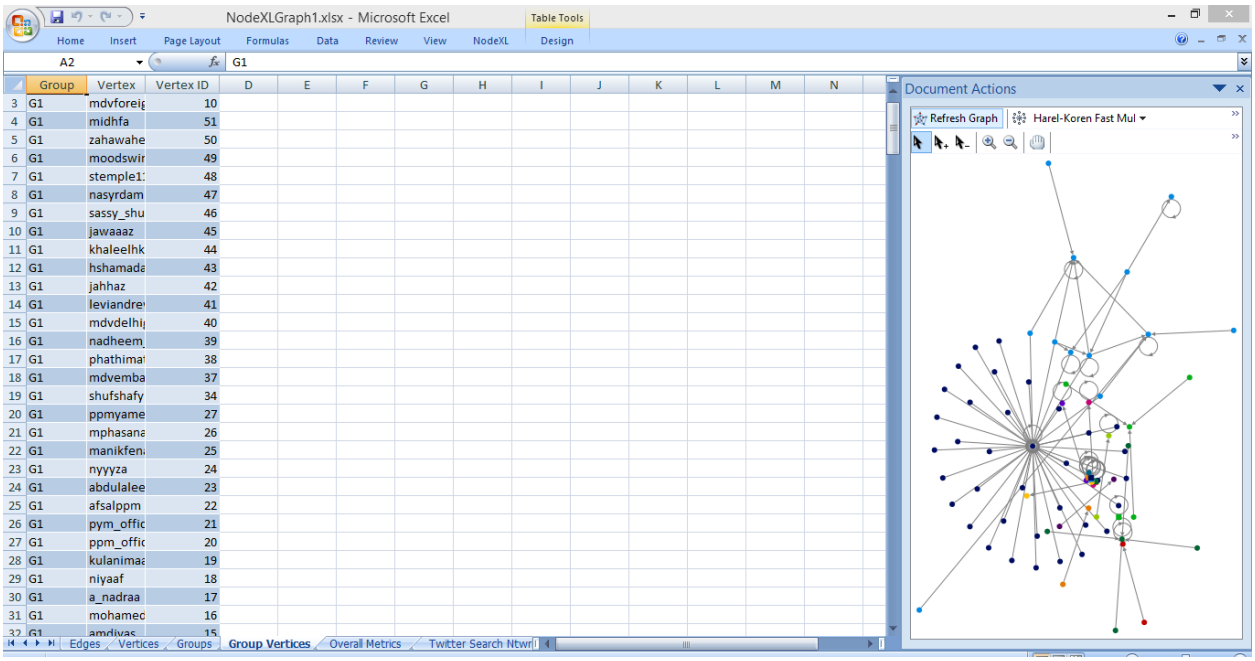
The Vertex worksheet displays attributes from Twitter along with a range of network metrics for eachvertex in the network. Each row represents one Twitter user who appeared in the results of the initial search query. A set of network metrics is listed for each user. Network metrics capture a range of qualities about thelocation and connection pattern of each user within the larger network.





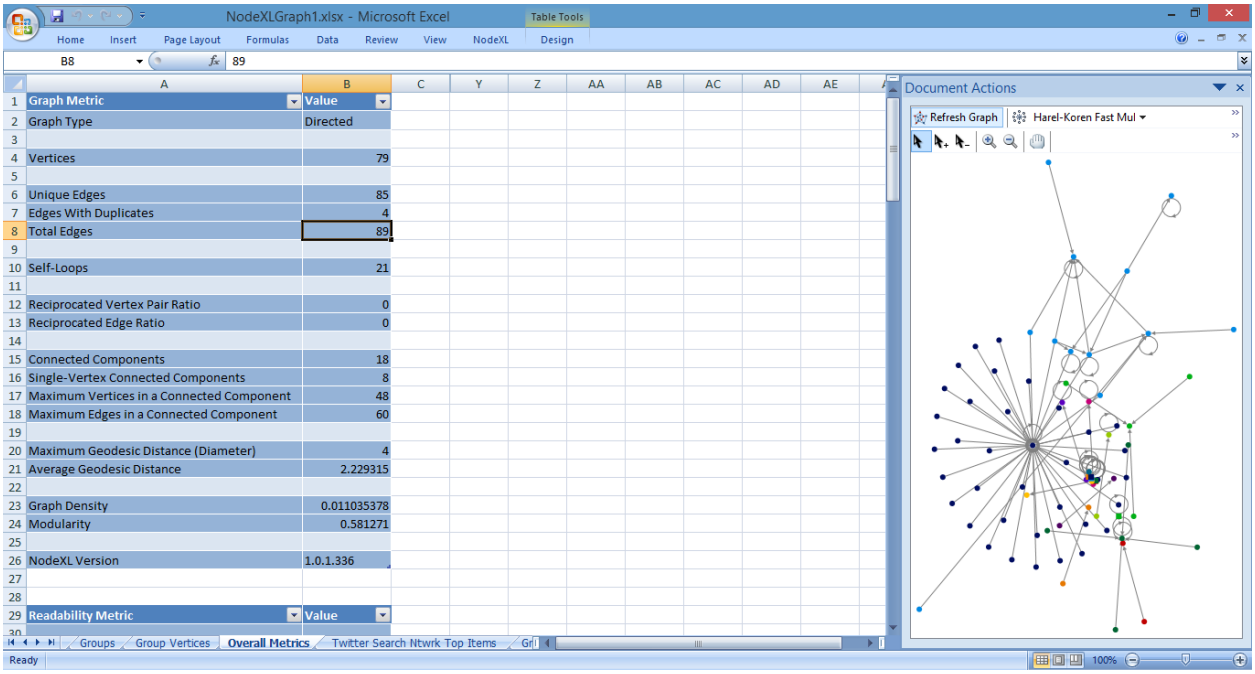
7. **Group Sheets** :

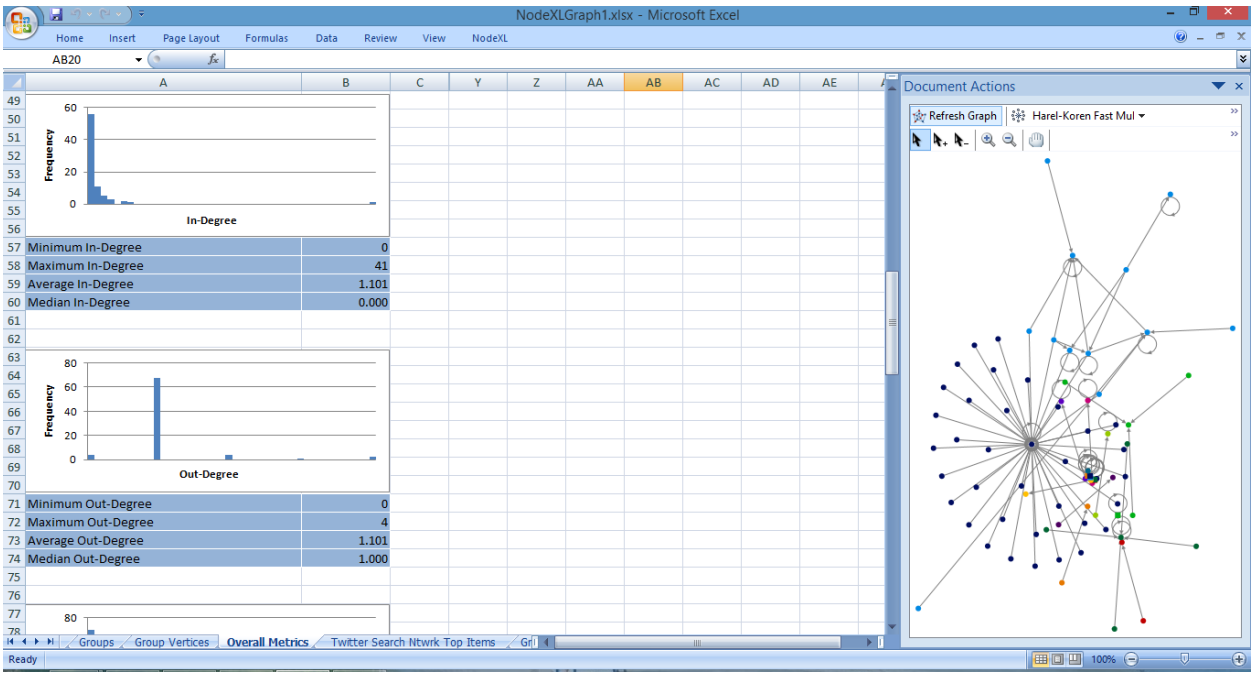




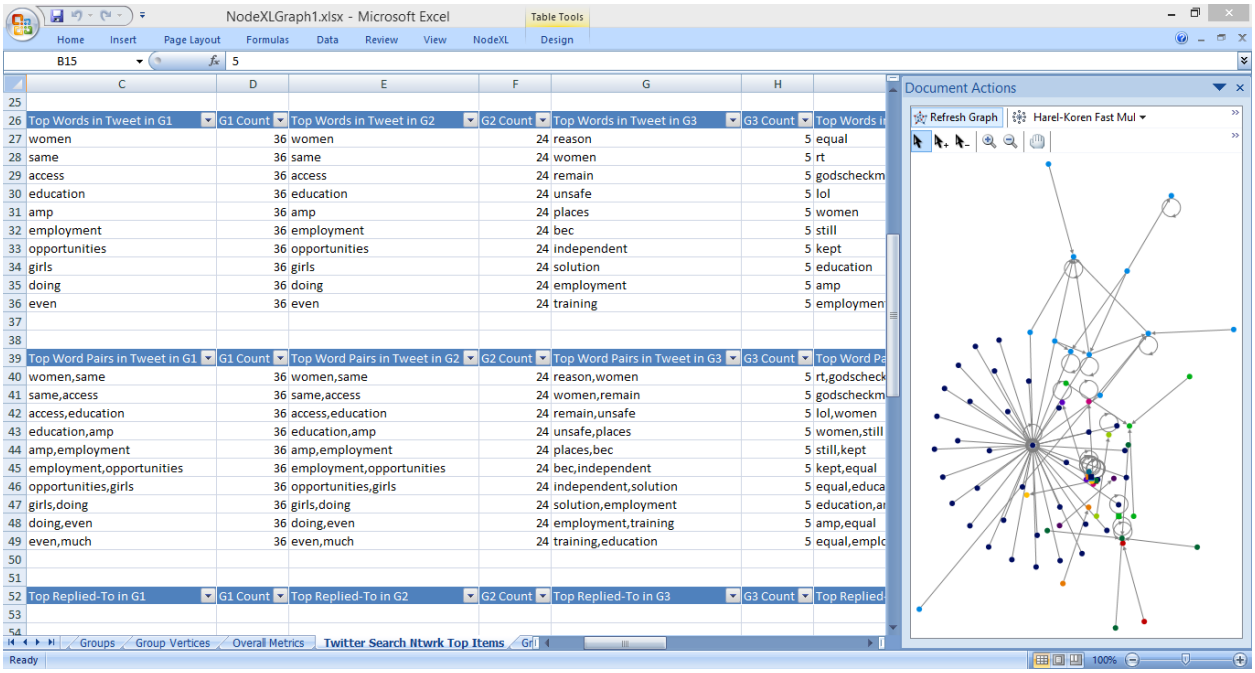
8. **Overall Metrics Sheet**:

These measures allow users to contrast networks to capture the ways networks in general and social media networks in particular compare to other networks and to themselves overtime.

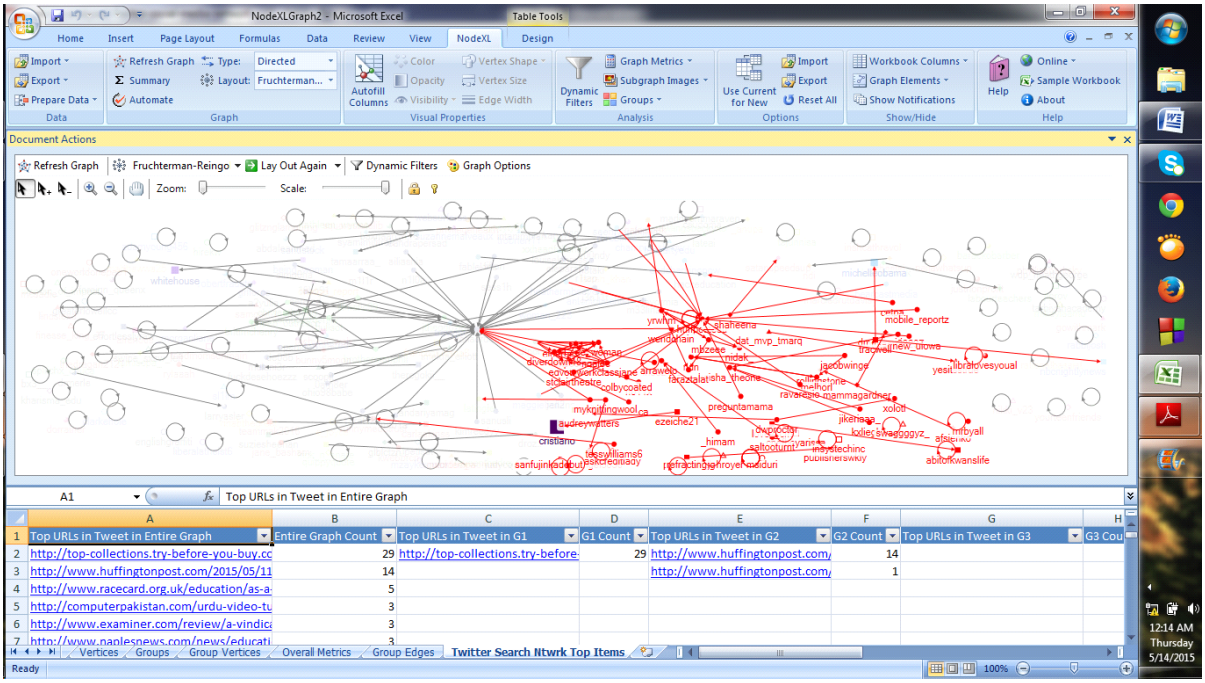




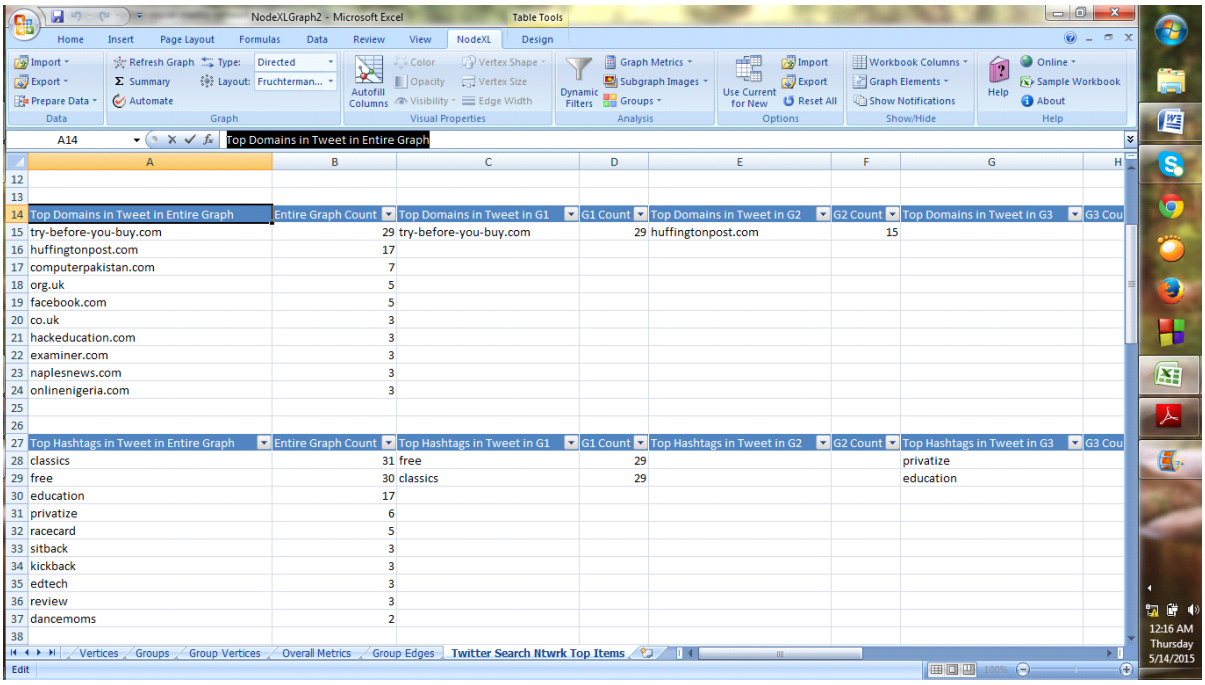
9. **Twitter Search Network Top Itmes**:



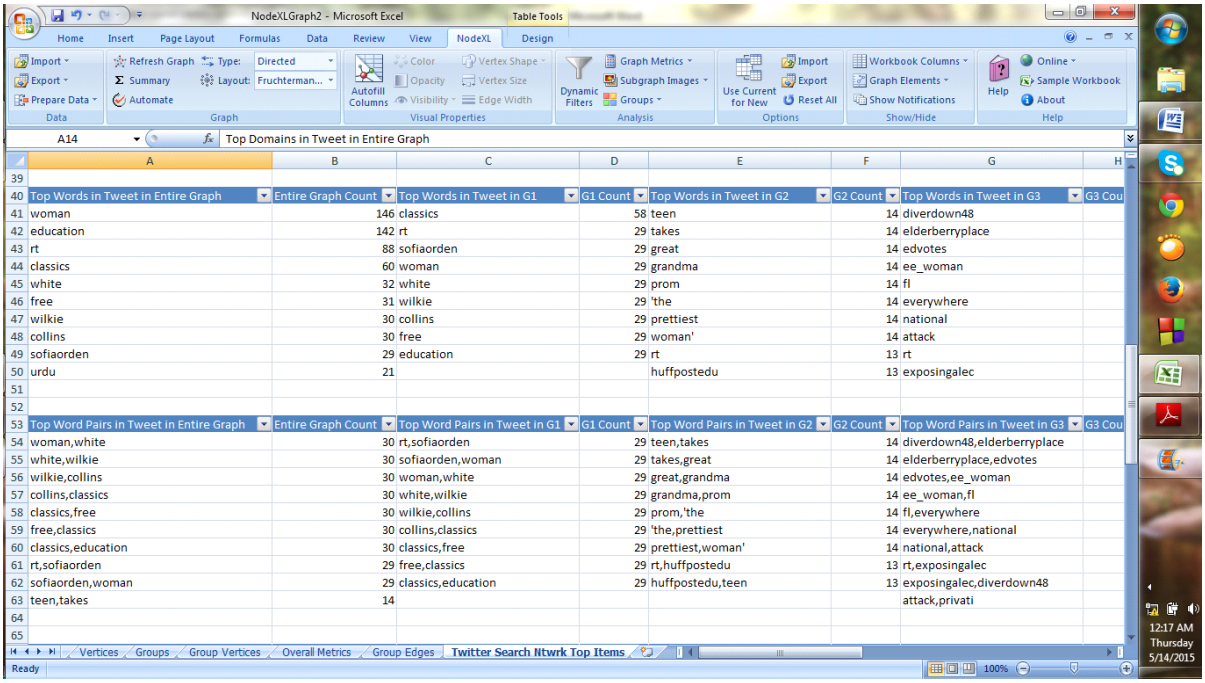
10. **Top URLs in Tweet in Entire Graph**

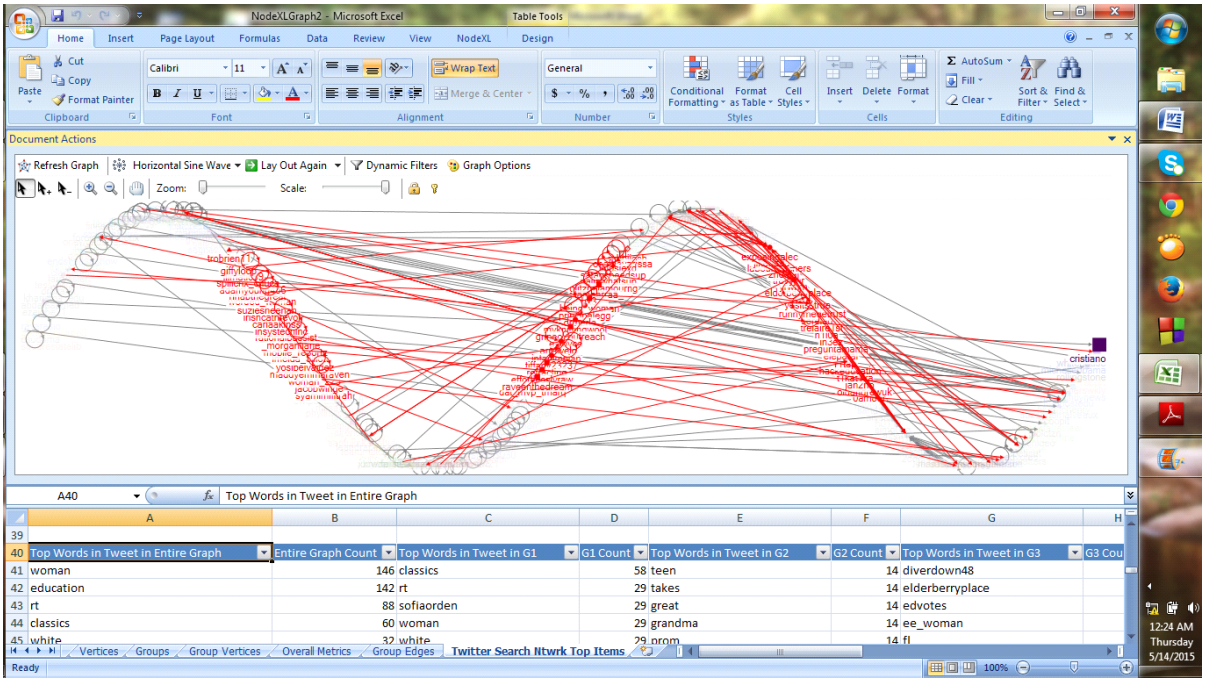


11.**Top Domains and Hashtag in Tweet in Entire Graph**

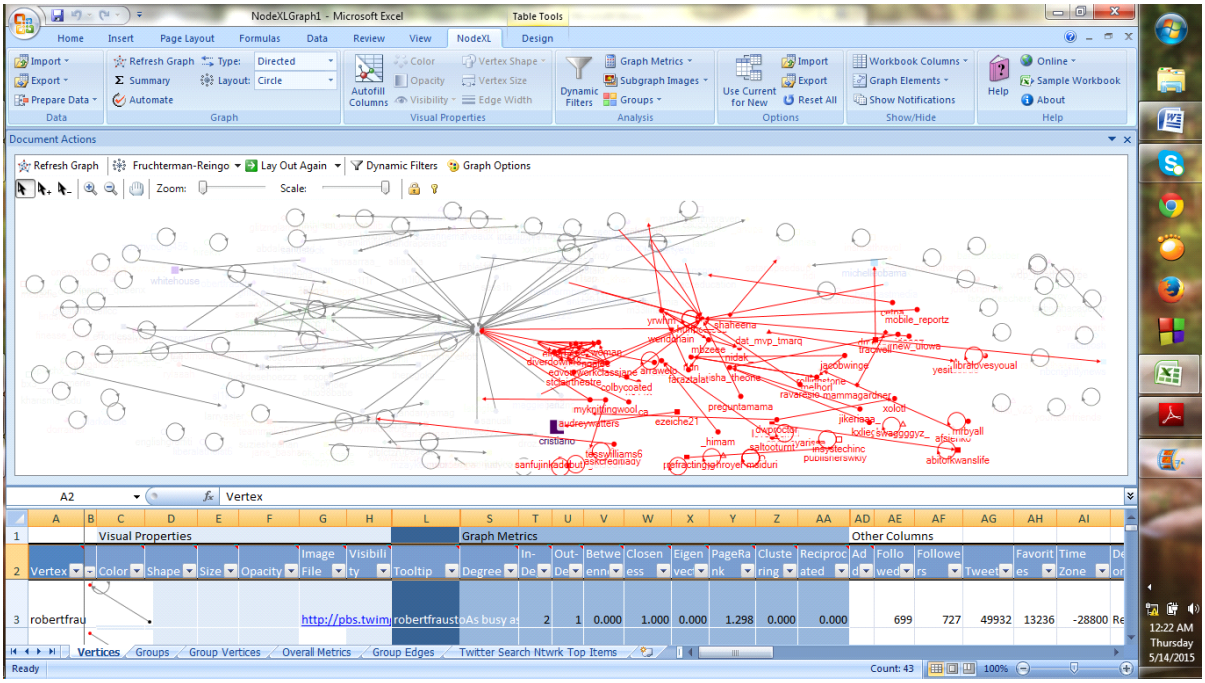


12 .**Top Words in Tweets in Entire Graph**

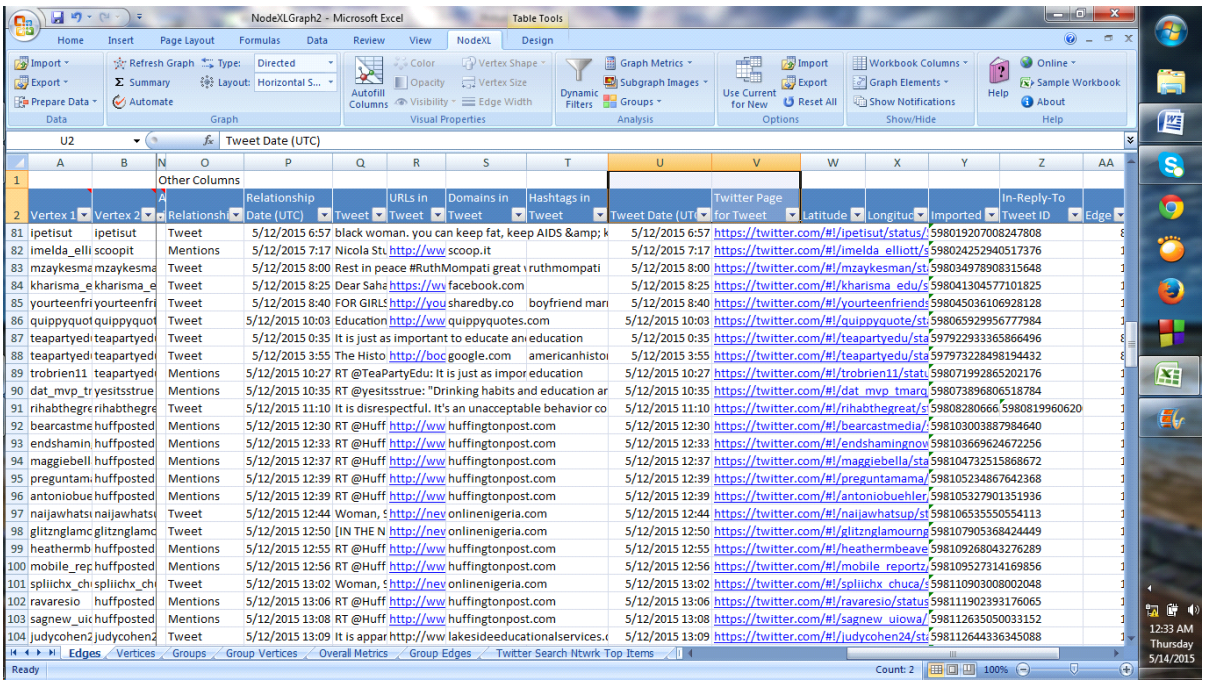




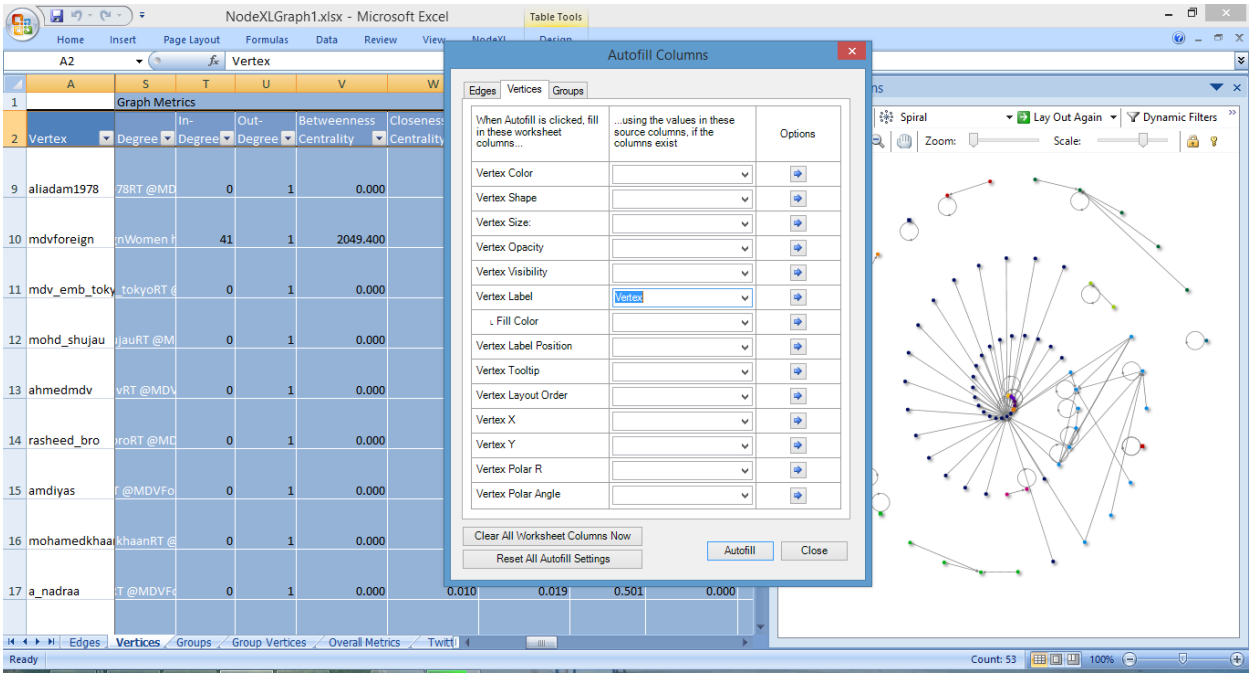
Tweets which are directly related to one common person



13.**Count and Merge Duplicates**



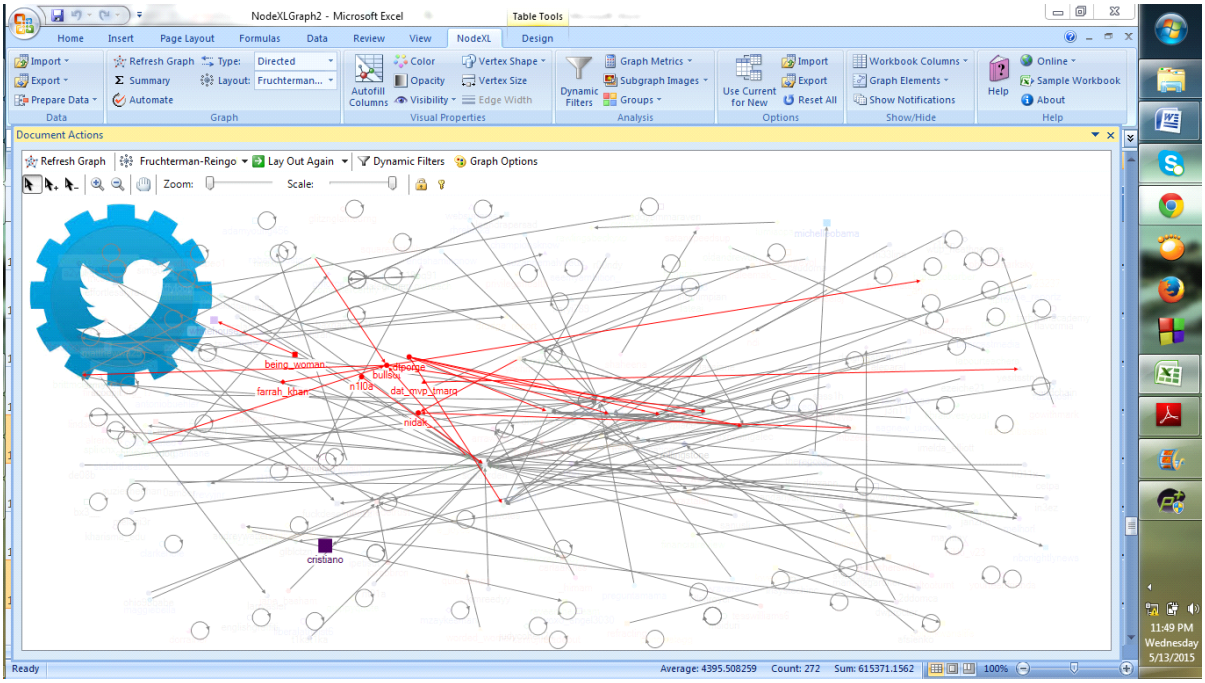
14. **AutoFill Columns** : NodeXL has a feature called “Autofill Columns” that makes it simple to pick attributes about each edge and vertex and map them to display attributes like the size, color, shape, or transparency of each vertex.

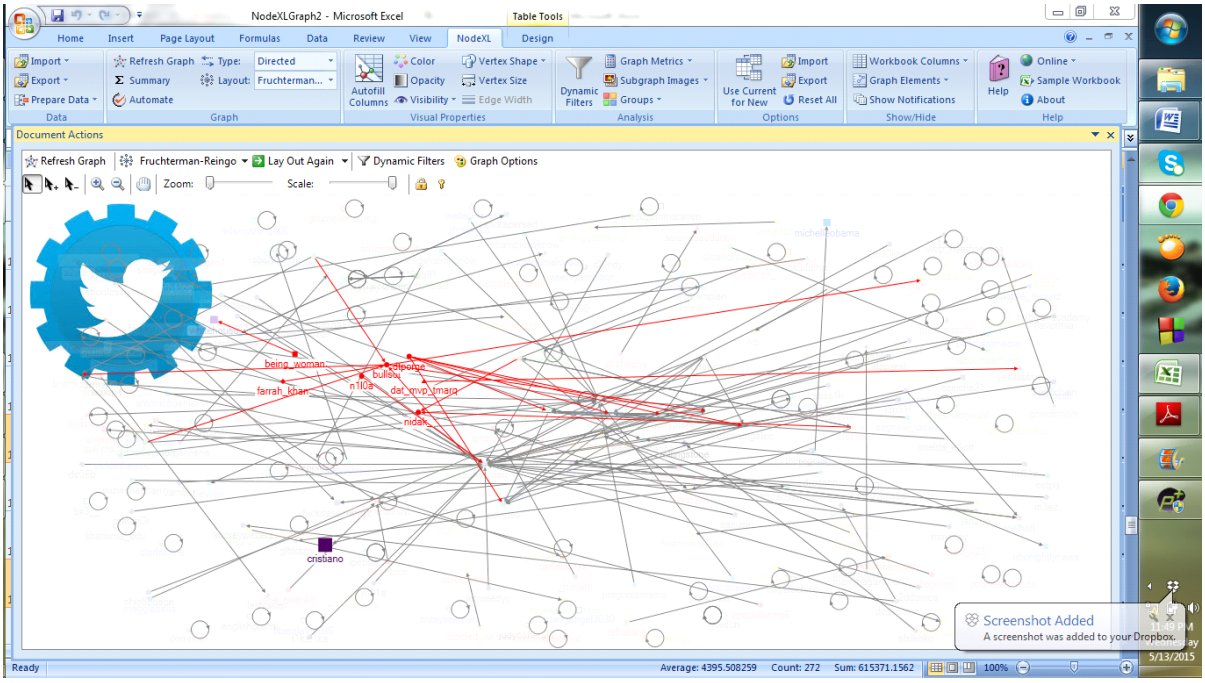


Using Autofill columns, we set the size of each vertex to be proportionate to the number of Followers the user had attracted. We also set the “opacity” of the vertex (which controls the transparency of the image) to be inverse to the number of Followers. This setting is controlled by the options which are accessed by the arrow on each row of the Autofill columns dialog. An inverse mapping for opacity means that the largest objects are somewhat transparent, allowing the other, smaller users to be seen through the images of the larger, more popular users.

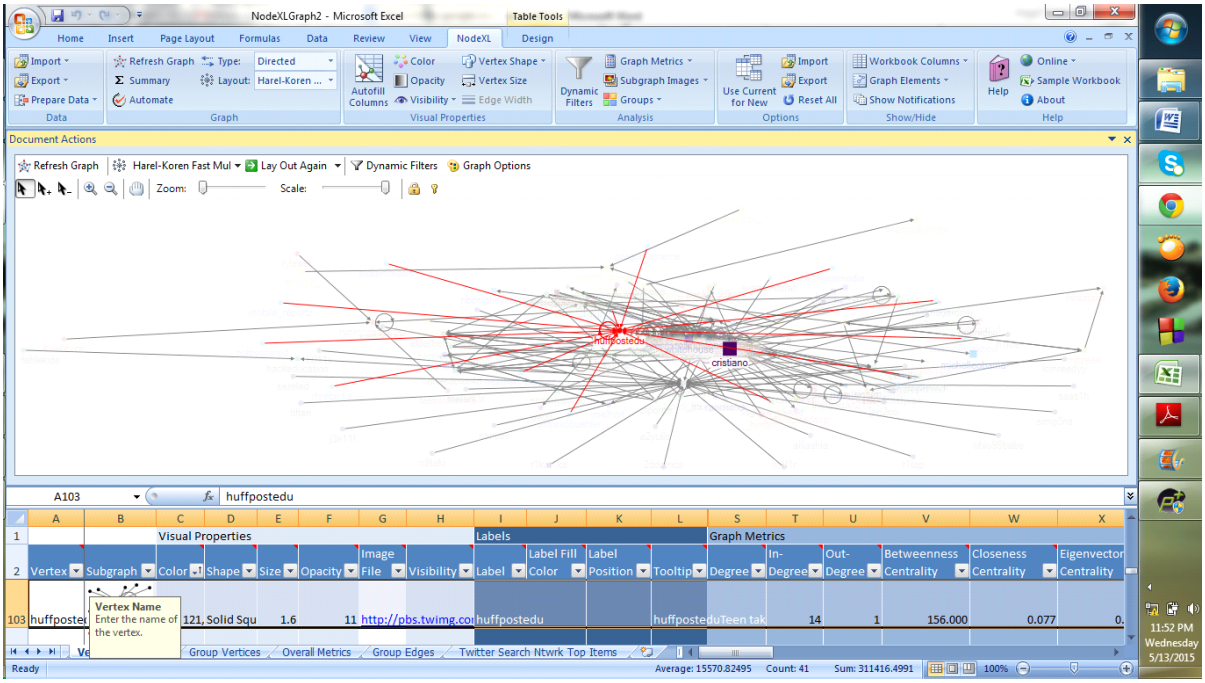
The vertex label is set to the name of the user. The vertex label is drawn beneath the profile photo from Twitter that is used to represent each user. The order of the layout controls the way each vertex is placed on the screen. Setting it to equal the same value as the size value means that objects often line up in size order.

Frusta Graph

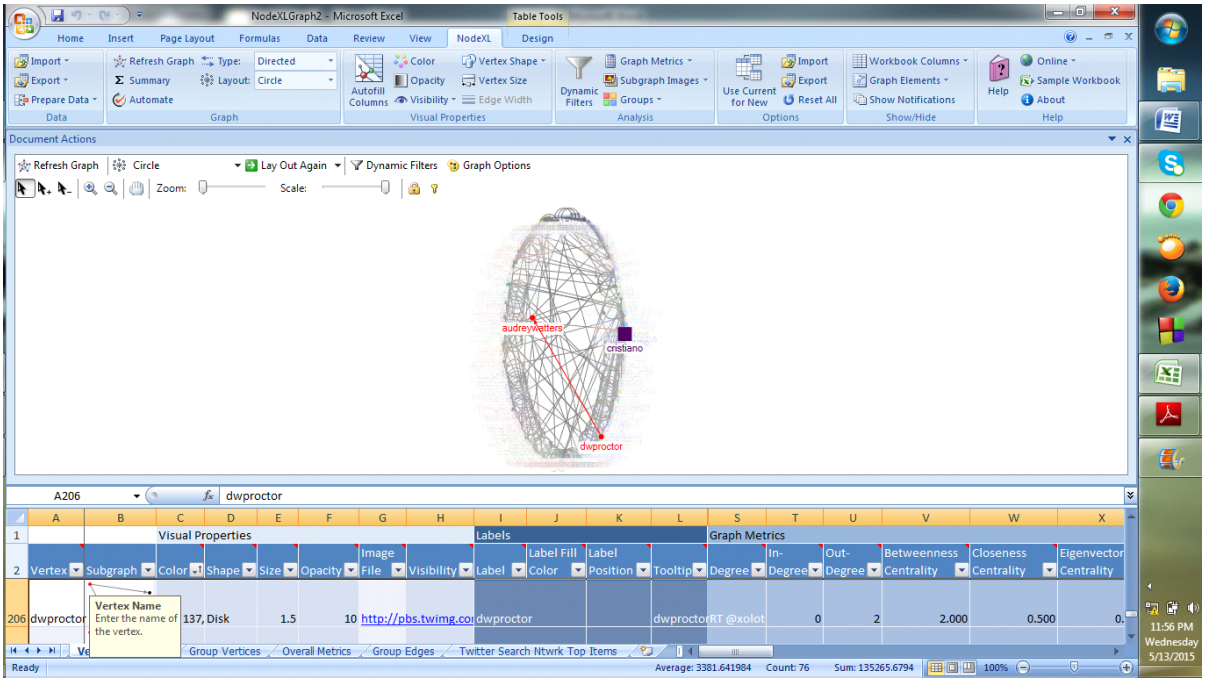

NodeXLGraph2 - Microsoft Excel
Table Tools
Home
Insert
Page Layout
Formulas
Data
Review
View
NodeXL
Design
Import
Export
Prepare Data
Refresh Graph
Summary
Automate
Type:
Directed
Layout:
Fruchterman...
Data
Graph
Autofill Columns
Color
Opacity
Visibility
Vertex Shape
Vertex Size
Edge Width
Visual Properties
Dynamic Filters
Graph Metrics
Subgraph Images
Groups
Analysis
Use Current for New
Import
Export
Reset All
Options
Workbook Columns
Graph Elements
Show Notifications
Show/Hide
Help
Online
Sample Workbook
About
Help
Document Actions
Refresh Graph
Fruchterman-Reingo
Lay Out Again
Dynamic Filters
Graph Options
Zoom:
Scale:
being_woman
farrah_khan
n1108
nidak
cristiano
Ready
Average: 4395.508259
Count: 272
Sum: 615371.1562
100%
11:49 PM
Wednesday
5/13/2015

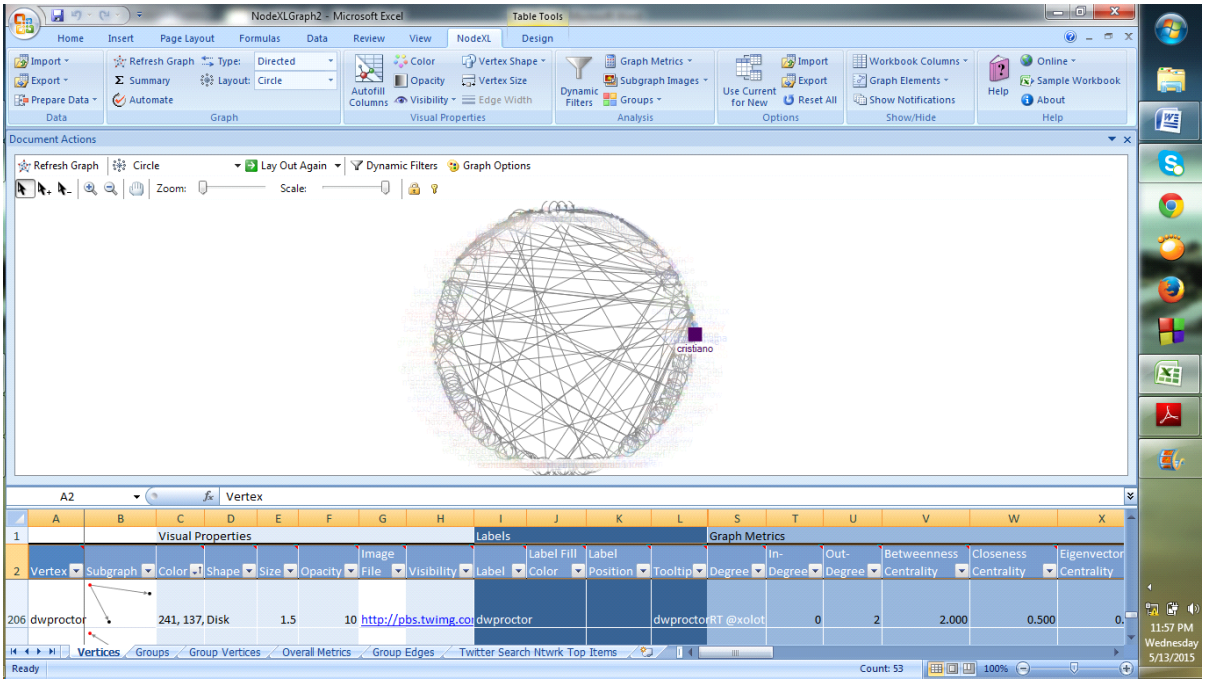

NodeXLGraph2 - Microsoft Excel
Table Tools
Home
Insert
Page Layout
Formulas
Data
Review
View
NodeXL
Design
Import
Export
Prepare Data
Refresh Graph
Summary
Automate
Type:
Directed
Layout:
Fruchterman...
Data
Graph
Autofill Columns
Color
Opacity
Visibility
Vertex Shape
Vertex Size
Edge Width
Visual Properties
Dynamic Filters
Graph Metrics
Subgraph Images
Groups
Analysis
Use Current for New
Import
Export
Reset All
Options
Workbook Columns
Graph Elements
Show Notifications
Show/Hide
Help
Online
Sample Workbook
About
Help
Document Actions
Refresh Graph
Fruchterman-Reingo
Lay Out Again
Dynamic Filters
Graph Options
Zoom:
Scale:
being_woman
farrah_khan
n1108
nidak
cristiano
Screenshot Added
A screenshot was added to your Dropbox.
Ready
Average: 4395.508259
Count: 272
Sum: 615371.1562
100%
11:49 PM
Wednesday
5/13/2015

Harel Koren

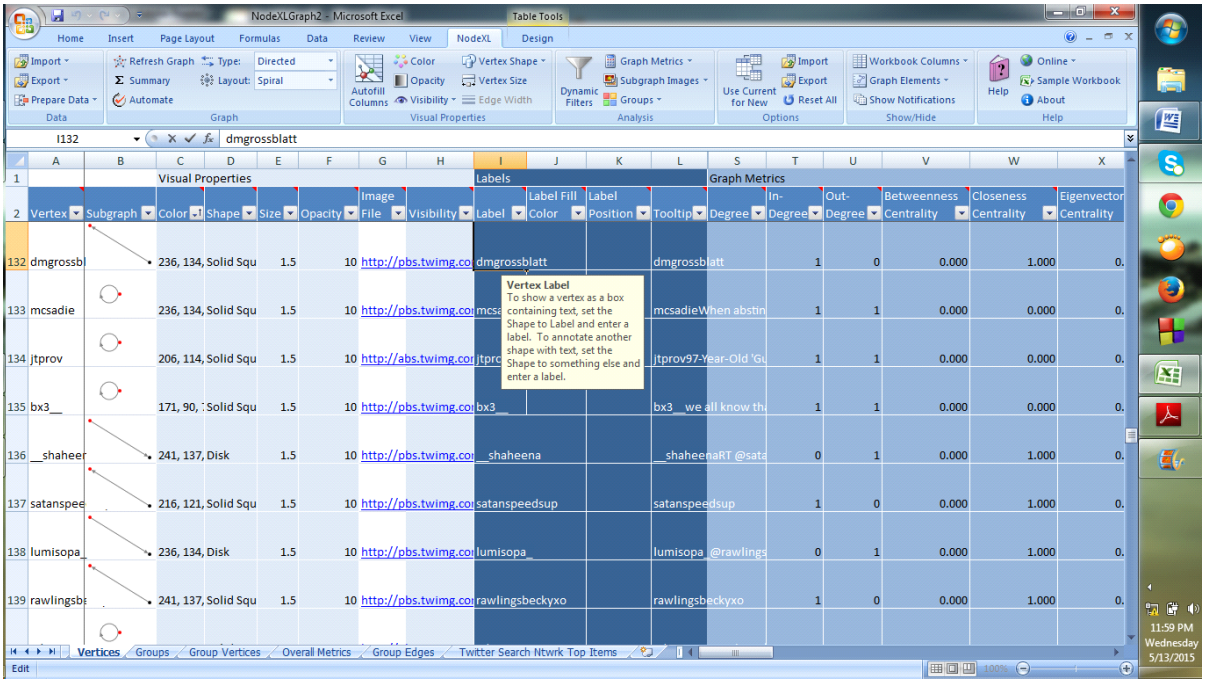

NodeXLGraph2 - Microsoft Excel
Table Tools
Refresh Graph
Harel-Koren Fast Mul
Lay Out Again
Dynamic Filters
Graph Options
huffpostedu
cristiano
A103
huffpostedu
Visual Properties
Labels
Graph Metrics
Vertex Name
Enter the name of the vertex.
Group Vertices
Overall Metrics
Group Edges
Twitter Search Ntwrk Top Items
Ready
Average: 15570.82495
Count: 41
Sum: 311416.4991
11:52 PM
Wednesday
5/13/2015

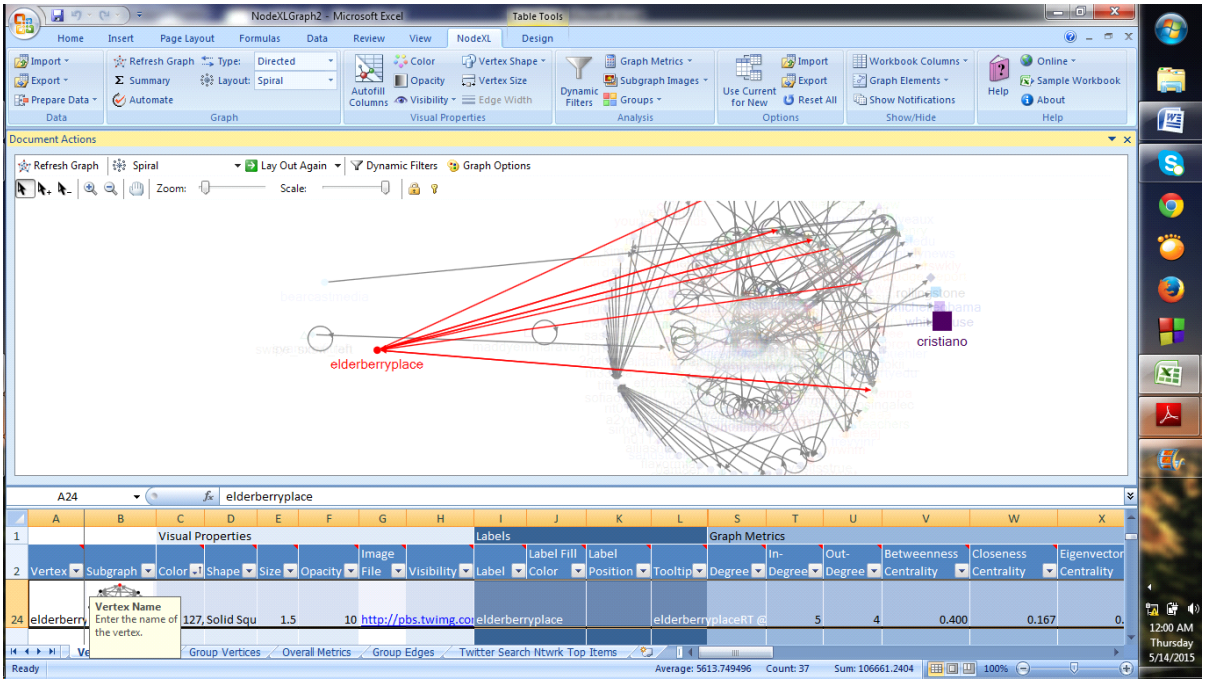
Circle

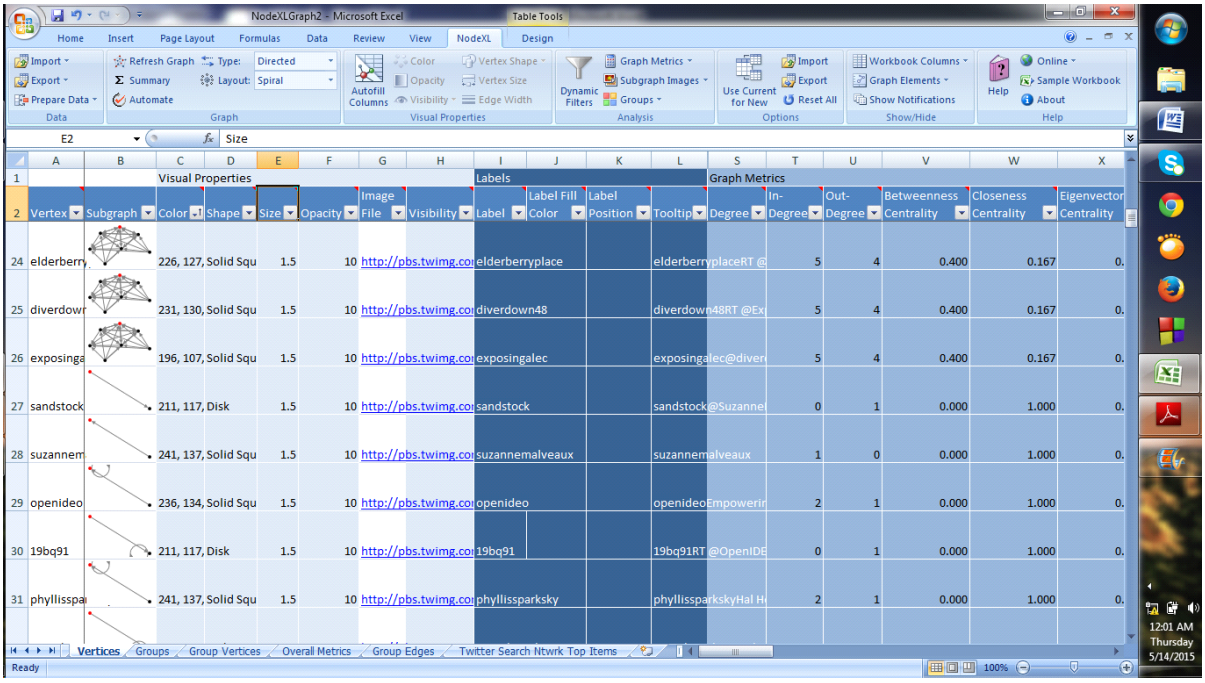

NodeXLGraph2 - Microsoft Excel
Table Tools
Refresh Graph
Circle
Lay Out Again
Dynamic Filters
Graph Options
audreywetters
cristiano
dwproctor
A206
dwproctor
Visual Properties
Labels
Graph Metrics
Vertex Name
Enter the name of the vertex.
Group Vertices
Overall Metrics
Group Edges
Twitter Search Ntwrk Top Items
Ready
Average: 3381.641984
Count: 76
Sum: 135265.6794
11:56 PM
Wednesday
5/13/2015


NodeXLGraph2 - Microsoft Excel
Table Tools
Refresh Graph
Circle
Lay Out Again
Dynamic Filters
Graph Options
Zoom:
Scale:
cristiano
A2
Vertex
Visual Properties
Labels
Graph Metrics
Vertex
Subgraph
Color
Shape
Size
Opacity
Image File
Visibility
Label
Label Fill Color
Label Position
Tooltip
Degree
In-Degree
Out-Degree
Betweenness Centrality
Closeness Centrality
Eigenvector Centrality
206 dwproctor
241, 137, Disk
1.5
10
http://pbs.twimg.co
dwproctor
dwproctor
RT @xolot
0
2
2.000
0.500
Vertices
Groups
Group Vertices
Overall Metrics
Group Edges
Twitter Search Ntwrk Top Items
Ready
Count: 53
11:57 PM
Wednesday
5/13/2015

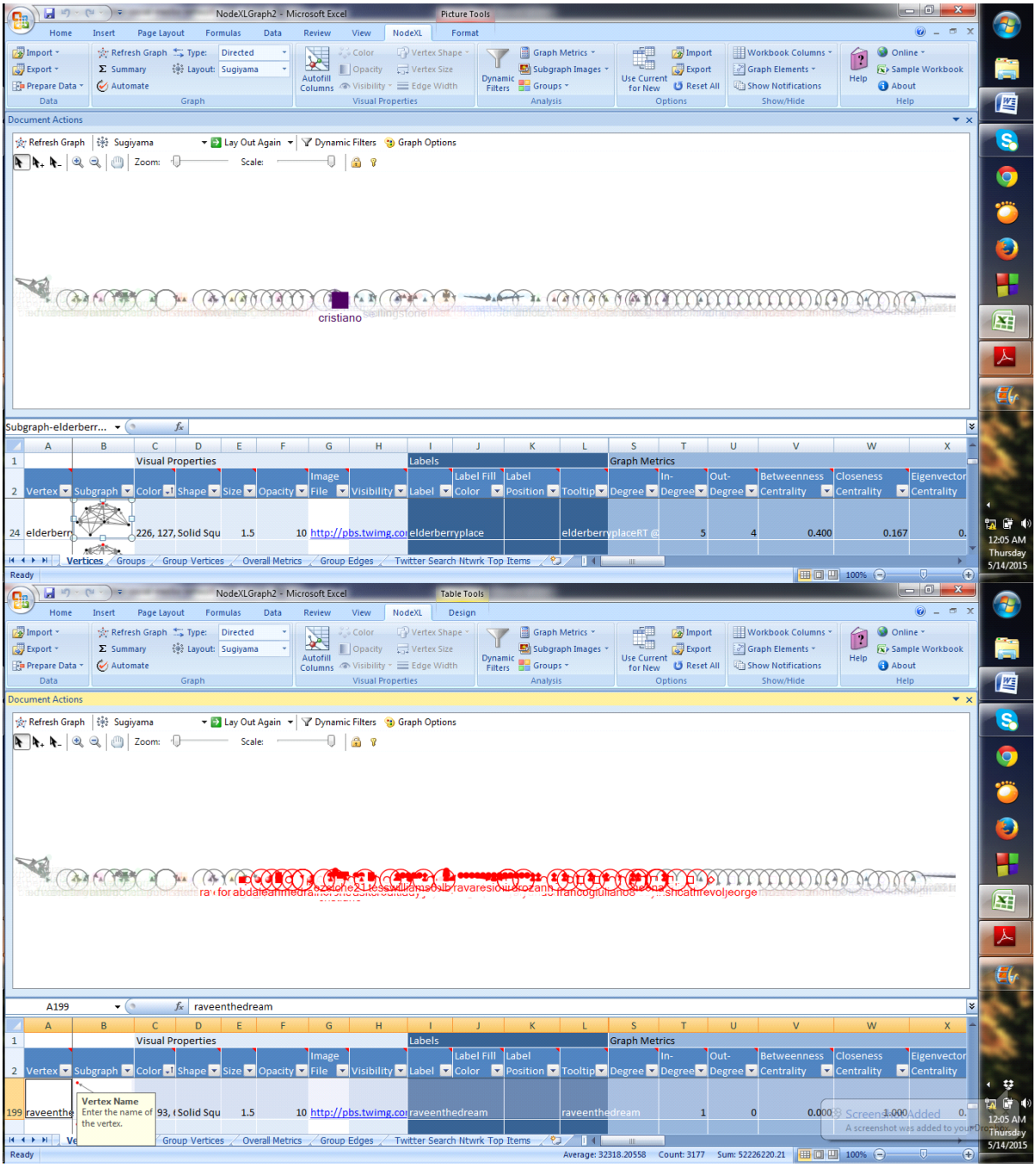
Spiral


NodeXLGraph2 - Microsoft Excel
Table Tools
Spiral
I132
dmgrossblatt
Visual Properties
Labels
Graph Metrics
132 dmgrossbl
236, 134, Solid Squ
1.5
10
http://pbs.twimg.co
dmgrossblatt
dmgrossblatt
1
0
0.000
1.000
133 mcsadie
236, 134, Solid Squ
1.5
10
http://pbs.twimg.co
mcsadieWhen abstin
1
1
0.000
0.000
134 jtprov
206, 114, Solid Squ
1.5
10
http://abs.twimg.co
jtprov97-Year-Old 'Gu
1
1
0.000
0.000
135 bx3__
171, 90, Solid Squ
1.5
10
http://pbs.twimg.co
bx3__
bx3__we all know th
1
1
0.000
0.000
136 __shaheer
241, 137, Disk
1.5
10
http://pbs.twimg.co
__shaheena
__shaheenaRT @sata
0
1
0.000
1.000
137 satanspee
216, 121, Solid Squ
1.5
10
http://pbs.twimg.co
satanspeedsup
satanspeedsup
1
0
0.000
1.000
138 lumisopa_
236, 134, Disk
1.5
10
http://pbs.twimg.co
lumisopa_
lumisopa_@rawlings
0
1
0.000
1.000
139 rawlingsbe
241, 137, Solid Squ
1.5
10
http://pbs.twimg.co
rawlingsbeckyxo
rawlingsbeckyxo
1
0
0.000
1.000
Vertex Label
To show a vertex as a box containing text, set the Shape to Label and enter a label. To annotate another shape with text, set the Shape to something else and enter a label.
Vertices
Groups
Group Vertices
Overall Metrics
Group Edges
Twitter Search Ntwrk Top Items
Edit
11:59 PM
Wednesday
5/13/2015


NodeXLGraph2 - Microsoft Excel
Table Tools
elderberryplace
cristiano
Vertex Name
Enter the name of the vertex.

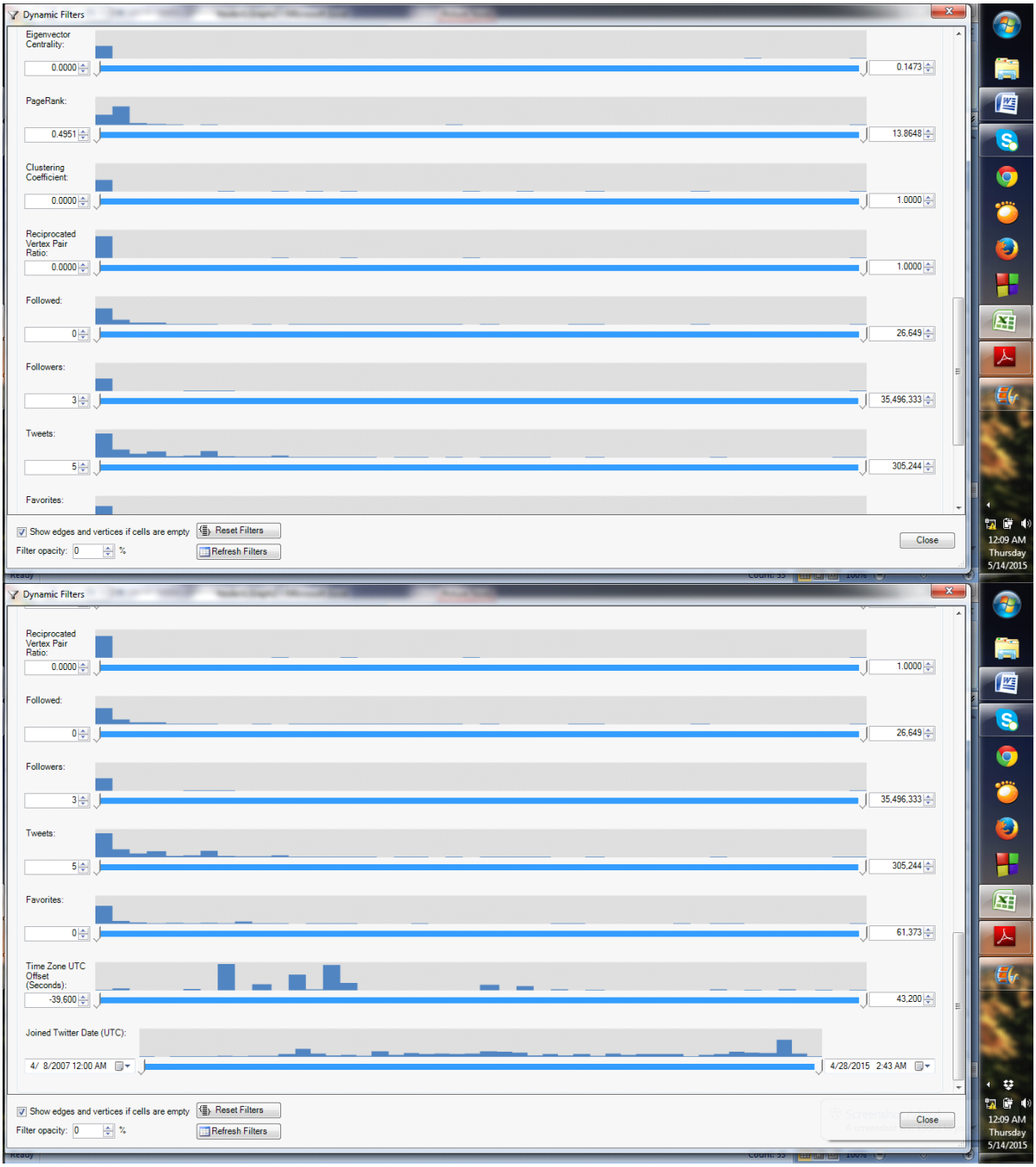

NodeXLGraph2 - Microsoft Excel
Table Tools
Vertices
Groups
Group Vertices
Overall Metrics
Group Edges
Twitter Search Ntwrk Top Items

Sugiyma



Dynamic Filter


Dynamic Filters
Edge Filters
Relationship Date (UTC):
5/11/2015 8:40 PM
5/14/2015 12:00 AM
Tweet Date (UTC):
5/11/2015 8:40 PM
5/14/2015 12:00 AM
Edge Weight:
1
16
Vertex Filters
Size:
1.50
10.00
Opacity:
10
100
Layout Order:
1.00
9,999.00
X:
514.00
9,484.17
Show edges and vertices if cells are empty
Reset Filters
Filter opacity: 0 %
Refresh Filters
Close
12:08 AM
Thursday
5/14/2015
Dynamic Filters
X:
514.00
9,484.17
Y:
4,664.80
5,348.64
In-Degree:
0
29
Out-Degree:
0
5
Betweenness Centrality:
0.0000
812.0000
Closeness Centrality:
0.0000
1.0000
Eigenvector Centrality:
0.0000
0.1473
Show edges and vertices if cells are empty
Reset Filters
Filter opacity: 0 %
Refresh Filters
Close
12:09 AM
Thursday
5/14/2015


Dynamic Filters
Eigenvector Centrality:
0.0000
0.1473
PageRank:
0.4951
13.8648
Clustering Coefficient:
0.0000
1.0000
Reciprocated Vertex Pair Ratio:
0.0000
1.0000
Followed:
0
26,649
Followers:
3
35,496,333
Tweets:
5
305,244
Favorites:
Show edges and vertices if cells are empty
Reset Filters
Filter opacity:
0
%
Refresh Filters
Close
12:09 AM
Thursday
5/14/2015
Dynamic Filters
Reciprocated Vertex Pair Ratio:
0.0000
1.0000
Followed:
0
26,649
Followers:
3
35,496,333
Tweets:
5
305,244
Favorites:
0
61,373
Time Zone UTC Offset (Seconds):
-39,600
43,200
Joined Twitter Date (UTC):
4/ 8/2007 12:00 AM
4/28/2015 2:43 AM
Show edges and vertices if cells are empty
Reset Filters
Filter opacity:
0
%
Refresh Filters
Close
12:09 AM
Thursday
5/14/2015

CHAPTER 6

Conclusion and Future Work

This work was taken up with an objective to propose an algorithm to extract and transform the seed words from social media effectively and efficiently and eliminating the redundancy in the tweets. In this respect it was essential to first study the different technologies used till now for twitter data analytics which gives cost effective methodologies to collect the keywords from open source distributed platform.

6.1 Conclusion

This project helped me understand the intricate process of Data Extraction, Transformation and Standardization. We could strengthen our understanding about Data Analytics and how we conclude the various issues on women's education and employment. The following proposed algorithm gives the result effectively and efficiently. The proposed code acquainted us with various such scenarios where minor mistakes would result in erroneous conclusions. The project also helps to scan and analyze social media feeds so that we can permenantly keep upto speed on the latest developments in the industry and its environment.

After immense toil and guidance, the algorithm was successfully realized and the extraction and statistical operations can now be smoothly executed within a matter of minutes. Further, we are going to implement this in real time environment.

6.2 Future Work

This includes

- User dependent threshold values.
- Updating the tweets every time from the twitter.
- Implementing with in-memory computing.
- Build it for another social media (Facebook, Youtube etc).
- Enable it for the real time query.
- To extend the limit of the number of tweets

References

[1] Smith and Shneiderman.”Analyzing social media with nodexl.” in University of Maryland . USA, 2011.

[2] Stephen G. Kobourov.” Spring Embedders and Force Directed Graph Drawing Algorithms.” in *University of Arizona*, pp. 4-9, 2012.

[3] Derek Hansen, Marc A. Smith, Ben Shneiderman.” EventGraphs: Charting Collections of Conference Connections.” in Connected Action Consulting Group, 2011.

[4] Derek Hansen, Ben Shneiderman.”Analyzing social media networks: learning by doing with Nodexl.” in University of Maryland, 2009.

[5] [Xindong Wu, Xingquan Zhu, Gong-Qing Wu, Wei Ding.]”Data Mining with Big Data,” *IEEE Ttransactions on Knowledge and Data engineering.,* vol 26, no 1, January 2014.

[6] Emilio Ferrara, Pasquale De Meo, Giacomo Fiumara, Robert Baumgartner.” Web Data Extraction, Applications and Techniques: A Survey.” july 2014.

[7] Adilmoujaheed “An Introduction to Text Mining using Twitter Streaming API and Python”,(adil moujahid) [online],2014.

[8] Wang Ling, Luís Marujo.” Crowdsourcing High-Quality Parallel Data Extraction from Twitter.”,2013.

[9] Shamanth Kumar, Fred Morstatter, Huan Liu.”Twitter Data Analytics”, in PROC Springer,2013