2016

Social Media Network Data Mining and Optimization

Neha Clare Jose
Louisiana State University and Agricultural and Mechanical College, neaclare@gmail.com

SOCIAL MEDIA NETWORK DATA MINING AND OPTIMIZATION

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Department of Engineering Science

by
Neha Clare Jose
B.Tech, Mahatma Gandhi University, 2013
August 2016

To my father and mother, Josekutty and Ancy, who believed in me and making me the person I am. My sister, Evangeline, who has always been my pillar and believed I could do anything.

ACKNOWLEDGEMENTS

This thesis is the result of not just my work but the hard work, love and support of the people in my life.

Foremost, I'd like to thank my faculty advisor, Dr. Gerald Baumgartner, for being the best advisor I could have asked for, for taking me in as his student and believing that I could be a contributor to his research, I would like thank him for always being available for my questions and always following up on my progress with my work, being flexible with my school schedule and being patient with me throughout this research.

My committee members, Dr. Michelle Livermore, whose Social Work initiative is the major reason behind this research, and her kindness and understanding made working with her a great experience. Dr. Jianhua Chen, for her support and knowledge about data mining that proved important for this research. Also, Andrii Dubytskyi, my colleague in this research, whose hard work was the foundation to this project.

I want to thank Dr. James J. Spivey, my employer, for funding me, the reason I could pursue my Master's degree at LSU and for being understanding towards my school work and my unavailability during the final days of this thesis.

My friends in Louisiana State University Danissa Rodriguez, Hari Perumal, Ibrahim Bashir, Joy Das, Karthik Chiluveru, Raju Marasini, Sonam Wadhwani, Utsav Agarwal and Vikram Gowrishankar who have been very instrumental with the completion of this thesis I cannot thank them enough for burning the midnight oil with me just so that I am not alone working, for answering my numerous questions and guiding me with this writing and coding, for being my family away from home.

Most importantly, my parents for letting me follow my dreams and letting me make this journey and believing that their girl child is capable of great things. My sister for believing in me and being my safe place.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii

ABSTRACT vi

1. INTRODUCTION 1
   1.1. SOCIAL CAPITAL AND SOCIAL NETWORKS 2
   1.2. INTERAGENCY NETWORK 3

2. TECHNICAL APPROACH 5
   2.1. BACKGROUND 5
   2.2. PROBLEM DESCRIPTION 9

3. CONCEPTS 17
   3.1. GRAPH 17
   3.2. NODE 18
   3.3. EDGE 18
   3.4. CENTRALITY 19
   3.5. CLUSTERING COEFFICIENT 20
   3.6. CUTSET 20

4. INVENTORY OF SNAP ALGORITHMS 22

5. SNAP INTERFACE WITH GRAPHML 24

6. KEYWORD SEARCH 26

7. FRAMEWORK TO ANALYZE USER RELATIONS 29
   7.1. ALGORITHM: MERGE GRAPH 31
   7.2. ALGORITHM: FIND ACTIVE USERS 31
   7.3. ALGORITHM: FIND CRUCIAL ORGANIZATION 33

8. EXPERIMENTAL RESULTS 34

9. SUMMARY 35

10. FUTURE WORK 37

11. REFERENCE 38

12. VITA 40

ABSTRACT

Many small social aid organizations could benefit from collaborating with other organizations on common causes, but may not have the necessary social relationships. We present a framework for a recommender system for the Louisiana Poverty Initiative that identifies member organizations with common causes and aims to forge connections between these organizations. Our framework employs a combination of graph and text analyses of the organizations' Facebook pages.

We use NodeXL, a plugin to Microsoft Excel, to download the Facebook graph and to interface with SNAP, the Stanford Network Analysis Platform, for calculating network measurements. Our framework extends NodeXL with algorithms that analyze the text found on the Facebook pages as well as the connections between organizations and individuals posting on those pages.

As a substitute for more complex text data mining, we use a simple keyword analysis for identifying the goals and initiatives of organizations. We present algorithms that combine this keyword analysis with graph analyses that compute connectivity measurements for both organizations and individuals. The results of these analyses can then be used to form a recommender system that suggests new network links between organizations and individuals to let them explore collaboration possibilities.

Our experiments on Facebook data from the Louisiana Poverty Initiative show that our framework will be able to collect the information necessary for building such a user-to-user recommender system.

1. INTRODUCTION

Louisiana is ranked 49th in the list of United States poverty rate. With poverty prevailing, we find the need for support and help for the people in the poverty sector of the state's population. There are numerous initiatives in the country and the state of Louisiana working towards eradicating poverty and supporting this section of the community. The importance of social relationships in the lives of the poor has been consistently documented. Poor individuals rely on members of their social networks to help them meet basic needs through the provision of food, shelter and money. Social relationships are also a focus of community interventions that aim to improve conditions in poor communities. This includes building relations among individuals so they can organize to act collectively or to produce needed goods and services for the community. Community development initiatives such as Comprehensive Community Initiative (CCI's), Empowerment zones and Enterprise Communities, attempt to "build community capacity" in low income communities.

The presence of various organizations in social media like Facebook, Twitter etc. can be used to bring these initiatives together through the online world. With the existence of small organizations the chances of them being aware of each other's presence and motives is limited. Organizations update and post their activities in the Facebook Fan page. Thus getting information from these posts and people who respond to the posts and their response in such a way that we get important content, and using this derived information to help the organizations with similar aims and initiatives to network with each other. Thus with this approach we aim at giving a chance to the smaller organization work

in bigger collaborations and this being more efficient through the data obtained from social media.

There are a lot of recommender systems existing in the market, like in Netflix where the recommender system recommends movies and TV shows to the user according to the viewing behavior , this is a user to product recommender system, another example is E-commerce websites like amazon that recommend products to user according to their search and buying trend. In this project the aim is to develop recommender system to connect the user to a user. It would be able to recommend various user, organization or person, to other users according to their posts and their Facebook network information.

For example let's assume there is a certain bill that is to be passed in the state legislature that affects the poverty ridden community of the state, consider a small non-profit organization A that is against the bill and is running a petition against this bill, another organization B is also doing the same and organization C is also against this bill and all three are voicing this on their Facebook pages respectively. Our recommender systems would gather information from all these organization and with text analysis find that all these organizations are talking about a same topic and then with analysis of their network the organizations would be recommended to talk to each other, this would give all three organizations to campaign the petition together, increasing the chances of larger support and thus the petition being considered.

1.1. SOCIAL CAPITAL AND SOCIAL NETWORKS

Research from Switzerland reflects the tendency of non-profit organizations to under-utilize the two-way communication potential of the Internet [9]. This project intends to utilize this two way communication for various poverty initiatives to network with each

other by collecting their data from Facebook pages and analyzing the posts and the users of the page. The aim is to bring data mining and graph analysis together to give an efficient recommender system that will recommend various organizations and users with common goals and initiatives to collaborate. In this project we have implemented the initial steps for the same. Firstly, a keyword search has been implemented. That allows to search certain keywords from the obtained data. Secondly algorithms have been implemented to analyze the obtained Facebook fan page data and finding the crucial organizations and the active users by calculating some network measurements. These algorithms can be further improved by including more measurements to get a more accurate a recommender system.

1.2. INTERAGENCY NETWORK

Research indicates that while community-wide initiatives involving the collaboration of multiple organizations in a local area have been effective in reducing numerous social problems and improving child development [5] such endeavors also face challenges and limitations [14, 4, 10]. Social Network analysis provides analytical tools to study the functioning of collaborative endeavors [7]. Network analysis has been used to enhance collaboration and coalition building efforts by revealing the structure of relationships among agencies [4, 16]. Prior research on interagency networks has found network range and cohesion are associated with collaboration and information sharing [12]. More recently, Cross et al. used network analysis to measure and improve interagency collaboration [8]. The evaluators used network graphs and measures to inform member agencies of the current status of relationships and to recommend changes in order to enhance collaboration.

An assessment of the structure of social relationships in a network can provide a baseline against which progress can be measured and guide intervention goals and strategies. Questions regarding the social structure of an interagency network often asked by organizers in the field include: How connected are the agencies to one another? Which ones are not connected? Which are the best organizations or individuals to spread information throughout the network.

2. TECHNICAL APPROACH

2.1. BACKGROUND

The research uses a plug-in that allows the developed algorithm to download data from Facebook pages and provide them in a format that can be visualized, the plug-in being used for this is called NodeXL which is a plug-in for Microsoft Excel, it allows the user to download data into excel sheets from Facebook pages and also provide a graph representation for the same. The backend code is written in the programming language C# and is open source. Thus, the algorithms for this thesis will be implemented in C#.

The output is obtained in a graph that consists of ‘nodes’ and ‘edges’. Nodes being the users who comment or like a post and the different posts. The edges connect the co-likers (users who like the same post), the co-commenter (users who commented on the same post), commenters who commented on the same post, likers who liked comments on the post, created the post (user and the post he created).

2.1.1 Facebook

As of the fourth quarter of 2015, the number of monthly active Facebook users in North America and Canada is 219 million (Figure 1), the monthly active users are the users who have logged into Facebook in the past 30 days. Facebook in its short term of existence has found a worldwide user base. This user base has led to people connecting with friends initially, then as the time passes people started connecting with a common interest that led to the introductions of groups, the came pages where organization, celebrity, business could create pages and have people follow them for latest updates and contacting them.

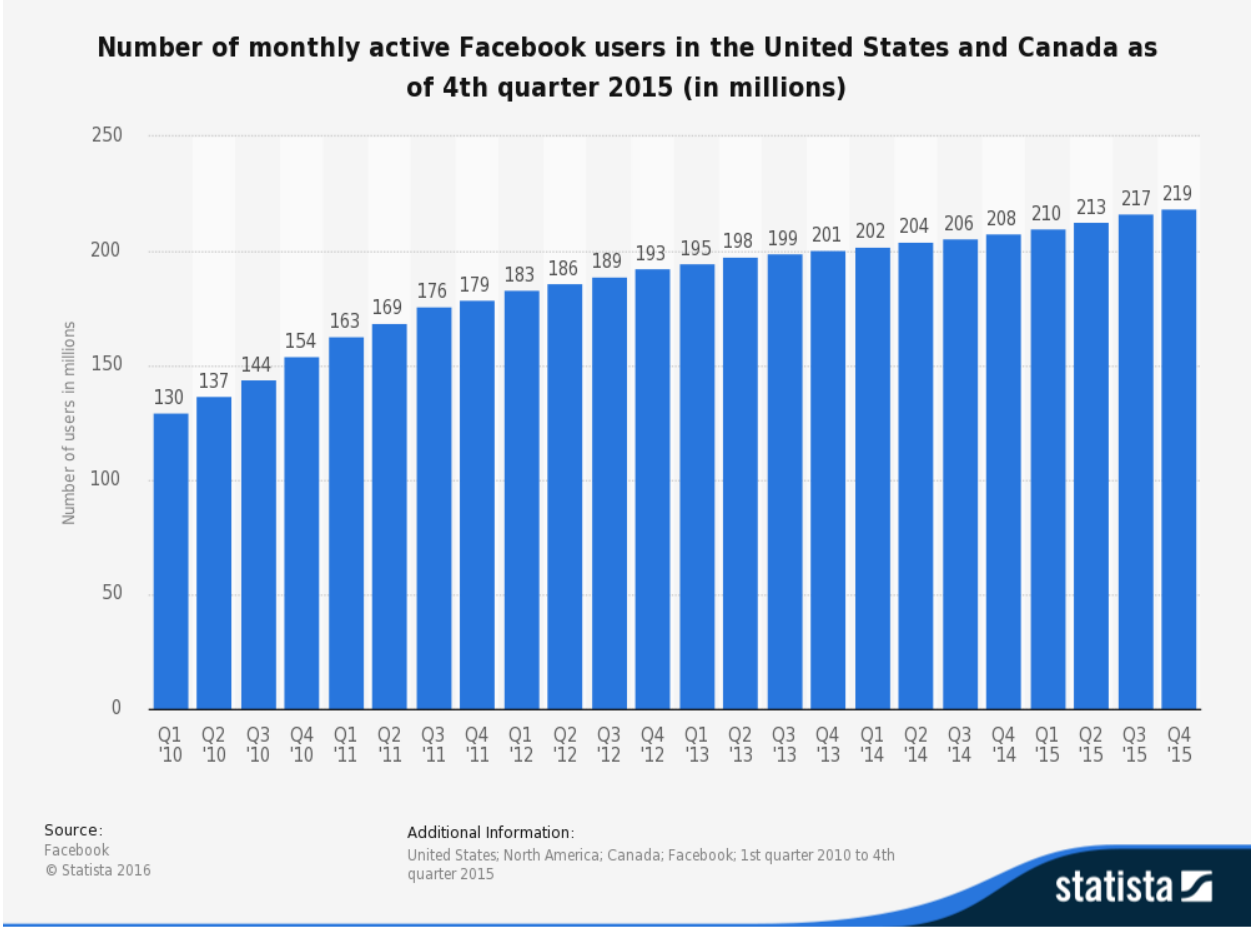


Figure 1: Facebook users in North America as of fourth quarter 2015 (Source: http://www.statista.com/statistics/247614/number-of-monthly-active-facebook-users-worldwide/, accessed 4/12/2015)

This aspect of Facebook is what is going to be used in this project, the pages created by different organizations in Louisiana that work towards different initiatives, our concentration being on the Louisiana poverty initiative, the pages will let us obtain information from different organizations, the posts from them and the users or other pages that comment on the post. This information obtained from Facebook will help us find organizations and people working towards the same cause or initiatives. Thus, we will be able to find the crucial and the right contacts that would want to talk to each other for more effective collaborations and effective impact on in the initiatives.

2.1.2. NodeXL

With the information on Facebook, the next thing required is an interface that would let us download the data from various pages on Facebook and also allow to visualize the same to understand the nature and behavior of the network we are considering that would help us choose the most optimal method to achieve the aim of this project.

The interface that we decided to use is NodeXL, which is a plug-in for Microsoft excel, that allows to download data from various social media. By provider definition, NodeXL is a free, open-source template for Microsoft® Excel® 2007, 2010, 2013 and 2016 that makes it easy to explore network graphs. With NodeXL, you can enter a network edge list in a worksheet, click a button and see your graph, all in the familiar environment of the Excel window [1].

This NodeXL recently revised its licensing and split NodeXL into two versions one being a free version and the other being the paid version which had additional features and analysis algorithms implemented.

The free version is the version is called the NodeXL Basic and the paid version is NodeXL Pro. NodeXL Pro offers additional features that extend NodeXL Basic, providing easy access to social media network data streams, advanced network metrics, and text and sentiment analysis, and powerful report generation. NodeXL Pro can create insights into social media streams with just a few clicks.

The version used in the project was the one before the revision and thus has features from both the Basic and Pro version. NodeXL is now used to teach SNA in dozens of classes around the globe. The ability to automatically capture data was essential for non-programmers, as were the layout algorithms [1].

NodeXL uses a highly structured workbook template that includes multiple worksheets to store all the information needed to represent a network graph. Network relationships (i.e., graph edges) are represented as an ‘edge list’, which contains all pairs of entities that are connected to the network. Complementary worksheets contain information about each vertex and cluster. Data importers allow users to grab networks and user data from popular social media networks such as Twitter, YouTube, Flickr, and email [6]. Add what data is downloaded from NodeXL ego group and fan page.

NodeXL uses various features that makes it ideal for the purpose of the project. The two major features are explained in the following sections.

2.1.2.1. GraphML. GraphML is a comprehensive and easy-to-use file format for graphs. It consists of a language core to describe the structural properties of a graph and a flexible extension mechanism to add application-specific data. Its main features include support of

- directed, undirected, and mixed graphs,
- hypergraphs,
- hierarchical graphs,
- graphical representations,
- references to external data,
- application-specific attribute data, and
- light-weight parsers.

Unlike many other file formats for graphs, GraphML does not use a custom syntax. Instead, it is based on XML and hence ideally suited as a common denominator for all kinds of services generating, archiving, or processing graphs.

Work on GraphML was initiated in a workshop during the 2000 Graph Drawing Symposium in Williamsburg, and a proposal for the structural layer was presented at the 2001 Graph Drawing Symposium in Vienna.

Since then, extensions have been provided that support basic attribute data types and the embedding of information for light-weight parsers. The next major steps will be extensions for abstract graph layout information and templates to transform such information into a variety of graphics formats [2].

2.1.2.2. SNAP. Stanford Network Analysis Platform (SNAP) is a general purpose, high-performance system for analysis and manipulation of large networks. Graphs consist of nodes and directed/undirected/multiple edges between the graph nodes. Networks are graphs with data on nodes and/or edges of the network.

SNAP was originally developed by Jure Leskovec in the course of his Ph.D. studies. The first release was made available in Nov 2009. THE SNAP library is implemented in C++, it has effective algorithms and procedures that allow us to understand large graphs and networks, and it allows calculations, graph generation. Thus, it allows to manipulate the data we obtain in nodes and edges and even allows dynamically changes to this information during computation [9]. SNAP allows us to work with large graphs, this proves to be important especially when dealing with huge amount of data that is usually related to social network analysis. The algorithms present in SNAP will be explained later in Section 4.

2.2. PROBLEM DESCRIPTION

In the project, we have written codes to deal with different steps to achieving our final goal to be able to visually analyze Facebook Fan Page data, to be able to help different people,

organizations to collaborate for different kinds of Louisiana poverty initiative. There are various aspects of this analysis that we have tried achieving in this thesis. The major aspects that are dealt with in this thesis are:

a) An algorithm that picks data from Facebook Fan pages that are entered manually or from an excel sheet using their Facebook page id, the Fan page downloaded is downloaded in XML format using GraphML, this XML file is the manipulated in the backend to achieve rest of the algorithms.

b) The downloaded graphs are required to be analyzed together, thus the next step is to merge the graph, this is done using an algorithm that allows user and allows the user to merge the two graphs. The merging keeps all the nodes and the edges that are common to both the graph and then adds a copy of any node that is not common and does the same with edges.

c) A data structure that lets you keyword search for certain keywords so that posts with those keywords can be extracted with the user who said it.

d) An algorithm that calculates various factors between each node, this part of the algorithm is what the research is currently dealing with.

e) Once the data has all been filtered, a tool called SNAP with interfacing with the GraphML file format will be used to visualize the data that will make it easier for the user to see the relations and decide which organizations or initiatives have the possibility of working together. This section of the research is going to be a part of the future work of this thesis.

2.2.1. Authenticating the User

Before being allowed to download data through NodeXL the user needs to authenticate themselves. This achieved by using an access token When using the NodeXL Plug-in GUI the Facebook login that is requested before the page download is the process that fetches the user access token and authenticates the user to download the page. In the backend, this is rather not as convenient but the access code has to manually fetch and entered to authenticate before downloading the graph. The access token is obtained using Facebook Graph API. The link to the Facebook Graph API is https://developers.facebook.com/tools/explorer. Figure2 shows how the Graph API looks.

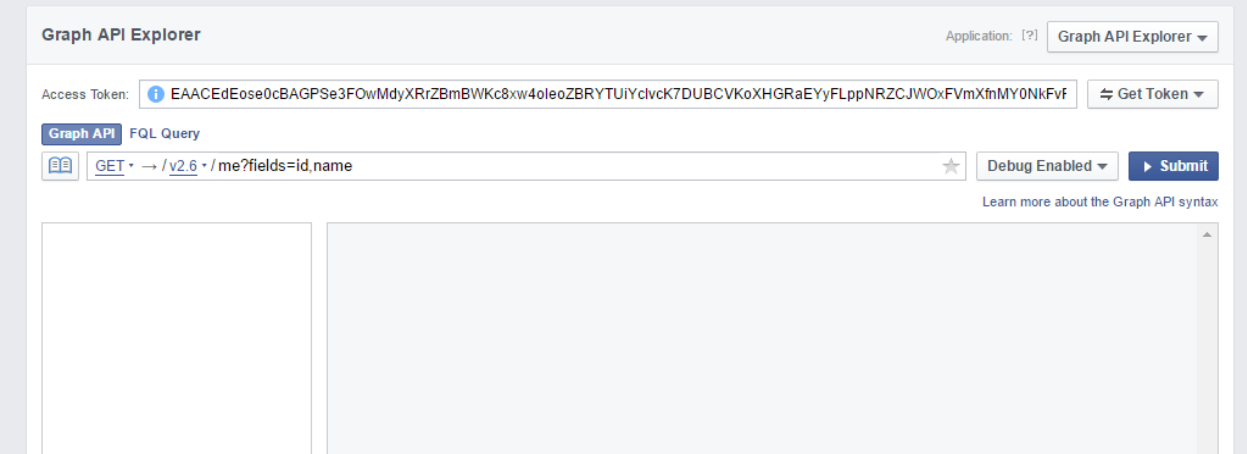


Figure 2: Facebook Page for Graph API.

The access token obtained through the Graph API has an expiry time thus needs to be obtained periodically and authenticated if downloading Fan Page Data.

2.2.2. Download Graph

In this section, we are going to see how a graph is downloaded using NodeXL as an excel plugin and also how the graph is downloaded in the backend and how the result of the download looks.

As explained in earlier sections NodeXL is a plugin from Microsoft excel that allows us to download social media data as an excel sheet as well as presents us with a graph representation of the downloaded data for visualizing.

The following are the steps to download Fan Page data into Microsoft excel with the pictorial representation for better understanding.

a. Choose the option to download from Facebook page from the “File”-> “Import“ tab. The option available are shown in Figure 3.

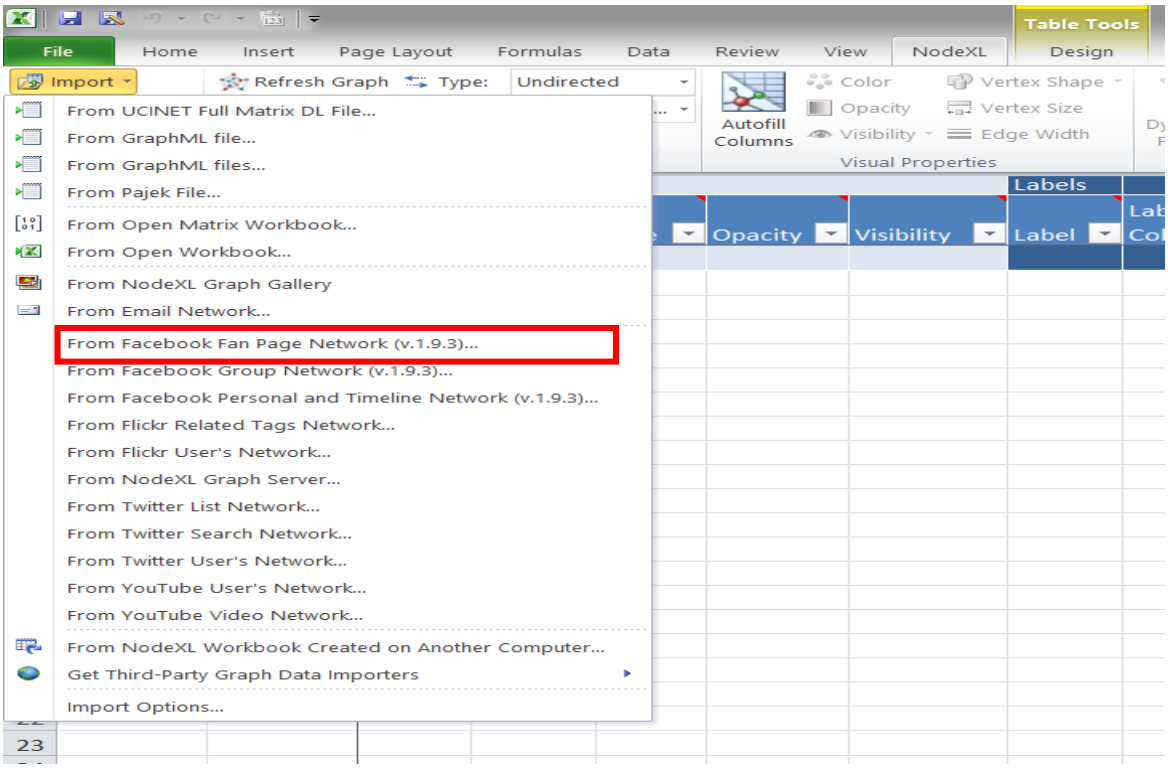


Figure 3: NodeXL options in Microsoft Excel (Source: http://simplysocialscience.blogspot.com/ 2014/08/nodexl-tutorial-part-1.html)

NodeXL allows user to download Fan Page Network, Group Network and Person and Timeline Network from Facebook, for the purpose of thesis download of Fan page is important. This we choose the option to do the same. The selection would lead to a pop-up window “Import from Facebook Fan Page Network”. The pop up window gives options as shown in Figure 4.

b. Select the page whose information is to be downloaded and filter the information to be downloaded, example: attributes of the page, various nework option and the options for downloading the post.

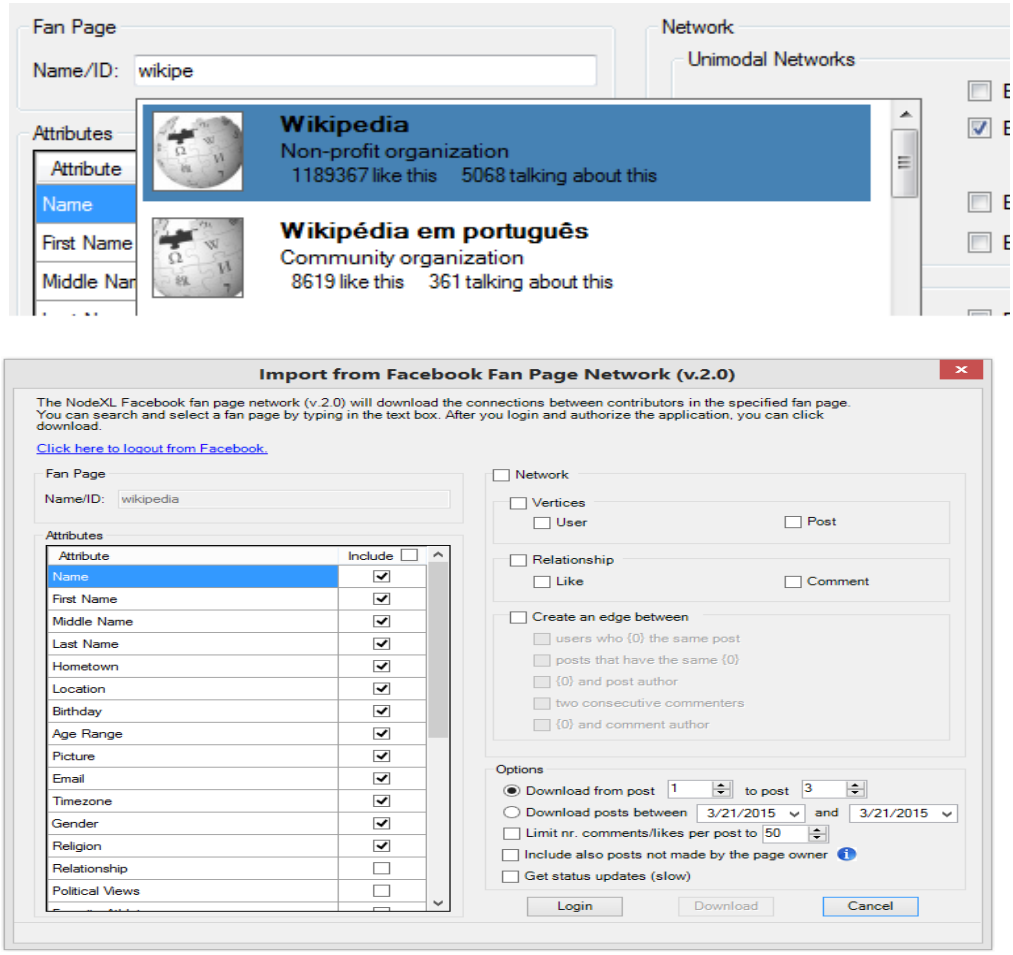


Figure 4: Window for NodeXL Fan Page Download (http: //www.smrfoundation.org/wp-content/uploads/2015/03/)

c. The output obtained would look as shown in Figure 5 both in data and graphical format.

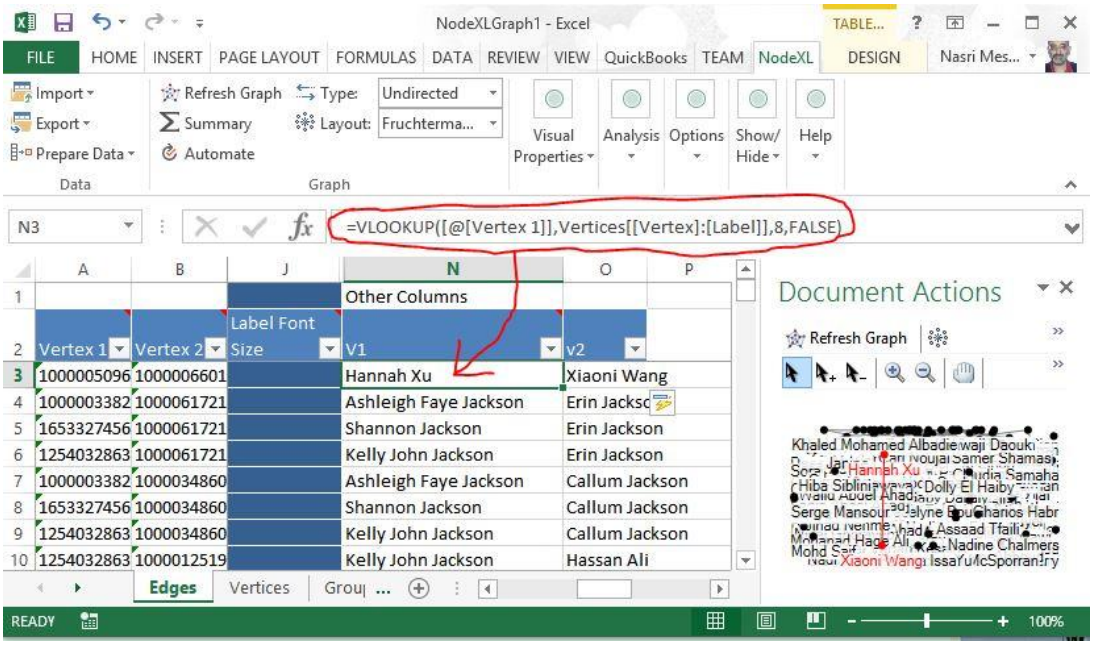


Figure 5: The Output obtained on FanPage Download using NodeXL as plugin (Source: http:// nasri.messarra.com/facebook- friends- social-graph-using-netvizz-and-nodexl/ accessed on 4/28/2016)

For downloading data in the backend using the NodeXL library in C# the graph can be download command, for reference let us consider the fan page for Capital Area Unit Way, the command will be:

download -id capitalareaunitedway -pl 100 -po 0 -cl 100 –f C:\...\graph1.xml

The output will be stored in the specified location under the name “graph1”, the file is GraphML format:

```
<?xml version="1.0" encoding="UTF-8"?>
    <graphml xmlns="http://graphml.graphdrawing.org/xmlns">
        <key id="Relationship" for="edge" attr.name="Relationship"
        attr.type="string" />
        <key id="MenuText" for="node" attr.name="Custom Menu Item Text"
        attr.type="string" />
        <key id="MenuAction" for="node" attr.name="Custom Menu Item
        Action" attr.type="string" />
```

```
        <key id="Image" for="node" attr.name="Image File"
        attr.type="string" />
        <key id="name" for="node" attr.name="Name" attr.type="string" />
    .
    .
    .
```

The format of the GraphML file obtained will be explained in Section 3. Concepts.

2.2.3. Merge

In the project the next objective is to merge graphs so that all the organizations we need to analyze that would help up realize the relations and their measurements. The merge happens between two graphs and the output graph obtained has a GraphML format with the xml tags different from the ones that were obtained during graph download.

To merge the graphs the graphs first need to be loaded into memory, and then merged.

To load the graph the memory the following command needs to run:

load C:\...\graph1.xml -as graph1

This code loads the graphML file “graph3” to the memory under the name “graph1”.

Once the graphs that are going to be merged are loaded into the memory the next obvious step is to merge them.

The command used for graph merging is as follows

merge -g graph1,graph2 -as graph3

This would merge “graph1” and “graph2” residing in the memory as “graph3”.

The next step is to save the merged graph into the system for further use and reference.

The command used is:

save graph3 -f C:\..\..\mergedgraph1.xml

This command saves the file with the specified filename (“graph3” in the above example) into the specified location under the specified name (“C:\..\mergedgraph1.xml”).

2.2.4. Keyword Search and Calculate Measures

The Keyword search and measurement calculation algorithms are the foundation of the network analysis for this project. These serve as the basis to analyzing the social network graphs of different organizations and finding the initiatives under them thus them being the major aspect of this thesis and are thus covered separately in Chapter 6 and Chapter 7 respectively.

3. CONCEPTS

3.1. GRAPH

A graph is a visual representation of data in the form of vertices and edges, vertices of a graph are the primary element, they represent the units and the relations between two vertices represented using lines called edges.

Graphs can be simple graphs (graphs that do not have multiple edges or self-loops), directed graphs (graphs with edges that have direction, usually represented with an arrow towards the direction) or weighted graphs (graphs with weighted edges).

In NodeXL, the graph is a directional graph. Thus, at the edges, it has a source node and a target node between which the relations are defined. The graph is the Fan Page with all the posts till the specified limit during download and specified the number of comments per post. Figure 6 gives a basic pictorial representation of the nodes and the relations between them that make an edge.

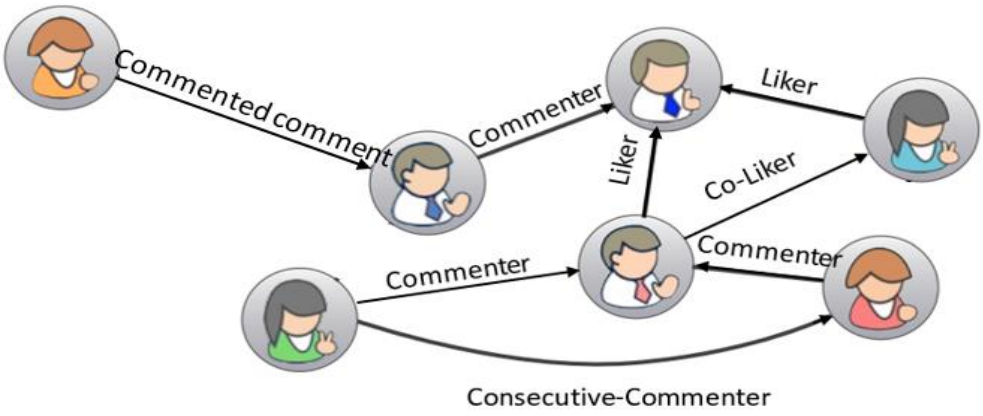


Figure 6: Representation of how NodeXL stores the graph

3.2. NODE

The vertices are the nodes which are the users and the post. The user can be either a commenter or a liker. A commenter being a person who comments on a post and a liker being a person who like a comment or a post.

Here is an example of a node in the graph file obtained

<node id="Alice Xyz">
<data key="V-Custom Menu Item Text">Open Facebook Page for This User</data>
<data key="V-Custom Menu Item Action">https://www.facebook.com/*35</data>**
<data key="V-Name">Alice</data>
<data key="V-First Name">Janet</data>
<data key="V-Middle Name"> </data>
<data key="V-Last Name">Xys</data>
<data key="V-Vertex Type">User</data>
<data key="V-User Type">Commenter</data>
<data key="V-Tweet">Thank you ABC organization for funding for LMN Ministries!!</data>
<data key="V-Tooltip">Alice XyzThank you ABC organization for funding for LMN Ministries!!<data>
<data key="V-Label"> Alice XyzThank you ABC organization for funding for LMN Ministries!!</data>
<data key="V-Likes Received">0</data>
<data key="V-Likes Created">2</data>
<data key="V-Comments Received">0</data>
<data key="V-Comments Created">1</data>
</node>

3.3. EDGE

The edges connect these users. The edges can be between (refer to Figure 6):

- Co-liker: two user nodes that liked the same post.
- Co-commenter: users who comment on the same post.
- Liker: liker and post author.
- Commenter: commenter and the post author.

- Consecutive commenter: two user who have consecutively commented on the same post or comment
- Created post: the creator of the post and the page in which the post was created.
- Liked comment: user who liked the comment and the commenter
- Commented comment: user who commented on a comment and parent commenter.

The edges tag obtained after the merge is as follows.

```
<edge source="Alice" target="Bob">
    <data key="e_type">User Liked Same Post</data>
    <data key="relationship">Co-Liker</data>
        <data key="post_content">for more than 90 years, we've led the way by funding hundreds of nonprofit agencies across the 10 parishes we serve. This year we're taking a new approach, one that will allow us to achieve more and have a greater impact on our community's most pressing needs. We call this approach the impact model. Learn more: </data>
        <data
        key="post_url">https://www.facebook.com/268187***68_124902231
        5****100</data>
 </edge>
```

3.4. CENTRALITY

Centrality is the concept that is the foundation for all the algorithms developed and to be developed in this thesis, centrality is an important term in graph analysis. Networks involve diffusion of information from one node to the other, some of which may be more important than others There are different measures of centrality that help us define or learn different aspect of a network graph [3].

There are different measures of centralities used for analysis some are discussed below:

a) Degree Centrality: It is defined formally as "The number of links incident upon a node". The degree is often considered as a means of analyzing how nodes can be

affected by flow inside a given network. Often links are associated with friendships — in-degree as a measure of being popular and out-degree as a metric for being gregarious [3].

b) Betweenness Centrality: Betweenness centrality quantifies “the number of times a node acts as a bridge along the shortest path between two other nodes”. Freeman notes that “vertices that have a high probability to occur on a randomly chosen shortest path between two randomly chosen vertices also tend to have a high betweenness” [3].

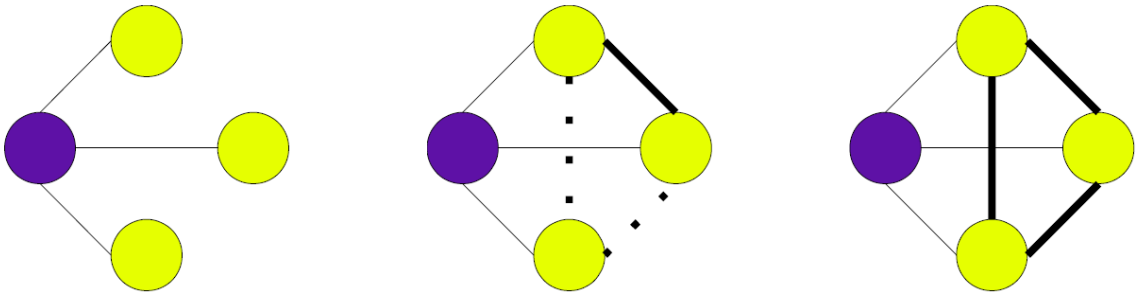
c) Closeness Centrality: Connected graphs often require a metric for the distance between node pairs — defined subsequently in the form of “length of shortest paths”. The farness of a node is formally defined as “the sum of its distances to all other nodes”, and its closeness is defined as “the inverse of the farness”. Thus, the lesser would be its total distance from other nodes, the more central a particular node will be [3].

3.5. CLUSTERING COEFFICIENT

It can be defined as the proportion friends who are also friends with each other. The clustering can be defined using the term triad: A triangle consists of three closed triplets, one centered on each of the nodes. The global clustering coefficient or triad is the number of closed triplets (or *3 x triangles*) over the total number of triplets (both open and closed) [13] (Figure 7).

3.6. CUTSET

A set of edges of a graph which, if removed (or "cut"), disconnects the graph (i.e., forms a disconnected graph) [15].



(a) No pairs formed among neighbors: $C = 0$

(b) One pair formed among neighbors: $C = 1 / 3$

(c) Three pairs formed among neighbors: $C = 3 / 3$

Figure 7: How to calculate Clustering coefficient (source: http:// recibe.cucei.udg.mx/revista/es/vol1-no1/computacion02.html, accessed on 5/11/2016

NodeXL provides algorithms to calculate these measures in a GraphML graph, in this research we try to find the optimal algorithm that would help us find 2 aspects of the graph

a. User graph that has all the users who post on the same page.
b. An algorithm to find a relation between two organizations to understand the possibility of the two organizations collaborating with each other.

Centrality measure that is apt for this research thesis is gregariousness centrality and the Clustering Coefficient. In addition to this measurement cutset is an interesting implementation of the graph, the cutset can help us realize the most important node and user in the network removing who can disconnect the graph thus we can obtain the most important connections in the network, that may also prove to be crucial point of contact for various collaborations and events. The implementation of these measurements and concepts will be explained later in Section 7.

4. INVENTORY OF SNAP ALGORITHMS

SNAP provides various procedures that can be performed on the downloaded data from NodeXL. This section we look at some of the important algorithms provided by SNAP that we use for this thesis or can be used in future algorithm implementation.

Figure 8 gives a brief overview of all the algorithms present in SNAP both being developed and the ones completed. The algorithms that crucial to this research reside in the header files:

a) The centr.h file that includes the calculation algorithms that are used or will be used for the project. The header file has algorithms for the calculation of different centrality measurements like the gregariousness calculation, the cluster coefficient, and the degree centrality. The usage of these measurements are described in Section 5.
b) The gio.h file has various adaptor algorithms that are used for the GraphML format of the graph obtained, it helps input the graph in the textual XML format. Also, the files can be saved in the formats. It not only supports the GraphML format but also other formats like Pajek, OJA.
c) Alg.h consists of basic algorithms for manipulating elementary parts of the graph lie the nodes and the edges and various functions performed on them
d) Triad.h this is for the clustering coefficient related calculations, it counts the triads and also measurement itself.

snap-core	
alg.h	Simple algorithms like counting node degrees, simple graph manipulation (adding/deleting self edges, deleting isolated nodes) and testing whether graph is a tree or a star.
anf.h	Approximate Neighborhood Function: linear time algorithm to approximately calculate the diameter of massive graphs.
bfsdfs.h	Algorithms based on Breath First Search (BFS) and Depth First Search (DFS): shortest paths, spanning trees, graph diameter, and similar.
bignet.h	Memory efficient implementation of a network with data on nodes. Use when working with very large networks.
centr.h	Node centrality measures: closeness, betweenness, PageRank, ...
cmty.h	Algorithms for network community detection: Modularity, Girvan-Newman, Clauset-Newman-Moore.
cncom.h	Connected components: weakly, strongly and biconnected components, articular nodes and bridge edges.
ff.h	Forest Fire model for generating networks that densify and have shrinking diameters.
flow.h	Maximum flow algorithms.
gbase.h	Defines flags that are used to identify functionality of graphs.
ggen.h	Various graph generators: random graphs, copying model, preferential attachment, RMAT, configuration model, Small world model.
ghash.h	Hash table with directed graphs (TNGraph) as keys. Uses efficient adaptive approximate graph isomorphism testing to scale to large graphs. Useful when one wants to count frequencies of various small subgraphs or cascades.
gio.h	Graph input output. Methods for loading and saving various textual and XML based graph formats: Pajek, ORA, DynNet, GraphML (GML), Matlab.
graph.h	Implements graph types TUNGraph, TNGraph and TNEGraph.
gstat.h	Computes many structural properties of static and evolving networks.
gsvd.h	Eigen and singular value decomposition of graph adjacency matrix.
gviz.h	Interface to Graphviz.
kcore.h	K-core decomposition of networks.
network.h	Implements network types TNodeNet, TNodeEDatNet and TNodeEdgeNet.
Snap.h	Main include file of the library.
statplot.h	Plots of various structural network properties: clustering, degrees, diameter, spectrum, connected components.
subgraph.h	Extracting subgraphs and converting between different graph/network types.
timenet.h	Temporally evolving networks.
triad.h	Functions for counting triads (triples of connected nodes in the network) and computing clustering coefficient.
util.h	Utilities to manipulate PDFs, CDFs and CCDFs. Quick and dirty string manipulation, URL and domain manipulation routines.

Figure 8: Brief description of SNAP functionality (source: http://snap.stanford.edu /snap/description.html, accessed on 4/15/2016)

5. SNAP INTERFACE WITH GRAPHML

The NodeXL provides a graphical front–end with various functions that help to interface the GraphML to SNAP, it allows us to perform various actions on the GraphML as well as implement the usage of SNAP in them. In the backend the library NodeXLNetworks.cs provides us with the important functions that can be tweaked according to our needs. Some of the functionalities that provided in the interface are:

- Graph Metric Calculations:

  NodeXL can easily calculate basic network metrics like degree, adds calculation of betweenness centrality, closeness centrality, clustering coefficient with the help of the SNAP algorithms mentioned in the previous chapter. There is the presence of the SNAP calculator which is an application that calculates various centrality according to the centrality algorithms.

- Flexible Import and Export:

  NodeXL imports graphs from GraphML, Pajek, UCINet, and matrix formats. For our research, we use GraphML import (Appendix A, Section A.1). The graph downloaded is XML format that is converted into GraphML format and allows to manipulate the graphs obtained in the format.

- Direct Connections to Social Networks:

  NodeXL allows for import of network data from Twitter, Facebook, Flickr, and YouTube, there are several plugins to get network information from other social media as well. In the case of this thesis the social media network we are interested in is Facebook, for Facebook, it allows us to download Facebook FanPage, User Ego

(User Personal Page), Facebook GroupPage and for download it also provides authentication function to make sure the user has allowed access to the data.

- Basic Manipulation tool:

  The interface provides with manipulation functions for the graphs, there are functions that allow traversal through vertices, traversal through the edges, and one crucial function is the merge function that has been mentioned in the previous sections. The merge happens on the first graph, a copy of the first graph is made to which the second is merged the merge is done. It checks if the vertex is already existing if it does its skipped else the vertex is cloned and added to the first graph. Then the edges between the vertices are added.

- Analysis functions

  Functions that help analyze the data in the graph, we can find the list of common vertices in two graphs, and the algorithm is similar to the node merging of the merge function. Another function available gives the count of a total number of nodes and graphs, these to functions can be very useful for further analysis on graphs that can be done in this project.

- Task Automation:

  Till now we were talking all the interfaces in the back end of NodeXL, all the algorithms present interface all the calculations, analysis from SNAP with the graph obtained as a result of download of a Facebook page. In front end they can perform a set of repeated tasks with a single click e.g. NodeXL>Graph>Automate>Run executes all the steps needed to process a network data set from start to finished, published report.

6. KEYWORD SEARCH

Before implementing the algorithm for the network analysis we decided to do the first simple but efficient way of getting required information from the graph. Thus, the Keyword search has been implemented in the project. The data obtained from the merged graph is stored in another data structure, this data structure would allow the user to search for certain words e.g. “blood drive” and return all the posts and comments containing the keyword and the user or page that posted it.

In this code there are three functions used for the keyword search reads every node from the merged graph and separates, different XML tags, to obtain the comments or post contents, the user or Fan Page linked to them and the link to them.

The obtained information is then sent to a function that takes the comments and posts as input and then performs basic data cleaning on them, and then create a new data structure a hash table with all the important words in the sentence as the key for the hash table and the username and the link to the user’s profile.

The third functionality is where the keyword search is implemented, where the user inputs the keyword they want to search, hash tables allow easy and optimal search algorithms. The data structure used is a ‘Dictionary’, which is a hash table implementation in C#. The dictionary as mentioned earlier would have the keywords from the comments and post. The values linked to each keyword is then accessible with a simple search and we obtain the all the users and pages that have mentioned the keyword in their post or comments. With obtaining the user and the link to their profile we also need the post or comment itself that had the keyword present since not all of them contain the keyword in the manner we want, thus knowing the content to understand the context of the said keyword is

understood and data is accordingly pulled out. Let us consider the example mentioned earlier in the paper:

```
<node id="Alice Xyz">
            <data key="V-Custom Menu Item Text">Open Facebook Page for
            This User</data>
            <data key="V-Custom Menu Item
            Action">https://www.facebook.com/***35</data>
        <data key="V-Name">Alice</data>
        <data key="V-First Name">Janet</data>
        <data key="V-Middle Name"> </data>
        <data key="V-Last Name">Xys</data>
        <data key="V-Vertex Type">User</data>
        <data key="V-User Type">Commenter</data>
            <data key="V-Tweet">Thank you ABC organization for funding for
            LMN Ministries!!</data>
            <data key="V-Tooltip">Alica XyzThank you ABC organization for
            funding for LMN Ministries!!<data>
            <data key="V-Label"> Alica XyzThank you ABC organization for
            funding for LMN Ministries!!</data>
        <data key="V-Likes Received">0</data>
        <data key="V-Likes Created">2</data>
        <data key="V-Comments Received">0</data>
        <data key="V-Comments Created">1</data>
    </node>
```

According to the algorithm the comment Thank You ABC organization for funding for LMN Ministries will be read and then cleaned to store each word like ABC, Organisation, LMN to be stored as the key for the dictionary and the username Alice Xyz, the link to her Facebook page (https://www.facebook.com/***035) and the entire comment will be stored in the dictionary. And thus, when the users search for say LMN they will obtain all including Alice's comment that had the keyword LMN in it. And thus, the user can read the comments and decide which comment they want to do anything about the information they obtained i.e. who they would like to contact and what comments are not needed for them. This algorithm cannot only be used to get the right contacts for collaborations but also for understanding responses for example to understand the response to a certain

event or drive that has already been held or is going to be held. Another trend observed currently with social media users is the use of hashtags. This is harder to search on but the algorithm allows the search for hashtags as well, as hashtags are very useful for trending or marketing things in the current social media scenario.

This is a primary form of text mining, where a very basic search algorithm from textual data is obtained. This algorithm may be taken forward to an extensive data mining algorithm for analysis of the comment and prediction of the possible collaborations and social networking accordingly, but for the purpose of this research, we stick to this basic search algorithm for now and would mention the rest in Future work for the project.

7. FRAMEWORK TO ANALYZE USER RELATIONS

For the intended recommender system the framework consists of combining text mining and graph analysis to recommend users to other users on the basis of common goals and motives towards poverty initiatives. Different from existing user to products recommender systems, our recommender system aims to connect users to other users according to what they post, comment or like on Facebook. SNAP provides us various measurement functions but does not differentiate a user between whether the user is a person or an organization, it only performs calculations in the graph. We separate users from organization in our algorithms and perform various measurements on the data after separating, there are different set of measurements that need to be implemented on the two kinds users, and thus separating them is important for the purpose of this research.

In this research we would download Facebook fan page in a graph format, the downloaded graphs are then merged into one single graph, and text mining will be performed on this merged graph to obtain interesting data, important keywords, initiatives, events etc. these interesting texts will then be extracted with its related nodes and then various measurements will run on with the obtained data to find an ideal recommender algorithm that would recommend important and crucial users both organization and people that may would want to network and collaborate as they have similar goals.

Example, consider the scenario mentioned in the Introduction of this thesis, organizations A, B and C are all against a bill X being passed and A and B are running petitions against the bill and C is voicing their support against the bill, on implementation of the recommender system, the program will recognize this bill that is mentioned in the pages of these organization, and it will find the organizations A , B and C in the network are all

against the same bill and thus will be recommended to be talking or networking among each other so that there is a possible collaboration between them. It will also recommend other people who have spoken against the bill in their Facebook Fan page. Facebook currently has recommender system that recommends a particular person one may know on the basis of mutual friends, our recommender system intends to do something similar by recommending on the basis of content and mutual connection together.

Procedure: Main Program (cmd, args)
Let

- *The command be cmd.*
- *The arguments for the command be args*

Input: Command for action on graph

```
START
READ cmd + args
IF cmd = download THEN
        Fanpage = args
        Download graph
ELSE IF cmd = merge THEN
        Graph1=NULL
        mergedGraph=NULL
        FOR all i from 1 to graphnames[count] in args
                graph2= graphnames[i]
                graph1= Merge(graph1, graph2)
        mergedGraph=APPEND(graph1);
        SAVE graph1
ELSE IF cmd = userGraph THEN
        userGraph= CreateuserGraph (args)
        SAVE userGraph
ELSE IF cmd= fcrudialOrg THEN
        crucialOrg= gindCrucialOrg(args)
        SAVE crucialOrg
STOP
```

In the following sections, we discuss the algorithms that perform merging on multiple graphs, and then run different measurements on the merged graphs to obtain meaningful results, that allows us to understand the data obtained better and thus to help us achieve the final result of this research.

7.1. ALGORITHM: MERGE GRAPH

This algorithm merges two graphs as mentioned in section 2.2.3. In the merge function a copy of the first graph is made. For every node in graph2 we check the existence of the node on graph1 if the node already exists we move on to the next node, if the node does not exist we create the node in the copy of graph1. After all the nodes are traversed for all the edges in graph 2 we find the vertices and add the edge for the vertices in the copy of graph1. And the resulting graph is the merged graph.

```
Procedure: Merge (graph1, graph2)
Input: graph graph1, graph graph2
Output: mergedGraph
    START
    mergedGraph=graph1
    FOR node IN all nodes (graph2)
            if node not in ( mergedGraph) then
                    nodeClone= CLONE(node)
                    mergedGraph.ADD(nodeClone)
                    next node
            else
                    next node
    for all edge in all edges (graph2)
            node1 =  sourceNode(edge)
            node2 =  targetNode(edge)
            edgeClone=CLONE(edge(node1,node2)
            mergedGraph. ADD(edgeClone)
    RETURN mergedGraph
```

7.2. ALGORITHM: FIND ACTIVE USERS

In this algorithm we first open the merged graph, as discussed earlier, in the graph there are two kinds of nodes defined at the XML tag *Vertex type* one is the user and the other is post. In this algorithm our aim is to filter out the users alone, thus the first step is to check if the node type is user. Under user type the node can either be a person or a page profile. They can be differentiated by the XML tag:

<data key="V-Image File"> </data>

The above tag is noted to be missing in user nodes that are associated with people rather than pages. Thus we filter out the nodes that do not have the tag.

Since our first aim is to find all the users that are most connected in the network graph we find the clustering coefficient of all the vertices and take the ones above a threshold value which can be manually decided, for testing purposes we used the value as .5, from these nodes we found the gregariousness of the nodes so that we know the most active users who post or comment on posts. These users if they network can help collaborate at a bigger and successful stage.

The next step to this algorithm should be adding edges to these nodes, such that the nodes that have commented on the same post or posted on the page share an edge as to understand and analyze their connectivity at a larger scale. This step is still at the development stage.

```
Procedure: createUserGraph (graph, threshold)
Input: graph, double threshold
Output graph userGraph
START
ClusteringCoeffecient css= SNAP->CalculateClusterCoeffecient(graph)
Table degree(node, degree)= SNAP-> OutDegree(graph)
Declare userGraph
FOR node IN all nodes (graph)
    IF node->type=user  AND node->imagefile does not exist AND
                node->css >= threshold THEN
        node->outdegree=degree(node,degree)
        userGraph.ADD(node)
FOR edge In all edges(graph)
    IF sourceNode(edge) EXIST(userGraph) AND
        targetNode(edge) EXIST(user Graph)
        node1 =  sourceNode(edge)
        node2 =  targetNode(edge)
        edgeClone=CLONE(edge(node1,node2)
        userGraph. ADD(edgeClone)
RETURN userGraph
```

7.3. ALGORITHM: FIND CRUCIAL ORGANIZATION

In this algorithm we try to find the crucial organizations. The purpose of finding the crucial organization is to find the organization that will be key to getting organizations to talk. It can be identified because when the organization is removed from the network, it should lead to disconnecting of the graph. If this organization is found it will be good to speak to this organization to get to other organizations that are less connected to participate and talk about possible collaborations.

The organizations chosen are the one with higher betweeness centrality as they are the ones that are the links between more number of other people or organization.

```
Procedure: findCrucialOrg (graph)
Input: graph
Output: graph crucialOrg
    START
    TABLE bw (node, between) = SNAP->CalculateBetweennessCentrality(graph)
    Declare crucialOrg
    FOR node IN all nodes (graph)
        IF (node->type=usee AND EXIST (node->imagefile) AND
            node->betweeness>threshold)
                crucialOrg.ADD(node)
    FOR edge In all edges (graph)
        IF (sourceNode(edge) EXIST(crucialOrg) AND targetNode(edge)
        EXIST(crucialOrg))
                node1 = sourceNode(edge)
                node2 = targetNode(edge)
                edgeClone=CLONE(edge(node1,node2)
                crucialOrg. ADD(edgeClone)
    RETURN crucialOrg
```

8. EXPERIMENTAL RESULTS

In this thesis, we were able to achieve two major targets of this research topic. We considered Facebook pages for different charitable organizations in the state of Louisiana so that we may be able to help in Louisiana poverty initiative to be able to help different organizations to network or collaborate with each other according to their aim or the causes they work for.

The project lets us store all the data obtained in the graph to be stored in another data structure for efficient search purposes where the user can search for certain keywords, even hash tags and obtain all the post and comments from the page that contain the searched word and in addition the information related to that comment including the link to the person's or page's profile and the comment itself to understand the data.

Secondly, we were able to develop algorithms that help us find the links between different nodes by finding all the users that have proven to be key connections by finding the pages and posts they have participated in by finding the users who have participated in the same group, we know that those users may be the ones that should be talking (Section 7). The output measurements gives the Cluster coefficient of all the nodes and their Gregariousness is as follows.

```
>user -g graph3 -t .5
GraphName:  graph3 threshold: 0.5
Before For
Vertex Lavaail Doughty has 0.5 clustering coef
Vertex Lavaail Doughty has 10 outgone edges
Vertex Nicole Strickland has 0.5 clustering coef
Vertex Nicole Strickland has 9 outgone edges
Vertex Olevia Williams has 0.5 clustering coef
Vertex Olevia Williams has 8 outgone edges
Vertex Stevie Toups has 0.5 clustering coef
```

Figure 9: Output for User Connectivity computation.

9. SUMMARY

In this world where social media is growing rapidly, the purpose of the existence of social media has come a long way, from initially being a platform for friends to stay in touch even from different parts of the worlds to then becoming a place to meet new people and stay updated with the old ones, and as its evolution continued came many fields where the social media started playing roles, Social media consists of networks, which has led to large number of researchers on social media network analysis, be it behavioral analysis or data analysis. Text mining has also become crucial in various analysis oriented research.

With so many possibilities through social media new ventures starting is also become a part, this can be put into utilization, in this research we try to use social media for various poverty initiatives in the state of Louisiana. There are a number of charitable organizations that work towards poverty initiatives and this research tries to be an instrument to the collaborations and networking that can happen by making all of these organizations and people talk to each other according to their online presence.

The research aims at developing efficient algorithms for knowledge discovery and data mining that meet the challenges of huge/noisy data set and complex connection patterns inherent in social network media in real life. The new methods and algorithms developed in this project advance the text mining and social network analysis research by combining text analysis, machine learning, and community detection in social networks and network optimization in a unified framework. The results of text mining and network analysis will be used in a search-based decision support tool, which will involve novel cost models for measuring network efficiency and potentially novel search algorithms.

The project also addresses important sociological research questions regarding the use of websites and social media by offline interagency networks. While research exists on the functioning and effectiveness of interagency collaborations, this research has occurred with offline networks. Similarly, research on social media use by organizations has focused on the strategies adopted by single organizations. This work compares online and offline interagency networks and develops a theory about how the two are related in a domain likely experiencing a digital divide. It also develops a theory of effective interagency network structure in tandem with the development of the decision support tool.

10. FUTURE WORK

This research on social media is to find the best way to make organizations collaborate. Being a vast field that social media analysis is, this research has many places it can be further elaborates and more features can be added to this project.

With the algorithms that have been mentioned in this thesis, the results, and with further graph measurements we should be able to create a model with combining the graph analysis and data mining an efficient decision support tool can be developed.

Also this project it can be expanded to be useful not only to the state of Louisiana but more states.

One of the most important function that need to be implemented is to successfully be able to demonstrate all the measurements and networks obtained visually, so that data can be easily understood at a glance and further providing option according to the visualized data.

Another section is the need to develop a theory regarding the relationships between interagency online and offline networks. With measurements obtained from the algorithms, we need to determine whether the obtained measurements and the conclusions made using these are analogous to the offline relations.

Also a feasibility review of the algorithms, the theories and models developed. To understand how useful this research has proved to be, and if it is same as the expected results.

11. REFERENCE

1. NodeXL definition. Retrieved from https://nodexl.codeplex.com/, accessed on April 12, 2016.

2. The GRAPHML file format. 2015. Retrieved from http://graphml.graphdrawing.org, accessed on April 29, 2016.

3. K. Batool, N. M. 2014. Towards a Methodology for Validation of Centrality Measures in Complex Networks. J*ournal.pone.0090283*. doi: 10.1371.

4. Kadushin C., M. L., D. Ryan, A. Brodsky, and L. Saxe. 2005. Why is it so difficult to form effective community coalitions. *City & Community, 4*, 255-275.

5. Osher D., K. D., and S. Jackson Sopris. 2003. Safe, supportive, and successful schools step by step. *West, Longmont*.

6. Hansen Derek, C. D., Ben Shneiderman. (2010). Analyzing Social Media Networks with NodeXL: Human-Computer Interaction Lab 27th Annual Symposium.

7. Cross J. E., E. D., R. Newman-Gonchar, and J.M. Fagan. (2009). Using mixed-method design and network analysis to measured development of interagency collaboration. *American Journal of Evaluation, 30*(3), 310-329.

8. Koelling D. I. a. A. M. 2009. The potential of web sites as a relationship building tool for charitable fundraising. *Public Relations Review, 35*, 66-73.

9. Leskovec J. Stanford Network Analysis Package (SNAP). Retrieved from http://snap.stanford.edu/, Accessed on April 29, 2016.

10. Kreuter M.W., N. A. L., and L. A. Young. E. 2000. Evaluating community-based collaborative medhanisms: Implications for practitioners. *Health Promotion Practice, 1*, 49-63.

11. Smith Marc A., B. S., Natasa Milic-Frayling, Eduarda Mendes Rodrigues, Vladimir Barash, Cody Dunne, Tony Capone. 2011. Analyzing (Social Media) Networks with NodeXL. *ACM* (978-1-60558-601-4/09/06.).

12. Reagans R. and B. McEvily. 2003. Network structure and knowledge treanste: The effects of cohesion and range. *Administrative Science Quarterly, 48*, 240-267.

13. Berkowitz B. 2001. Studying the outcomes of community-based coalitions. *American Journal of Community Psychology, 29*, 213-227.

14. Brandes Ulrik, M. E., Jürgen Lerner, Christian Pich. *GraphML Progress Report.*

15. Weisstein E. W. Cut Set. Retrieved from http://mathworld.wolfram.com/ CutSet.html, Accessed on April 29. 2016.

16. Punuru J. and J. Chen. May 2006. Automatic acquisition of concepts from domain texts. In *Proc. of IEEE Int. Conf. on Granular Computing*, Atlanta, Georgia.

12. VITA

Neha Clare Jose is from Kerala, India. She graduated from Mahatma Gandhi University, Kerala, India in the 2013 with a Bachelor of Technology in Information Technology. Started working as a Technical Consultant soon after in GapBlue Software Pvt. Ltd following which moved to the United States in 2014 and joined Louisiana State University to do her Master's in Science Engineering Science. Her Technical Interest is in Optimization, Data Mining and Database Systems.