VIRGINIA TECH

CS 5604 - INFORMATION STORAGE AND RETRIEVAL

Project CINETGraphCrawl: Constructing Graphs from Blogs

Author:
Rushi KAW
Hemanth MAKKAPATI
Rajesh SUBBIAH

Supervisor:
Dr. Edward FOX

December 11, 2012

CINETGraphCrawl: Constructing Graphs from Blogs

Rushi Kaw, Henmath Makkapati and Rajesh Subbiah
Dept. of Computer Science
Virginia Tech
rushik@vt.edu, makka@vt.edu, and csrajesh@vt.edu

Abstract

Internet forums, weblogs, social networks, and photo and video sharing websites are some forms of social media that are at the forefront of enabling communication among individuals. The rich information captured in social media has enabled a variety behavioral research assisting domains such as marketing, finance, public health and governance etc. Furthermore, social media is believed to be capable of providing valuable insights into understanding information diffusion phenomena such as social influence, opinion formation and rumor spread. Here, we propose a semi-automated approach with prototype implementation that constructs interaction graphs to enable such behavioral studies. We construct first and second degree interaction graphs from Stackoverflow, a programming forum, and CNN Political Ticker, a political news blog.

1 Introduction

Social media has gained unprecedented attention recently after radically changing the way people communicate with each other. Internet forums, weblogs, social networks, and photo and video sharing several websites are a some forms of social media that are commonly found. Social media technologies extend an easy means to connect and communicate with a wider community of people as opposed to the traditional means of communication. Ease of use and access, wider reach, low cost and high availability of social media has lead to its widespread adoption among individuals, business enterprises and governments.

Social media enables users to post content and share it with their peers. The typical postings on social media are known to contain valuable information such as opinions, expectations, decisions, feelings and interests etc of the individuals. The richness in personal details makes the social media data highly conducive for conducting social and behavioral studies. Social media data has been used to study domains such as marketing, public health and finance etc. Moreover, latest research suggests social media to be an effective vehicle for studying information diffusion and related phenomena such as rumor spread and opinion formation. [9] [3] [2] are some efforts towards this. Information diffusion is generally studied by employing a variety of graph analysis techniques over graphs that represent the underlying ecosystem being studied. In this project, we aim to build a semi-automated graph construction mechanism to aid the information diffusion research at Network Dynamics and Simulation Science Laboratory (NDSSL). At NDSSL, active research is being pursued to study rumor spread and opinion formation over a network of people and, the role played by social influence in shaping these phenomena. To facilitate this research, we aim to construct graphs that represent the interactions of users over social media. These graphs will be further utilized by research groups at NDSSL to study various information diffusion phenomena.

In particular, we will be investigating construction of graphs from the more descriptive forms of social media such as internet forums and blogs. Forums and blogs differ from other kinds of social media like microblogging and social networking platforms in their purpose, structure and governance. Popular social media platforms like Facebook, Twitter, LinkedIn, Flickr, Youtube, Pinterest and Google+ etc enforce a strict structure on the postings, which makes it easier to access and process the information such as other users tagged, topical tags, geographic locations and events etc. Also, the postings on these platforms tend to be succinct, which makes it harder to analyze the content. On the other hand, forums and blogs employ a relaxed structure and allow the users to post rich and descriptive information, which is more conducive for analysis. However, as blogs and forums are hosted by many diverse entities, they lack a common structure, which poses challenges in extracting relevant information. In this manuscript, we discuss a metamodel-based mechanism to semi-automate the process of graph construction from blogs. We present the extracted graphs from Stackoverflow [1], an internet forum, and CNN Political Ticker [6], a political news blog.

The rest of the paper is organized as follows. Section 2 explains our methodology towards constructing graphs from blogs. Section 3 explains our implementation details. Section 4 presents the different graphs constructed using our prototype implementation. Section 5 investigates the different ways in which we can extend our work and Section 6 concludes.

2 Methodology

Blogs are informational websites that enable content sharing and interaction among authors and their followers. Typically, a user or small group of users post content that may be themed around a single subject in a posting. Most blogs let the visitors read and comment on the posts to express their opinions. The mechanism of commenting allows users to interact with each other and engage in a conversation. The blog posts and their comments offer a rich source of information to analyze how conversations grow over time and eventually contribute to opinion formation on a wider-scale. However, analyzing blogs pose several challenges as they do not follow one common structure and semantics. For instance, some blogs of-

Project Tech Report'12, Dec 12, 2012, Virginia Tech, VA, USA.

fer reputation mechanism to suggest usefulness of comments and users. While some blogs allow anonymous usage, others do not. Furthermore, in some blogs users can associate themselves with various affiliations based on their preferences like political parties, programming languages, and sports teams etc. Blogs further differ in the way they support comments. Some blogs allow comments to be posted in a hierarchical manner while some employ a flat style.

In this project, we propose a semi-automated approach to construct graphs from blogs. Our approach employs a metamodel to integrate distinct features of various blogs into one form. As shown in Fig. 1, we crawl the blogs using custom crawlers that are written specifically to match the structure and other features of the blog in consideration. The custom crawlers extract relevant information from the blogs and convert them into an instance of our metamodel. Going further, we utilize the extracted content to construct various kinds of graphs based on user input. This metamodel-based approach gives us the flexibility to introduce many custom crawlers and process a variety of blogs consequently.

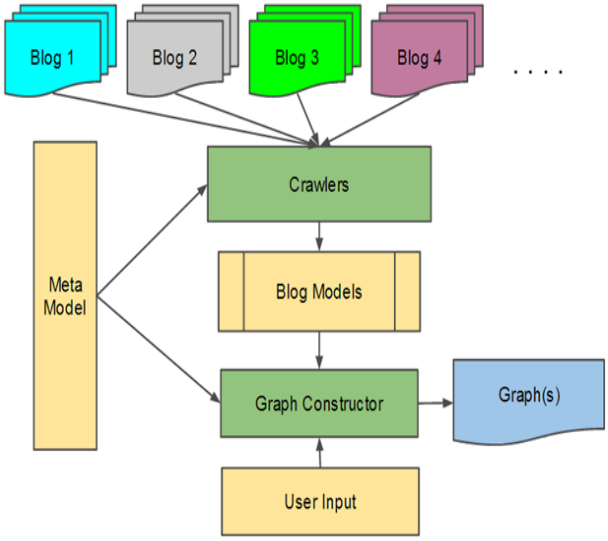

Figure 1. An overview of the methodology for graph construction from blogs

2.1 Metamodel

Fig.2 depicts the metamodel we devised for this project by including some of the most commonly found features in blogs. A blog is treated as a collection of posts, expressed by "*has post* " relation, that are posted by the users of the blog as captured by the "*posted by*" relation. Users may be registered or anonymous depending on the governance policies of the blogs. All anonymous users are currently mapped to one representative user. Further, the "*has comment*" relation enables posts to carry comments that are posted by other users of the blogs, which is captured by the "*commented by*" relation. Posts can also carry tags with the help of "*tagged as*" relation to indicate the topic being discussed. Moreover, comments may in turn carry other comments facilitating a hierarchical structure as indicated by the reflexive relation "*has comment*". Comments can also be tagged through the "*tagged as*" relation. Both comments and users can carry reputation as indicated by the blog users with the help of "*has reputation*" association. Finally, users affiliation can be captured using the "*affiliated with*" relation.

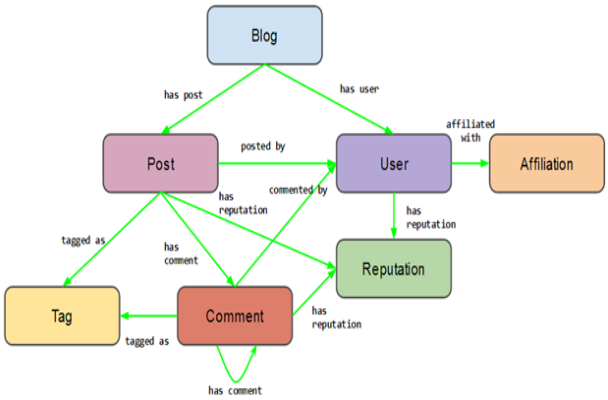

Figure 2. The blog metamodel employed for standardizing various blog structures

2.2 Crawlers

Crawlers are programs that aim to find URLs to various posts of a blog and extract relevant information for graph construction. From each URL, information pertinent to posts and their associated comments is collected. For every post, the title, author, content, comments, tags and reputation etc are collected. In turn, for each comment, title, author, authors affiliation, subject, tags and reputation etc are collected. This information, once extracted, is converted into representation that adheres to the blog metamodel and stored for graph construction.

2.3 Graph Constructor

Graph Constructor is a generic program that can construct specific graphs from the information extracted by the crawlers. With the understanding of metamodel, graph constructors are oblivious and independent of the blog structure to construct graphs. Graph constructor can construct well-known graphs like first and second degree interaction graphs [14]. Furthermore, we envision enabling the users to construct custom graphs by specifying input that indicates the nodes and edges to be created in the resultant graphs. We discuss our approach towards constructing first and second degree interaction graphs between users in the following sections.

2.3.1 User-user Interaction Graphs

Assume a set of blogs B, each consisting a set of users U and posts P. Each post of blog B_i is written by a user $Uij \in B_i$. Also, each post can carry a set of comments C, each written by a user. We follow the below described procedure to construct the user-user first and second degree interaction graphs:-

1. Choose a blog B_i from the collection of blogs B
2. Retrieve all users belonging to blog B_i
3. For each user $U_{ij} \in B_i$, retrieve all the posts P written by U_{ij}

4. For each such Post P_{ijk}, retrieve all the comments C posted by users such that $U_{id} \in B_i, U_{id} \neq U_{ij}$
5. Create an edge between U_{ij} and U_{id}
6. Repeat steps 4 and 5 for all the posts P
7. Retrieve all the comments $`C$ written by user U_{ij}
8. For each comment in $`C$, retrieve the parent post$P_{i`j`k}$ written by $U_{i`j}$
9. Create an edge between U_{ij} and $U_{i`j}$
10. Repeat steps 1 through 9 to complete the 1^{st} degree user-user interaction graph

Further, to construct the second degree interaction for user U_{ij}, we follow the following steps:-

1. Retrieve all the comments $`C$ posted by user U_{ij}
2. For each comment in $`C$, retrieve its parent article say $P_{i`j`k}$ written by $U_{i`j}$
3. Retrieve all the comments $``C$ associated with parent article $P_{i`j`k}$
4. For each comment in $``C$, fetch the user $U_{i``d}$ who posted the comment and establish the second degree interaction with the user U_{ij}
5. Repeat steps 1 through 4 to complete the 1^{st} degree user-user interaction graph for the user U_{ij}
6. Repeat these steps for all the users to complete the second degree interaction graph

2.3.2 *Hierarchical Representation*

To understand the different levels of interactions happening in the blogs, we represent the blog as a hierarchical tree. We represent the blog itself as the root node and its set of posts as 1^{st} level child nodes. Then for each post, we create its child nodes representing its comments. Then, for each comment node, we in turn retrieve the associated comments and represent them as the next level child nodes, and so on. This graph gives an overall structure of the blog and also gives an idea of most popular posts based on the comment spread.

3 Implementation

In this section, we describe a prototype implementation of the above proposed methodology over two data sets, a programming forum and a political blog. We store the information extracted from the two data sets into database adhering to a data model that realizes our metamodel. The graph construction algorithms interact with the database to fetch the information and construct appropriate graphs. The extracted graphs are represented using an adjacency matrix in a plain text file, which is then used to visualize the graphs in Gephi [4].

3.1 Data sets

We primarily use two data sets to demonstrate our approach, Stackoverflow and CNN Political Ticker. Stackoverflow is an internet forum that features questions and answers on a wide range of programming-related topics. It is the most used forum in the Stack Exchange family with 1.2M users, 3.3M questions, 6.6M answers and 13M comments as on August 2012 [13]. Stack exchange periodically creates data dumps of all the conversations under a creative commons license. We downloaded the quarterly Stack Exchange dump, which is available as a set of XML files. We have written custom XML parsers to parse, extract and store relevant information required for graph construction. Stackoverflow has a structure that is very similar to a blog with users posting questions and answers. Additionally, stackoverflow allows questions to carry comments, which are generally clarifications asked over the questions. As shown in Figure 3, we map a question to a post, a comment to a comment and an answer also to a comment, but a specialized comment.

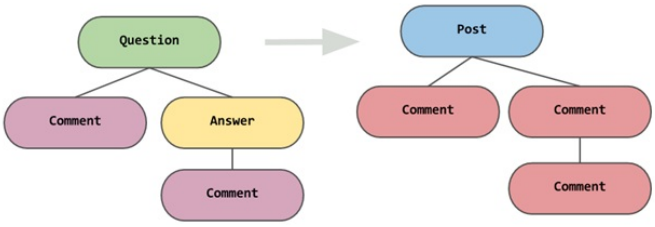

Figure 3. Mapping from stackoverflow elements to metamodel elements prior to graph construction

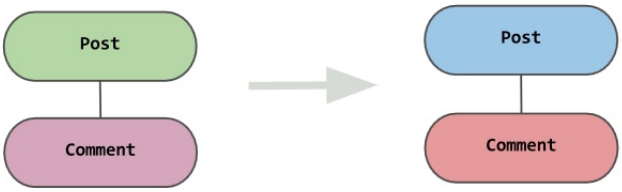

Figure 4. Mapping from CNN political ticker elements to metamodel elements prior to graph construction

CNN Political Ticker is a news blog featuring the latest political developments tracked by CNN. The CNN journalists post political stories that the users read and leave comments on. Users can leave comments using their CNN accounts or anonymously as well. When users post anonymously, it is harder to distinguish between users who have posted anonymous. In this project, we treat all the anonymous and duplicate usernames as one user. CNN Political Ticker employs the structure of a traditional blog with a set of posts that can carry comments on them. This enables us to do a simple one-on-one mapping of the blog structure to our metamodel as shown in Figure 4.

3.2 Data Store

Figure 5 shows the data model we used to realize our metamodel. All the tables are represented in third normal form. The tables Post, User and Comment are direct representations of the corresponding elements in the meta-model. The additional tables capture the blog specific information that is lost in the conversion to metamodel. The COMMENT_MAPPING table is used to capture the blog specific post-comment relationship and its semantics. For example, in our stackoverflow data set we represent the answer as special type of comment (COMMENT_MAPPING.INTERACTION_LEVEL = 1) and we use the COMMENT_MAPPING table to differentiate it with the regular comment (COMMENT_MAPPING.INTERACTION_LEVEL = 2).

3.3 Crawling and Extraction

Crawling and content extraction are the two primary steps prior to graph construction. While crawling we first fetch the list of post URLs from the seed URLs - URLs to blog. We use the popular GNU command WGET [8] to fetch the desired set of post URLs from the seed URLs. We use the following command to get the desired URLs

" *wget –no-parent -nv -r –spider 'seed URL'* "

where,

–spider is the argument to indicate WGET to not download the

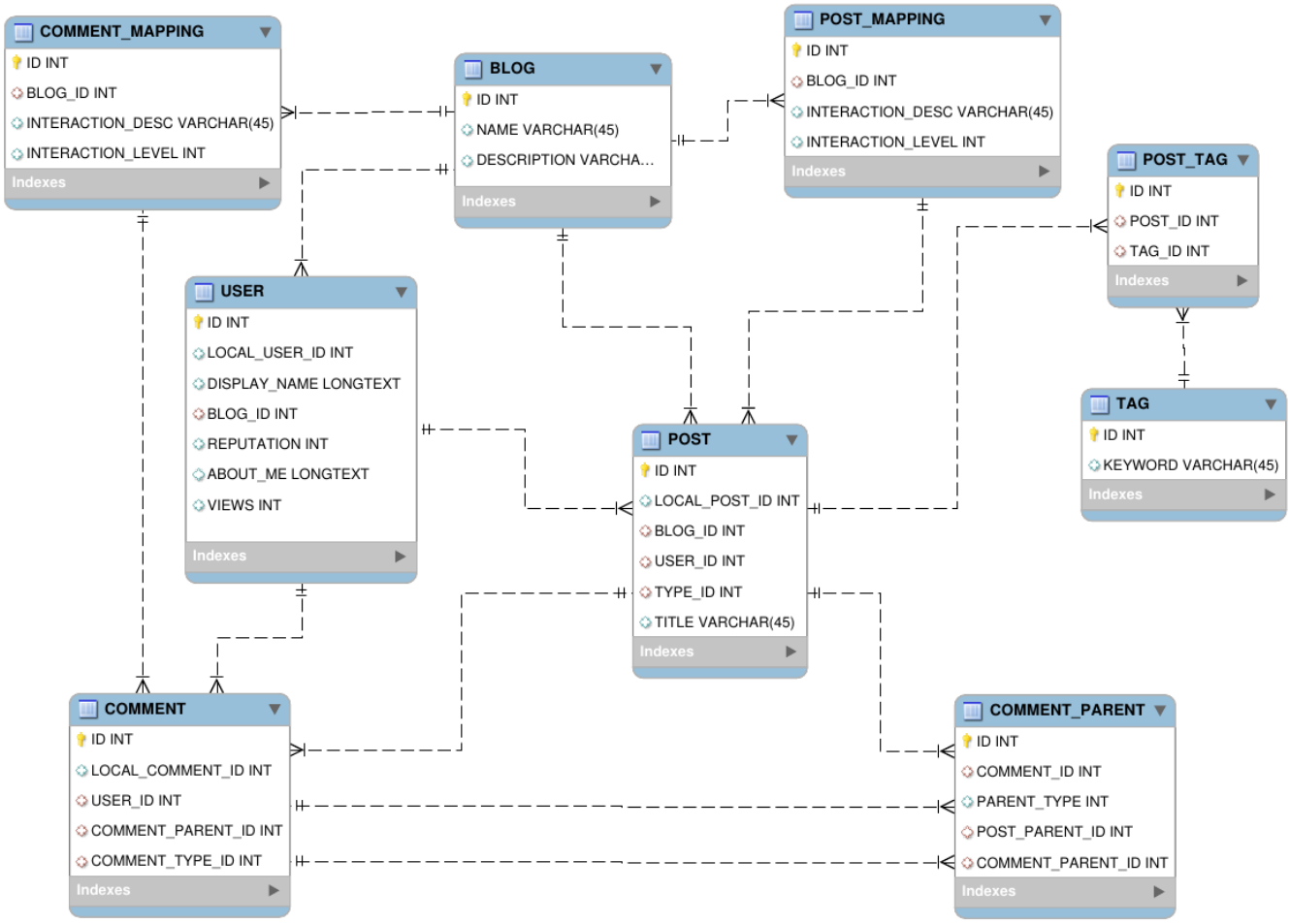

Figure 5. The data model used for dumping the extracted contents from blogs into database for further processing

content
–no-parent argument indicates WGET to not ascend to the parent directory
-r argument suggests recursive extraction of URLs from the sub-directories of the seed URLs

Further, we download the document from the retrieved set of URLs using Urllib [11]. The downloaded pages are then parsed for posts, comments, users, tags and the associations between them using Beautifulsoup [10].

3.4 Graph Construction

3.4.1 User-user Interaction Graph

We use SQL queries to retrieve the necessary information for graph construction algorithm discussed in the Section 2.3.1.

1. Retrieve each blog from the blog table using the query:- SELECT * from USER WHERE BLOG_ID = [blogID]
2. Then, retrieve the posts written by the user using this query:- SELECT ID from POST WHERE USER_ID = [userID] AND BLOG_ID = [blogID]
3. Get all the 1^{st} level comments associated with the post using the query :- SELECT ID, USER_ID from COMMENT WHERE POST_ID = [postID] AND COMMENT_TYPE = [Interaction level]
4. Finally, get all the comments posted by the user and get its parent post based on the comment type
   (a) If COMMENT_TYPE is 1, then the parent node is of POST type, fetch the user who wrote the post.
   (b) Else if COMMENT_TYPE is 2, then the parent node is of COMMENT type 1^{st} *levelcomment*, get the user who wrote that comment
5. This completes the 1^{st} degree interaction edges for the user
6. To get the second degree interaction graph, fetch all the 1^{st} level comments associated with the parent node retrieved at Step 4

After we retrieve the associated edge list for each user, we store the edge list information in a graph file. The format of our graph file is a simple adjacency matrix represented as:

"Source Node, Destination Node, Weight, Label"

3.4.2 Hierarchical Graph

To construct the hierarchical graph representation, we retrieve the posts and comments from the blog and represent them in a tree structure. For each retrieved post we use the Breadth First Search algorithm to retrieve different levels of comments associated with the blog post. We finally store the constructed tree in JSON format. We then use the Data-Driven Documents (D3) JavaScript Library

[5] to represent it as a dendrogram. Figure 6 shows the dendrogram for stackoverflow data. Since, the D3 library has scalability limitations w.r.t number of nodes, we reduced the number of nodes to be considered for dendrogram.

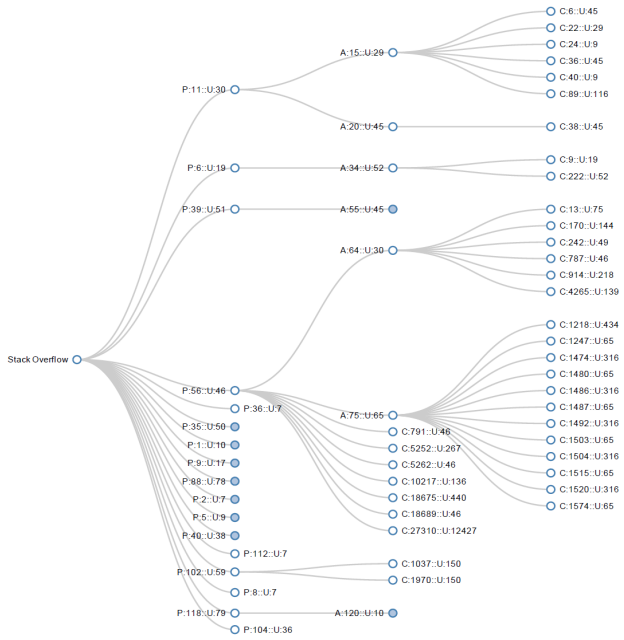

Figure 6. Hierarchical Representation of stackoverflow posts and comments

4 Results

In this section, we showcase and briefly describe the graphs extracted using our prototype implementation. We extracted the first degree interaction graph for the Stackoverflow dataset, and both first and second degree interaction graph from CNN Political Ticker. We used Cyber-Infrastructure for Network Science (CINET) [12] to compute certain characteristics of the graphs. CINET provides an infrastructure to computer many graph analysis measures on graphs. Here, we collect number of nodes, number of edges, minimum degree, maximum degree, average degree and number of triangles of a graph to elicit certain properties of the graphs extracted. Especially, 'number of triangles' represents the number of cycles present in the graph and a good indicator of the amount on interconnectedness between nodes.

4.1 Stackoverflow

We extracted the first degree interaction graph of Stackoverflow as described in Section 2.3.1. A few characteristics of the graph are enlisted in Table 1. A high value for NT, 51 million on this case, signifies that the graph is highly interconnected. As all users are equally eligible to post questions, answer and comment, there is a greater chance of users interacting with many other users directly. We couldn't visualize this graph as the ND is greater than 50,000, a limitation of Gephi [7]. However, we constructed a reduced version of Stackoverlfow graph with only the first 35,000 users. A visualization of this graph is depicted in Figure 7 and characteristics are enlisted in Table 2. It is evident that the decrease in NN resulted in the decrease of other characteristics as well except AD. Users of Stackoverflow are ordered as per their reputation in the dump. A user of higher reputation would have a higher number of questions, answers and comments, which leads to a higher degree. Hence, the AD increases for the first 35,000 users as all of them have a relatively higher degree.

Number of Nodes (ND)	80562
Number of Edges (NE)	9.3M
Minimum Degree (MiD)	1
Maximum Degree (MxD)	21869
Average Degree (AD)	23.1
Number of Triangles (NT)	51M

Table 1. Characteristics of stackoverflow first degree interaction graph

Number of Nodes (ND)	34265
Number of Edges (NE)	1M
Minimum Degree (MiD)	1
Maximum Degree (MxD)	5601
Average Degree (AD)	59.5
Number of Triangles (NT)	8.7M

Table 2. Characteristics of reduced stackoverflow first degree interaction graph with approximately 35,000 nodes

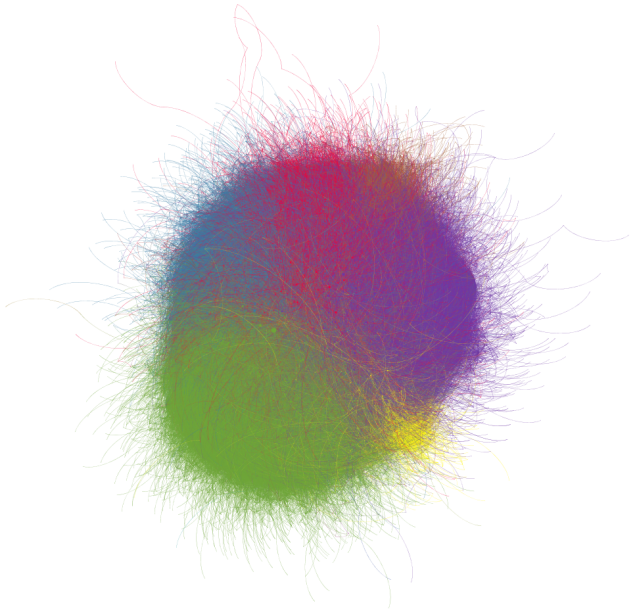

Figure 7. Visualization of reduced stackoverflow first degree interaction graph

4.2 CNN Political Ticker

The first degree interaction graph extracted from the 270 posts of CNN Political Ticker between 1 January 2012 and 2 February 2012 is depicted in Figure 8. Table 3 showcases the characteristics of the first degree interaction graph, which is relatively less dense with low AD and NT. This characteristic is a direct consequence of the fact that only a certain set of users, the CNN journalists, are allowed to post on CNN Political Ticker and others, the users, can only read the posts. This results in a reduced first degree interactions among the users. The two classes of users can be evidently seen in the visualization with blue and yellow dots representing the

journalists and users respectively. The nodes are colored based on their modularity class.

Number of Nodes (ND)	2945
Number of Edges (NE)	5117
Minimum Degree (MiD)	1
Maximum Degree (MxD)	943
Average Degree (AD)	3.47
Number of Triangles (NT)	35

Table 3. Characteristics of reduced CNN political ticker first degree interaction graph

Going further, we extracted another graph for the same set of 270 posts with both first and second degree interactions. This graph, however, was too dense to visualize with Gephi but Table 4 shows the characteristics to bring the contrast w.r.t the first degree interaction graph. The NE, AD and NT clearly depict the higher density of this graph as opposed to the first degree interaction graph. The second degree interactions take into the account the indirect interactions between users through the journalists, which results in a high NT.

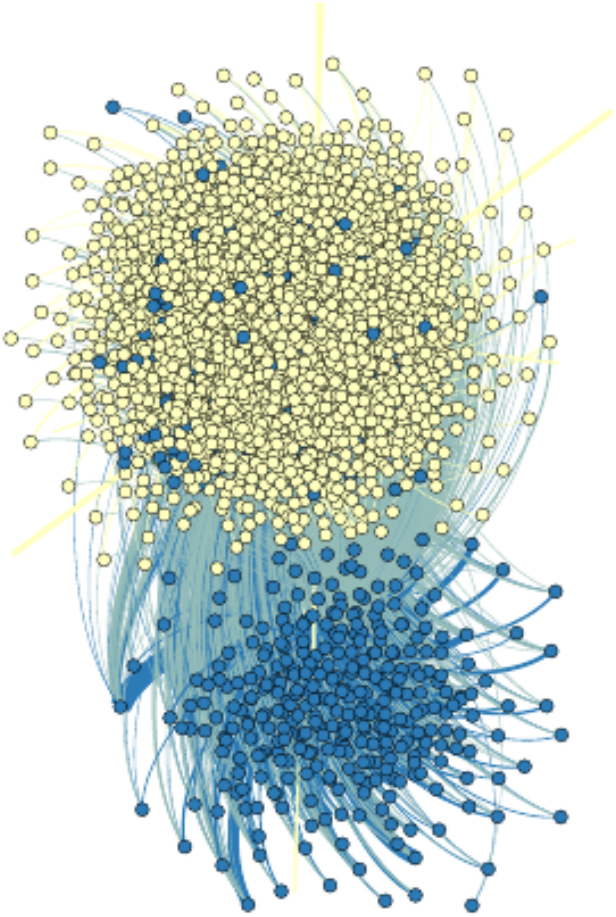

Figure 8. Visualization of CNN first degree interaction Graph

5 Future Work

Owing to time limitations, in this project, we considered only one blog to construct graphs. With a minor effort, our approach can be extended to address multiple blogs. It is vital to consider

Number of Nodes (ND)	2945
Number of Edges (NE)	0.4M
Minimum Degree (MiD)	1
Maximum Degree (MxD)	2120
Average Degree (AD)	294.5
Number of Triangles (NT)	54.2M

Table 4. Characteristics of CNN political ticker second degree interaction graph

blogs on different topics and structures. However, with the current approach, one would have to write custom crawlers to extract information from blogs of different structures. It would be very useful to investigate the possibility of a generic content extraction mechanism. Doing so, many new blogs can be considered for graph extraction with minimal changes. Furthermore, richer graph representations may be required to incorporate the variety of information found on blogs such as images, videos, tags, hashtags and urls etc. Richer graph representations may help boost the analysis to gain further insights.

6 Conclusion

We proposed an approach to construct graphs from blogs that can be used in social media analysis. We demonstrated a prototype implementation to show the feasibility of such an approach using two data sets, an internet forum on programming and a political blog. We presented some characteristics of the extracted graphs that show their representativeness to the original sources.

7 References

[1] Stackoverflow. *http://stackoverflow.com/*.

[2] L. A. Adamic and N. Glance. The political blogosphere and the 2004 u.s. election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, LinkKDD '05, pages 36–43, New York, NY, USA, 2005. ACM.

[3] A. Apolloni, K. Channakeshava, L. Durbeck, M. Khan, C. Kuhlman, B. Lewis, and S. Swarup. A study of information diffusion over a realistic social network model. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, CSE '09, pages 675–682, Washington, DC, USA, 2009. IEEE Computer Society.

[4] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. 2009.

[5] M. Bostock, V. Ogievetsky, and J. Heer. D3; data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301 –2309, dec. 2011.

[6] CNN Politics. Cnn politicalticker. *http://politicalticker.blogs.cnn.com/*, .

[7] Gephi Community. Gephi:- performance and scalability. *https://gephi.org/2008/performance-and-scalability/*, .

[8] GNU. Wget. *http://www.gnu.org/software/wget/*, .

[9] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 491–501, New York, NY, USA, 2004. ACM.

[10] Python Library. Beautiful soup. *http://www.crummy.com/software/BeautifulSoup/*, .

[11] Python Library. Urllib. *http://docs.python.org/2/library/urllib.html*, .

[12] Sherif Elmeligy Abdelhamid et al. CINET: A CyberInfrastructure for Network Science. In *Proceedings of the 2012 IEEE Eighth International Conference on eScience, 2012*, Oct. 2012.

[13] StackExchange. Stackoverflow. *http://blog.stackoverflow.com/category/cc-wiki-dump/*, August, 2012.

[14] C. Wilson, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao. Beyond social graphs: User interactions in online social networks and their implications. *ACM Trans. Web*, 6(4):17:1–17:31, Nov. 2012.