2012 International Conference on Social Informatics

SIN: A Platform to Make Interactions in Social Networks Accessible

Roozbeh Nia*, Fredrik Erlandsson†, Prantik Bhattacharyya*, Mohammad Rezaur Rahman*, Henric Johnson†, S. Felix Wu*
*University of California, Davis
{rvnia, pbhattacharyya, mrrahman, sfwu}@ucdavis.edu
†Blekinge Institute of Technology
{fredrik.erlandsson, henric.johnson}@bth.se

***Abstract*—Online Social Networks (OSNs) are popular platforms for interaction, communication and collaboration between friends. In this paper we develop and present a new platform to make interactions in OSNs accessible. Most of today's social networks, including Facebook, Twitter, and Google+ provide support for third party applications to use their social network graph and content. Such applications are strongly dependent on the set of software tools and libraries provided by the OSNs for their own development and growth. For example, third party companies like CNN provide recommendation materials based on user interactions and user's relationship graph. One of the limitations with this graph (or APIs) is the segregation from the shared content. We believe, and present in this paper, that the content shared and the actions taken on the content, creates a Social Interaction Network (SIN). As such, we extend Facebook's current API in order to allow applications to retrieve a weighted graph instead of Facebooks unweighted graph. Finally, we evaluate the proposed platform based on completeness and speed of the crawled results from selected community pages. We also give a few example uses of our API on how it can be used by third party applications.**

***Index Terms*—Social Networks, Social Interactions, Facebook Graph API**

I. Introduction

Facebook has over 950 million users and still growing. There are over 2.7 billion likes and comments posted on Facebook on a daily basis as of February 1st 2012[1]. The fundamental block of the Facebook platform is the social graph. A social graph can be defined as set of nodes and edges, where each node represents a user and each edge represents a connection between two users. Moreover, along with the growth in social graph, Facebook has introduced technologies for users to share multiple level of information. Users share personal information related to their *name, contact details, photo, current location, hometown, interests, activities* among other examples. Users also share non-personal information in the form of content from the traditional Web. 'Share', 'Like' and 'Recommend' buttons typically help users share this set of content. Interactions on Facebook pages also creates an additional set of content. Facebook allows users to interact with each other through many different means through the content shared on its platform. For instance, users can like, comment on, or re-share a content that is posted by another user. Users are not limited to interact with their immediate friends, but they can interact with anyone on Facebook through Facebook Pages and/or Groups.

Most online social networks including but not limited to Facebook, Twitter, and Google+ provide APIs for third party applications to request parts of the social graph. The traditional API provided by Facebook is able to capture the static information regarding the social graph as described above, but is limited in regards to providing the social interactions on its platform. For instance, many "occupying movement" pages have been created recently on Facebook, which brings up the question of how would one capture the social interactions of one page and combine or compare the results with the social interactions of another related page. For example, users of Occupying San Francisco seem to be interacting more around the idea of "Tea Party," while the users of Occupying Chicago had more interactions on the issue of large corporations. Often times, we do not need to know who in particular has had interactions, but we are interested to know what the society as whole is interested in; therefore, we can anonymize the networks to preserve users privacy without loss of critical information.

This particular limitation arises in Facebook APIs due to the fact that the social graph is completely segregated from the content shared on these networks except for the ownership of the content. For example, an application can access the content shared by the logged in user or the immediate friends of the user if they have the required permissions, which gives a weak social relationship between the content and people. We believe that the content shared and the actions taken on the content, whether it is to like, comment, or re-share the content, creates a Social Interactions Network (SIN). The graphs generated from SIN connects people through actions and thus interactions with other users instead of the traditional friendship connections. We believe that the social interaction networks can be very useful and may represent a closer social network to the real life human interactions. The SINs could be used to solve many of the existing problems in today's world such as the Social Search Engine, Friend Finder, and/or Related Shopping Items.

978-0-7695-5015-2/12 $26.00 © 2012 IEEE
DOI 10.1109/SocialInformatics.2012.29

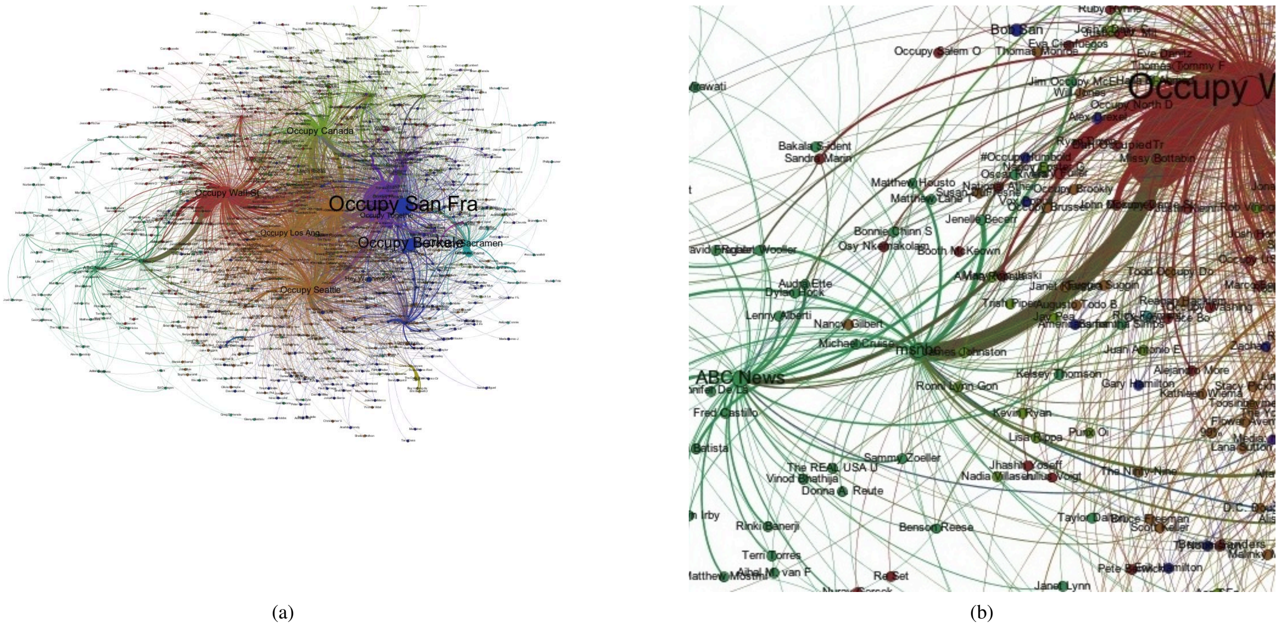


Fig. 1: shows interactions around the contents shared on several Facebook public pages in the third week of occupying movement. The users shown have interacted with other users on at least four communities. (a) Shows news agencies such as ABC News and MSNBC on the left side of the graph and the occupying movement communities on the right side of the graph. (b) A closer look at the left portion of the graph shows that MSNBC has a much stronger tie to the occupying movements than ABC News. This could result in a higher influence from MSNBC on the occupying movements compared to ABC News.

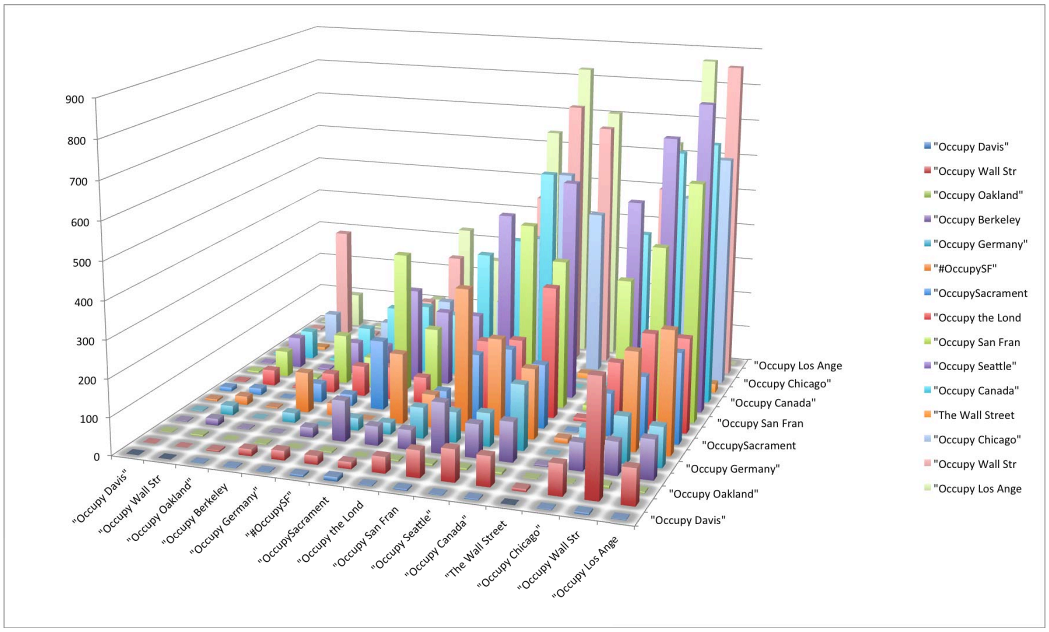


Fig. 2: shows for every two occupying movement pages how many users have interacted on both pages within the same time frame. The data above shows the interactions that had taken place during one week. For instance, the data shows that around 900 people interacted with other users both on “Occupy Los Angeles” page and “Occupy Chicago” page.

In this paper, we ask the question on how we can design a Social Content based API to support the interactions between social network users and the contents shared. We introduce a new set of API calls in addition to the current Graph API supported by Facebook, which allows third party applications to create Social Interactions Networks based on a given context. Our API is comparable to Facebook's Graph API, making it easy for further developers to easily adopt the new API.We also address the scalability issues of effectively capturing social interaction information from Facebook where the number of interactions are many and happen very quickly. We evaluate our API based on completeness of the results returned and the speed of our platform.

We also use our platform development efforts to analyze the influence of social interactions of a particular community has on other communities/pages on Facebook. Figure 1 shows the influence of news agencies on social interactions on some of the occupying movement communities on Facebook. Furthermore, figure 2 shows the influence of occupying movement pages on each other by showing for every two pages how many users have interacted on both pages within the same time frame.

The rest of the paper is organized as follows. In Section II, we talk about the related work in the area of social interactions network. We then describe the idea of social interactions network in section III. We give a few example applications for our API in section IV. Section V describes the details of our proposed API. In section VI we discuss the security, privacy, and implementation challenges. We evaluate our API in sections VII. Finally, section VIII talks about our future plans.

II. Related Work

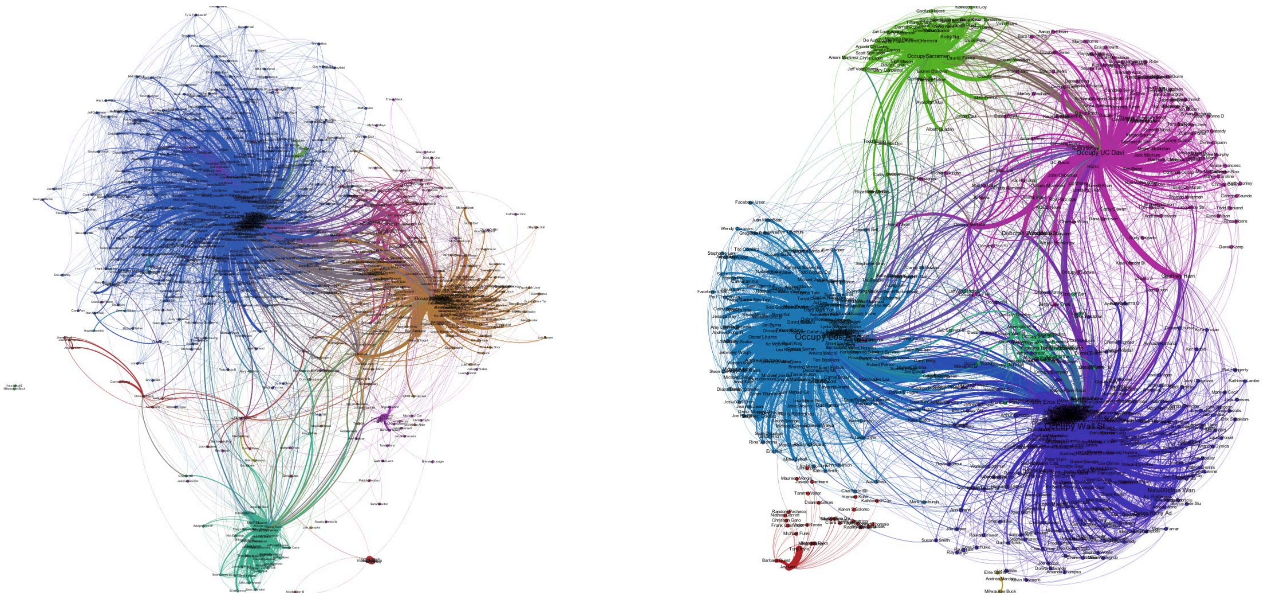
Researchers have begun to look at the real-world social interactions instead of the social networks of friendships or followers provided by OSNs. One of the original papers to study the emerging social network phenomena focused on the Club Nexus website of Stanford University[2]. Ever since, there's been work done on CyWorld, MySpace, Orkut[3], YouTube, Flickr, LiveJournal, and Orkut[4]. Yet another study focused on profiling social network evolution on Flickr and Yahoo! 360[5]. Finally, a recent measurement study analyzed the growth of Flickr social network using a three month crawl data[6]. These studies confirm that online social networks obey power-law scaling characteristics[7] and exhibit high clustering coefficients, firmly establishing them as small-world networks[8].

Recent studies analyzed the online communication patterns among the users in a large IM trace[9], and in an online social network[10]. The IM study also reported relatively higher value of average path length for the graph formed from user interactions. However, the IM interaction graph is more resilient to node removal than the interactions graph in Facebook, as Christo Wilson, et al indicated in their study[11], where they introduce the interaction graph as a more accurate representation of meaningful peer connectivity on social networks. They believe analysis of interaction graphs derived from their Facebook data reveal different characteristics than the corresponding social graph. They conclude with experiments to evaluate effects of interaction graphs on two well known social applications. The performance of RE[12] improves with the use of interaction graphs, as the streamlined link structure helps control spam proliferation. In the case of Sybilguard[13], the system becomes less able to effectively classify nodes once its assumptions about graph structure are violated. Researchers have shown that the social interaction networks represents a strong representation of active developers in OSS projects[14]. They further show that social interaction networks are very stable in presence of noise or lack of enough information and still have a very strong correlation with the active developers network[15].

III. Social Interactions Network

Next, we will discuss how we leverage the Facebook's platform to design one architecture that provides the social interactions networks. Most Social Networks today, such as Facebook, Twitter, and Google+, provide APIs for third party applications to build applications on top of their platform. Looking at the data of many different Facebook pages on how the users interact with each other, we have found that the social network graph that arises from these interactions differs a lot in type and structure based on the type of interactions we are looking at. The social graph provided by Facebook currently does not provide the enormous amount of information we can gain from the social interactions networks formed around Facebook communities. We believe that the content shared and the context of items shared on Facebook groups and pages plays a huge role in the formation of these social networks. Although, it is possible to recreate these networks from Facebook's current API, one has to make many different requests; since, Facebook only returns a limited portion of the data with each request, and do a lot of analysis and computations on the data retrieved in order to accomplish this task; therefore, not only recreating these social interactions networks from Facebook will require a lot of work, but a naive implementation may results in an incomplete results set due to instabilities, or a slow application due to delays in a sequential and non-parallel implementations.

In Social Interactions Network (SIN), which is an extension of FAITH[16], we provide a set of API calls in addition to Facebook Graph API calls to allow third party applications to retrieve the Social Networks formed around the contents shared on Facebook groups and pages efficiently and easy. From now on in this paper we will call these networks Social Interactions Networks. Our API uses the same ideology and interface as Facebook's Graph API, which makes it very easy for third party applications to adopt our API.



(a)

(b)

Fig. 3: (a) and (b) show the interactions done through comments and likes on the posts shared on UC Davis's Facebook page. (a) shows the interactions before the pepper spray incident, while (b) shows the interactions after the pepper spray incident.

We believe that each community (i.e. page, group, or a user's profile) on Facebook gives a context around which people will interact with each other. Looking at the community structure of social interactions network we believe that the context plays a huge roll in how people interact. For example, the social interaction network formed on Jay Leno's page is very different from the social interaction network formed in the Citi Bank page on Facebook. Facebook gives the option to page admins to allow or disallow fans to post on the page's wall. For instance, the Citi Bank page does not allow its fans to post content on their wall, so users can only like or comment what has already been posted by the page admins. Although, there is some interactions between users by liking comments that were posted by fans of the page, the average path length on the SIN formed around the contents shared on this page is one. Table I shows how the social interactions networks formed around the contents shared on different public pages on Facebook differ in number of members, the way users interact with each other, whether it is through likes or comments, the amount of interactions, and other network properties such as the overall average path length and clustering coefficient of the networks. Furthermore, the data shows us that the interactions on the same page can differ a lot in different time periods or around different events. Figure 3 shows the social interactions network around the contents shared on UC Davis's Facebook page. Figure 3 (a) shows the interactions before the pepper spray incident[17] at UC Davis, while Figure 3 (b) shows the interactions immediately after the incident. Given the community id (i.e. the context) our API will retrieve the social interaction network that is formed around the contents shared in that context.

IV. Applications

Friend Suggestion: One of the biggest challenges that most popular social networks face is their friend suggestion feature[11], called People you may know in Facebook. Although, Facebook does a great job of showing the people that we might know who are not among our Facebook friends, they do a very poor job of finding people that we do not know, but may be valuable friends. There are over 900 million users of Facebook, and based on the social interactions that form around the contents shared, Facebook or third party applications should be able to suggest people who have similar tastes, ideologies, and/or believes to each other to be friends. We believe that the social interactions network is able to identify these people based on how they have been interacting with each other through the content shared in a given context on Facebook. Our API provides data to create this functionality quite easy. Since, we know the context that users are interested in (i.e. the Community) and based on the corresponding social interaction networks used we can find people who share similar interactions on the community shared content.

Better News Feed: People spend hours and hours on Facebook every day. However, they are only bound to see the posts shared by their immediate friends and the pages they

Community	Posts	Comments	Likes on Posts	Likes on Comments	Fans	Avg Path Length	Clustering Coefficient
Against SOPA	5690	8564	3864	15529	117000	3.3	0.172
Chase Community Giving	196	27231	4410	40481	3300000	2.3	0.331
Citi Bank	204	2801	4257	1307	300000	1	0.362
Jay Leno	41153	42236	50789	43042	423000	2.4	0.271
Chase Slates	105	229	1705	127	58000	4.4	0.218

TABLE I: Shows the different social interactions networks formed around the contents shared on different public pages on Facebook.

have liked. Through our API, it is possible to see what kinds of posts the user has been interacting with and find similar posts based on the SIN formed around it that the user has not interacted with. This will create a more dynamic newsfeed rather than the current one where users see the same posts over and over again throughout the day. Again, we know the context that the user is interested in, and using the corresponding social interactions network we can identify which posts the user would be interested in but has not interacted with yet. Therefore, the user will only see posts that he/she has not seen before and the content is related to what he/she likes. The social interactions network constantly changes based on user's behavior. Therefore, we can even show relative content to users current mood depending on how they have been interacting with the content shared recently. Figure 4 shows how SINs change over time.

Social Search: Social Search[18] is one of the hottest areas in the market and companies like Google, Facebook, and Microsoft are spending billions of dollars in the race of building the best social search experience. We believe that the SINs formed around the content shared on these page and groups give better results when combined with a search engine than the friendship networks currently used. While a group of users have very similar and close interactions around the content shared on Facebook, we can use this information when a person from this group queries something. We know the group's interests and that will help us serve the user with better social search results. Since there is a cap on how many friends users can have on Facebook, the social search will be limited to the number of direct friends. In addition to the limited social network, there are no guarantees that users immediate friends will share the same taste, thought process, or needs. In our approach we link users with many interactions on related content to provide better search results. Based on the query we can identify the context and use the matching social interactions network to find related content.

V. SIN API

We have adopted the same interface as Facebook's Graph API, which makes it easy for third party applications to use. We introduce the following API calls to enable third party applications to interact with the social networks formed around the content shared on Facebook.

Definition:

A **Community** on Facebook can be one of the following: Page, Group, or User. Usually, every community has an owner or an admin who keeps the community active. Each community usually defines a context around which people share content. Then the users interact with each other through the content that is shared on a given community by liking, leaving a comment, or re-sharing the content.

A **Post** is anything that is shared on Facebook. It could be a simple text message, a link to a third party web site, an image, or a video. There are many pages and groups with millions of members. It is amazing that posts in popular pages and groups get tens of thousands of likes and comments and hundreds or thousands of shares. The SINs that form around these posts are very large and have been neglected for the most part.

Our code is done in PHP. Developers will need to use our SDK instead of the one Facebook provides and once an application creates an instance of the Facebook class and assign it to the $facebook object as they would with a normal Facebook application using their Graph API, they can simply use our added api functionalities by calling the following methods:

$facebook→api('/faith/{Post-ID}', [$limit]) - This call returns the Social Network that is created by the interactions of users around a single post. We first retrieve general information about the post itself and then we iterate through the likes, comments and shares related to the post. We create the SIN around that single post and return the results to the user. We give different weights to different actions that have been taken on the post. Shares have the highest weight on the link from the person who shared the content to the person who posted the content originally. Comments have lower weight than shares but higher than likes on the link created from the person who left a comment to the person who posted the original content. Finally, likes have the least weight on the link from the person who liked the content to the person who originally posted the comment. Figure 5 shows a complete interactions network around a single post made by a page on Facebook.

$facebook→api('/faith/{Post-ID}/comments', [$limit]) - The Social Graph returned by this call contains only links created by comments around the given post. Basically, a link in this graph simply means that a user has left a comment on the content. This Graph is a star shaped graph. The weight of the links depends on how many comments each user has left on the given post. The higher the number of comments

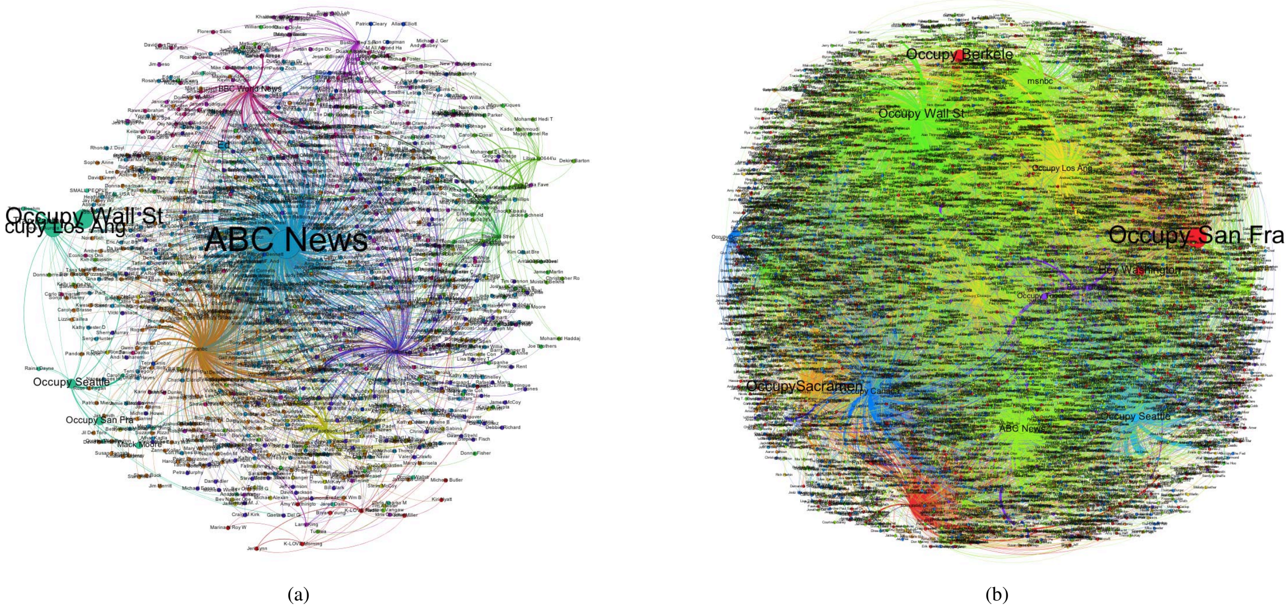


Fig. 4: Shows the interactions around 49 different Facebook public pages. The users have interacted with other users on at least two different communities. (a) Shows the interactions that have taken place during a one week period exactly a week after the occupation movements started. (b) Shows the social interactions that have taken place during a one week period three weeks after the occupation movements started.

the stronger the link from the user to the middle of the star, which represents the originator of the post. Figure 6 shows the interactions done through comments on the same post as above. Fig 6(a) shows everyone who has left a comment on the post. Fig 6(b) shows only users that have left more than three comments.

\$facebook→api('/faith/{Post-ID}/shares', [\$limit])- The Social Graph returned by this call contains only links created by re-shares of the original content. Each link represents a re-share of the content between the person who has shared the content to the person who has originated the content. The Social Graph returned by this API is also star shaped. Facebook allows users to re-share posts on different places, such as their own wall, their friends wall, or a page or a group's wall if they have the permission. Therefore, the weight of each link depends on how many times a user has shared the content.

\$facebook→api('/faith/{Post-ID}/likes', [\$limit]) - This call is similar to the previous call, except it only returns the graph created by likes action. In other words the links in the Social Graph returned represent a like from one person to another. The Graph returned has a star structure where the originator of the content is in the middle and all other users who have liked the content are connected to the person in the middle only. Since, each user can only have one like on the content, the weights of all the links are the same in this graph. Figure 7 shows the interactions networked returned by this API call for a public post made on the "Against Stop Online Piracy Act SOPA" page.

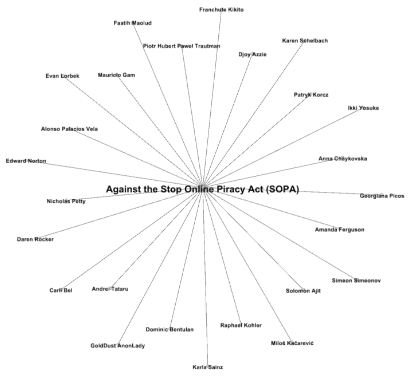


Fig. 7: shows a social interactions network of "likes" around a single post. Since, each user can only like a post once, all edges have the same weight. There were a total of 25 likes on this post at the time we crawled. Clusters are separated by the colors in the graph

\$facebook→api('/faith/{Community-ID}', [\$limit]) - This call will return the whole Social Network of all interactions around all the posts in the given community. For

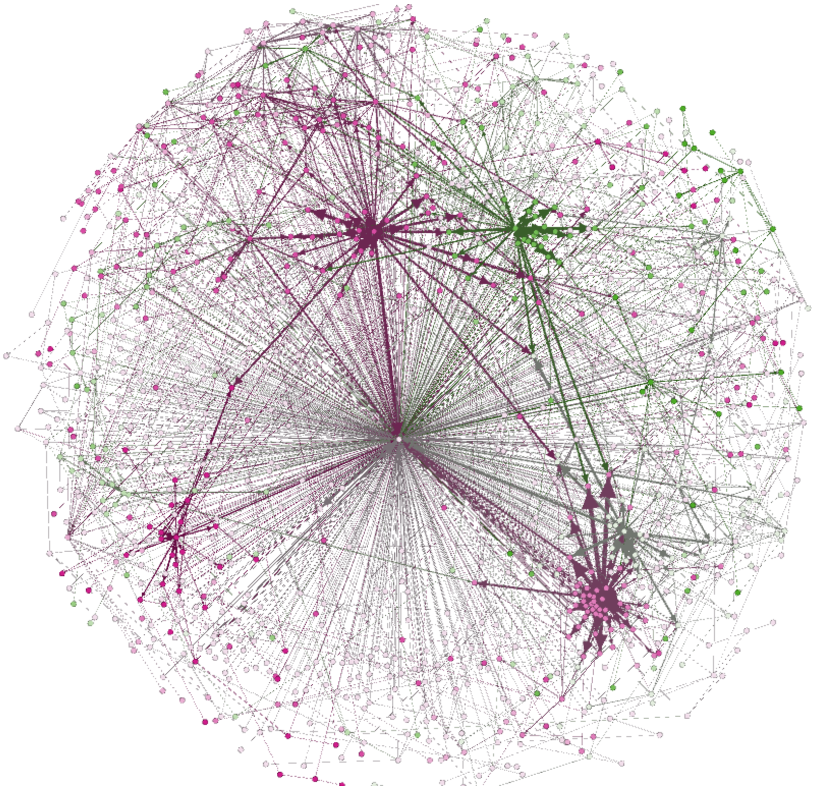


Fig. 5: Shows directed and weighted graph of a network of social interactions formed around a single post on a public Facebook Page. Clusters in the graph are identified by colors. The network consists of 1097 nodes and 2028 edges. There were a total of 25 likes on the post itself, 888 comments on the post and 1252 likes on the comments.

each individual post we make separate calls to receive all the comments, likes, and shares of that particular post. Much like the separate calls described above. Fig: 8 shows the entire network formed around the community page "Against Stop Online Piracy Act SOPA".

\$facebook→api('/faith/{Community-ID}/comments', [\$limit]) - This call is similar to api('/faith/Community-ID'), except we return the network that is created based on comments. In other words, We only return the portion of the network, where links are created by user comments and we do not take likes and shares into account. We have to send separate API calls to Facebook for each post in order to retrieve the comments, but it is still relatively faster than retrieving the whole graph since we do not need to send additional API calls for likes and shares.

\$facebook→api('/faith/{Community-ID}/shares', [\$limit]) - This call is similar to api('/faith/Community-ID'), but here we only consider the sharing of the **Community-ID** posts when we create the graph. From what we have seen in our datasets this network is considerably smaller than the likes network, which suggest that on Facebook it is more likely that people like a post than re-share it.

\$facebook→api('/faith/{Community-ID}/likes', [\$limit]) - This call is similar to api('/faith/Community-ID'). The difference is that we only look at the Social Community based on users likes. This requires fewer requests to the Facebook servers, since we do not need to retrieve the comments and shares any more; therefore, it is significantly faster than getting the whole graph. On the downside, we believe that the graph returned by this call is relatively weaker than the graph returned by the previous call; since, likes have the least weight among actions a user can take on a post.

The optional **\$limit** variable limits the number of items to be returned. Many times it is sufficient to receive a subset of the graph and are just interested in the latest interaction of users on a given context or content. Using the \$limit variable the third party applications have the ability to retrieve as much data as they need and not more. For the community API calls, the \$limit variable simply limits the number of posts returned and for each post we still retrieve the complete interaction data. For the post API calls, the \$limit variable simply limits the number of interactions taken place on the given post. The default value for \$limit is the same as Facebook, 25.

All these API calls return a response in JSON, which contains the weighted graph. We calculate the weight of the links based on the type of interaction (i.e. whether it's a like,

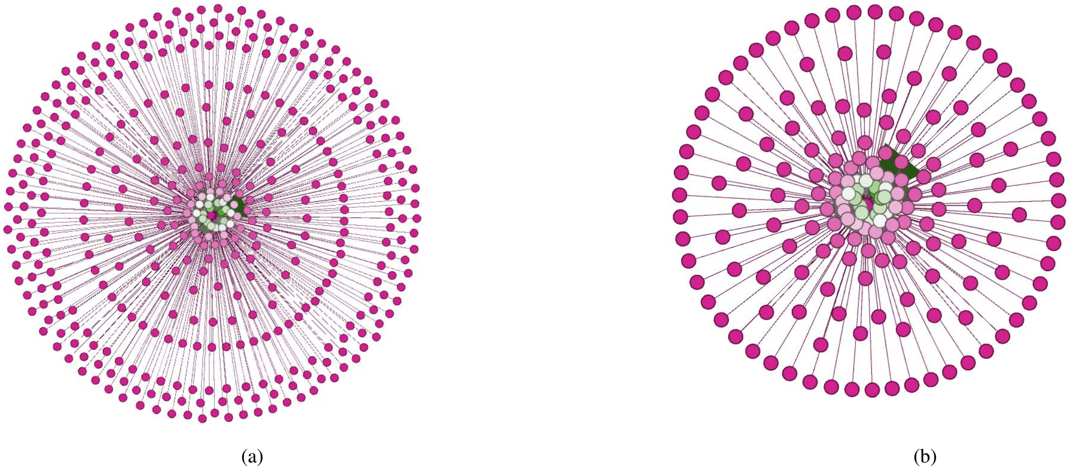


Fig. 6: shows the interactions done through comments on the same post. There were a total of 406 users interacting on this given post at the time the data was generated. The closer a node to the center of images means that the user has had more interactions on the post than the users that are further away from the center.(a) shows everyone who has left a comment on the post. (b) shows the people who have left more than three comments on the post. Applying this filter reduced the number of remaining users to 176.

comment, or a share) and the number of interactions between users. The graph returned is a directed Graph as opposed to Facebooks Social Network which is an undirected graph. We also include the timestamp on when each of these links were created, which allows us to recreate the whole social interaction network graph through time.

VI. Security Issues and Implementation Challenges

Issues about security and privacy of user data is a cause of major concern in online social network API development as discussed in [19]. In our current implementation, we consider the related issues and our API only fetches public data. As part of our continued research efforts, we are currently looking at methods to anonymize the SIN returned in order to protect the privacy of users who have interacted in a given community. The major cost to select an algorithm that can successfully anonymize the data is based on the algorithm's effectivness on preserving the original graph properties during the anonymization step [20][21][22]. A more detailed explanation of our future solutions is out of the scope of this paper.

Since, Facebook does not allow applications or platforms to store any of its data, we would need to get all the information we need through the API calls on the go. For API calls that try to get the structure of the whole community this requires a lot of calls depending on the amount of interactions on the page. One way for us to make things faster would be to use more threads and make our Facebook API calls in parallel with more nodes and save as much time as possible. We first make an initial call to get all the posts shared by a community and then in order to get the details of each post (i.e. likes, comments, likes of comments, and shares) we would make requests in parallel[23]. Also the amount of data retrieved with each Facebook API request is very limited due to their Paging mechanism[24]. For example, by default each api request to get posts of a Community only returns 25 results. In order to get the next 25 results one would need to make another API request to Facebook's servers. For comments, likes, and shares, each call only returns 50 items and in order to get all of the items one would need to make many requests depending on how many interactions that have taken place on a given post.

As of today, there is a bug in Facebook's API for retrieving information about re-shares of a post. We currently are not able to provide this data because of this bug[25]. We believe that once Facebook fixes this bug, our API should be able to retrieve the re-sharing data correctly.

VII. Evaluation

We have taken many steps in order to deal with software failure while generating data from Facebook. There are time-out errors, when using a browser, the browser might time out, or PHP execution time may exceed the server configuration. These timeouts could be increased from the default value, but API errors due to making too many API requests too quickly, or any other server errors on Facebook's side is harder to handle. At any time, we keep track of where we are in the process of data generating, so in case of a software failure we can simply continue fetching from where we left off instead of restarting from the beginning.

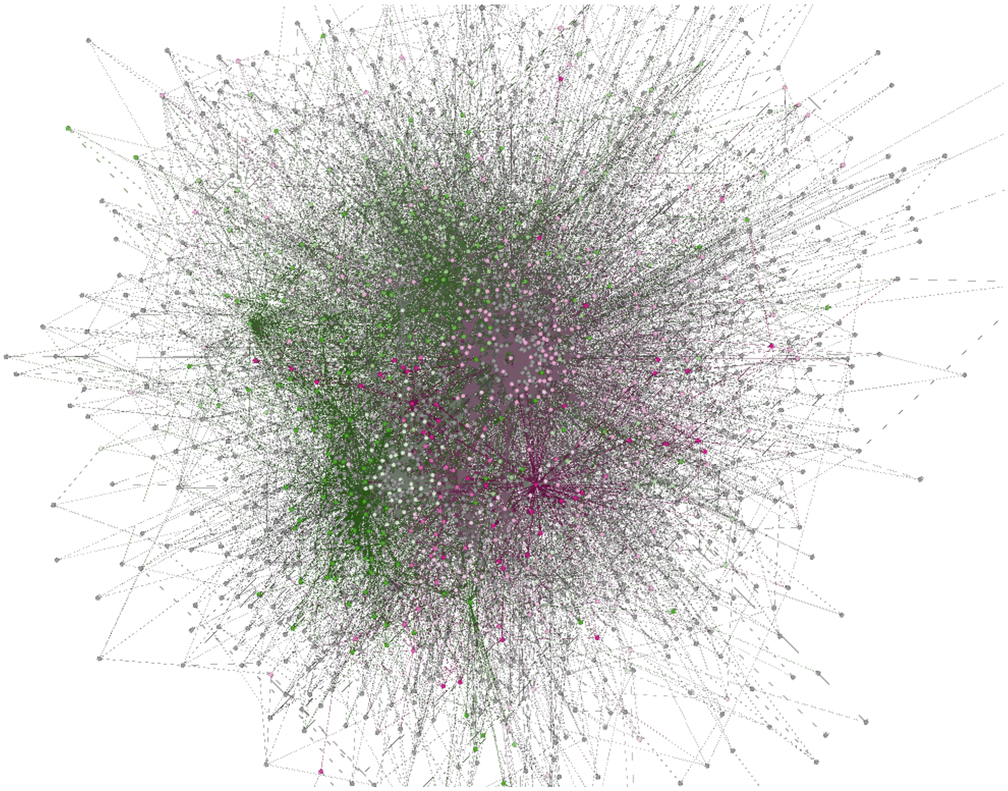


Fig. 8: shows the complete social interactions network of the public Facebook page, “Against Stop Online Piracy Act SOPA”, over all the contents shared on the page. At generation time there were 5690 posts, 3864 likes directly on the posts, 8564 comments to the posts and 15529 likes on the posted comments. Clusters are separated by colors. For better visibility all nodes with a degree of lower than six have been removed from the graph; hence; the above graph contains 16% of the total nodes of the original graph and 36% of the total edges of the original graph.

For example, processing the community page of Against Stop Online Piracy Act SOPA, which is a public page, with over 117 thousand members crashed 12 times over a 20 hour period. Table II shows the number of pages posts fetched during each run and how long the code ran before failure. We fetched likes, comments, shares, and likes of comments of over 6000 posts of this community over a 20 hour long time period, which suggests that on average each post on this page takes on average 11 seconds to be fully fetched.

We use two phases for the API calls to generate the SIN of the whole community. The first phase is a sequential phase, where we keep making API calls to Facebook in order to get the full list of posts shared on the given community. This phase gives us an idea about the interactions around each post (i.e. likes count and comments count are given). In the second phase, we try to divide and balance the posts among different machines as much as we can based on the amount of interactions done on posts and crawl the posts in parallel. Table III shows how much time we have saved during the second phase of crawling using 4 and 16 machines on Emulab instead of running the phase sequentially.

Round #	Duration (secs)	Posts crawled
Round 1	3307	393
Round 2	1410	188
Round 3	2	0
Round 4	2	0
Round 5	2	0
Round 6	12380	994
Round 7	7086	885
Round 8	5324	615
Round 9	7866	642
Round 10	9229	910
Round 11	10859	615
Round 12	8753	840
Round 13	5046	397
Total	71267	6479

TABLE II: This table shows the details of how many posts were successfully crawled before a software failure occurred. After each failure, the new round automatically starts again. Over a 20 hour long period we were able to crawl the data regarding interactions that had happened around 6479 posts shared on Against Stop Online Piracy Act SOPA fan page.

Community	Run Type	Average Time (s)
EU Community (3041)	Sequential	12’460
EU Community (3041)	Parallel (10 threads)	1’166
Milwaukee Bucks (5400)	Sequential	79’897
Milwaukee Bucks (5400)	Parallel (10 threads)	5’189
New York Knicks (66020)	Sequential	976’864
New York Knicks (66020)	Parallel (10 threads)	65’563
Jay Leno (41152)	Sequential	179’636
Jay Leno (41152)	Parallel (10 threads)	16’320

TABLE III: Shows how much time we saved during the second phase of our crawler by using a parallel approach with 10 threads.

VIII. Future Work

We plan to find more ways to create social interactions networks. Other than liking, commenting, and sharing posts users interact in many other ways on OSNs. Users can also send content to each other through a message on Facebook. Currently, we do not crawl these interactions which might give valuable results. Other interesting examples are the relations on how users interact through third party applications built on top of Facebook’s platform. Obviously, we cannot crawl this data using Facebook’s API, but more traditional ways of crawling, such as parsing the html of the applications, might be used in order to extract this information. Another feature is the tagging done in Facebook. Everyone who shares a posts on Facebook or leaves a comment can tag their immediate friends in the post. This is another indication of interaction between users that we would like to consider in future versions of SIN.

We are planning on using our API to create applications that leverage the results. We talked about some of the ideas for applications in previous sections. Social Search Engine, Friend Suggestion, and a Dynamic News Feed are among the projects that we are planning to build using our API; furthermore, we would like to enhance some of our previous projects, such as the TrustWiki [26] application under FAITH, that relied on the traditional friendship networks by using the SIN networks. We believe that using the social interactions network will provide much more accurate information than the social networks provided by the current API.

IX. Acknowledgements

This paper is based upon work supported in parts by the National Science Foundation under Grant No. CNS-1152320, IIP-1161015, and CNS-0832202, Army Research Office under the Multi-University Research Initiative (MURI) grant W911NF-07-1-0318.

References

[1] T. Cheredar. (2012, Feb.) Facebook user data. [Online]. Available: http://venturebeat.com/2012/02/01/facebook-ipo-usage-data/

[2] L. A. Adamic, O. Buyukkokten, and E. Adar, “A social netowrk caught in the web,” in *First Monday*, 2002.

[3] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, “Analysis of topological characteristics of huge online social networking services,” in *Proceedings of the Inetnational World Wide Web Conference*, 2007.

[4] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and analysis of online social networks,” in *Proceedings of Internet Measurement Conference*, 2007.

[5] R. Kumar, J. Novak, and A. Tomkins, “Structure and evolution of online social networks,” in *Proceedings of KDD*, 2007.

[6] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, “Growth of the flickr social network,” in *Proceedings of WOSN*, 2008.

[7] A.-L. Barabasi and R. Albert, “Emergenceof scaling in random networks,” in *Science 286*, 1999.

[8] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. Stanley, “Classes of small-world networks,” in *Proceedings of National Academy of Sciences*, 2000, pp. 11 149–11 152.

[9] J. Leskovec and E. Horvitz, “Planetary-scaleviews on a large instant-messaging network,” in *Proceedings of the International World Wide Web Conference*, 2008.

[10] H. Chun, H. Kwak, Y. H. Eom, Y. Y. Ahn, S. Moon, and H. Jeong, “Comparison of online social relations in volumes interactions: a case study of cyworld,” in *Proceedings of Internet Measurement Conference*, 2008.

[11] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, “User interactions in social networks and their implications,” in *Proceedings of the 4th ACM European conference on Computer systems*, 2009.

[12] S. Garris, M. Kaminsky, M. Freedman, B. Karp, D. Mazi’eres, and H. Yu, “Re: Reliable email,” in *Proceedings of NSDI, San Jose, CA*, 2006.

[13] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, “Sybliguard: defending against sybil attacks via social networks,” in *Proceedings of SIGCOMM*, 2006.

[14] C. Bird, A. Gourley, P. T. Devanbu, M. Gertz, and A. Swaminathan, “Mining email social networks,” in *MSR*, 2006, pp. 137–143.

[15] R. Nia, C. Bird, P. T. Devanbu, and V. Filkov, “Validity of network analyses in open source projects,” in *MSR*, 2010, pp. 201–209.

[16] R. Lee, R. Nia, J. Hsu, K. N. Levitt, J. Rowe, S. F. Wu, and S. Ye, “Design and implementation of faith, an experimental system to intercept and manipulate online social informatics,” *Social Network Analysis and Mining, International Conference on Advances in*, vol. 0, pp. 195–202, 2011.

[17] CNN. (2011, Nov.) California campus police on leave after pepper-spraying. [Online]. Available: http://edition.cnn.com/2011/11/20/us/california-occupy-pepperspray/index.html

[18] P. Bhattacharyya, J. Rowe, S. F. Wu, K. Haigh, N. Lavesson, and H. Johnson, “Your best might not be good enough: Ranking in collaborative social search engines,” in *7th International Conference On Networking, Applications and Worksharing*, 2011.

[19] F. Erlandsson, M. Boldt, and H. Johnson, “Privacy threats related to user profiling in online social networks,” in *SocialCom/PASSAT*, 2012, pp. 838–842.

[20] K. Liu and E. Terzi, “Towards identity anonymization on graphs,” in *In Proceedings of ACM SIGMOD*, 2008, pp. 93–106.

[21] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh, “k-anonymization of social networks by vertex addition.” in *ADBIS (2)*, vol. 789, 2011, pp. 107–116.

[22] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, “k-symmetry model for identity anonymization in social networks,” in *EDBT*, 2010, pp. 111–122.

[23] D. H. Chau, S. Pandit, S. Wang, and C. Faloustos, “Parallel crawling for online social networks,” in *Proceedings of the 16th international conference on World Wide Web*, 2007.

[24] J. Bowen. (2012, Feb.) Paging with the graph api and fql. [Online]. Available: http://developers.facebook.com/blog/post/478

[25] FacebookBlog. (2012, May) Shares count on facebook graph api not available. [Online]. Available: http://developers.facebook.com/bugs/289679207771682/

[26] H. Zhao, W. Kallander, T. Gbedema, H. Johnson, and S. F. Wu, “Read what you trust: An open wiki model enhanced by social context,” in *SocialCom/PASSAT*, 2011, pp. 370–379.