ON SOCIAL INTERACTION METRICS
SOCIAL NETWORK CRAWLING BASED ON INTERESTINGNESS

Fredrik Erlandsson

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2014:06
Department of Computer Science and Engineering
On social interaction metrics
social network crawling based on interestingness

Fredrik Erlandsson
On social interaction metrics
social network crawling based on interestingness

Fredrik Erlandsson

Licentiate Dissertation in
Computer Science

Department of Computer Science and Engineering
Blekinge Institute of Technology
SWEDEN
"A squirrel dying in front of your house may be more relevant to your interests right now than people dying in Africa."

Mark Zuckerberg, 2011.
Abstract

The use of online social networks poses interesting big data challenges. With limited resources it is important to evaluate and prioritize interesting data. This thesis addresses the following aspects of social network analysis: efficient data collection, social interaction evaluation and user privacy concerns.

It is possible to collect data from most online social networks via their open APIs. However, a systematic and efficient collection of online social networks data is still challenging. Results in this thesis suggest that the collection time can be reduced to 48% by prioritizing the collection of posts.

Evaluation of social interactions requires data that covers all the interactions in a given domain. This has previously been difficult to do. In this thesis we propose a tool that is capable of extracting all social interactions from Facebook. With the extracted data it is for instance possible to illustrate interactions between different users that do not necessarily have to be connected. Methods using the same data to identify and cluster different opinions in online communities have been developed and evaluated.

The privacy of the content produced and the end-users’ private information provided in social networks is important to protect. Users should be aware of the privacy-related consequence of posting in online social networks in terms of privacy. Therefore, mitigating privacy risks contributes to a secure environment and methods to protect user privacy are presented.

The proposed tool has, over the period of 20 months, collected 38 million posts from public pages on Facebook which include, 4 billion likes and 340 million comments from 280 million users. The data collection is, to the best of our knowledge, the largest research dataset of social interactions on Facebook, enabling research in the area of social network analysis.
This thesis consists of four articles that have been submitted, peer reviewed and published in conferences. The thesis also contains a submitted journal article. The articles have been written together with other colleagues from Blekinge Institute of Technology and University of California Davis. The thesis material has appeared in the following publications:


5. Fredrik Erlandsson, Martin Boldt, Henric Johnson, S. Felix Wu ”Interaction metrics to support crawling prioritization in online social networks“, Submitted to Information Sciences June 2014.
Publication 1 deals with privacy issues identified by the authors, in which the thesis author is the main driver. Publications 2 and 4 are related as they form part of the motivation for the data collection process discussed in publication 5. For both publications 2 and 4 the thesis author contributed with the dataset, experiment design and the development of the SINCERE search engine shown in 8.3.5. The thesis author were highly involved in the writing of publication 2, together with the co-authors. Publication 3 is a pre-study for publication 5 with the thesis author as the main driver and contributor of the material. For publication 5, the thesis author was the main driver, conducting and developing experiments and tools. The thesis author is also the principal driver of the writing, together with the senior co-authors.
# Contents

Abstract ................................................................. i
Preface ................................................................. iii

1 Introduction ......................................................... 1

2 Background ......................................................... 5
   2.1 Related Work ................................................. 6
   2.2 Terminology ................................................. 7

3 Approach ............................................................ 9
   3.1 Aim ............................................................. 9
   3.2 Scope .......................................................... 9
   3.3 Research Questions ......................................... 9
   3.4 Research Methodology ...................................... 11

4 Results ............................................................... 13
   4.1 Contributions ............................................... 13
   4.2 Discussion .................................................... 16
   4.3 Conclusion ................................................... 17
   4.4 Future Work ................................................ 18
   4.5 References .................................................. 18

5 Privacy Threats Related to User Profiling in Online Social Networks .......................... 23
   Fredrik Erlandsson, Martin Boldt, Henric Johnson
   5.1 Introduction .................................................. 23
8 The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

Teng Wang, Keith C. Wang, Fredrik Erlandsson, S. Felix Wu, Robert Faris

8.1 Introduction ................................................. 66
8.2 Related Work ................................................. 67
8.3 Opinion Classification ........................................ 69
8.4 The Influence of Feedback with Different Opinions on User Continued Participation ........................................ 77
8.5 Discussion and Conclusions ................................... 82
8.6 Limitations and Future Work ................................. 83
8.7 Acknowledgment .............................................. 84
8.8 References .................................................. 84

9 Interaction metrics to support crawling prioritization in online social networks

Fredrik Erlandsson, Martin Boldt, Henric Johnson, S. Felix Wu

9.1 Introduction .................................................. 89
9.2 Related Work ................................................ 91
9.3 Terminology and Application ................................ 93
9.4 A Platform for Making Interactions Accessible .......... 94
9.5 Interestingness of Posts ..................................... 104
9.6 Evaluation of Interestingness ................................ 107
9.7 Discussion .................................................... 112
9.8 Conclusion .................................................... 115
9.9 References .................................................... 115
Online Social Networks (OSNs), also called Social Media, such as Facebook, Twitter, Instagram and Google+, are attracting increasingly more interest from Internet users. With the possibilities of being connected and interacting with each other anytime and anywhere, OSNs influence peoples’ daily routines and everyday behaviors. For instance, 48% of US adults between 18 and 34 check Facebook the first thing when they wake up [1]. In addition, Facebook alone increased its number of users with 22% between 2012 and 2013, and in January 2014 there were 1.31 billion active users [1]. In total, the Internet has 2.94 billion users, i.e., 44% of the users of the Internet are active Facebook users.

Apart from changing the way people interact and communicate, OSNs also provides novel means of news aggregation. Today it is possible to stay in touch with the latest news from the world by following certain people and newsgroups within OSNs, i.e., the need of watching the news or reading the newspaper to keep up with the world is emerging to just checking the OSN feed. Moreover, with the growing use of OSNs the democratic powers has also changed. By using OSNs like Twitter and Facebook you do not have to be a reporter on a newspaper or television to form opinions and reach a critical mass. With OSNs everyone has the means of publishing thoughts and opinions as a citizen journalist. There are many examples of this; the most widespread and discussed is the role social media played in Arab Spring [2–4] of 2010–present.

Gathering data and the corresponding user interactions from OSNs is becoming more and more interesting for researchers and businesses. The means of observing human behavior via OSNs have been called the social media lens by Zafarani et al. [5].
1. Introduction

This social media lens provides us with golden opportunities to understand individuals at scale and to mine human behavioral patterns otherwise impossible.

Zafarani et al. [5]

OSNs pose interesting big data challenges regarding storage, management and analysis of users’ online activities. In January 2014, Facebook stated that they are storing data of the magnitude of exabytes ($10^{18}$ bytes), and this number is steadily growing with roughly 9 million messages sent every hour. Nevertheless, the handling of storage and processing of this data is not the only challenge. There is a need to develop methods evaluating the meaning and semantical/informational value of the content, here referred to as interestingness (also described in Section 2.2). This challenge further entails studies on the efficiency of handling the informative content and the relations and interactions between the users.

Although the major OSN providers offer publicly available APIs to access their data, challenges still exist to collect data systematic and efficient. For instance, data from a post on Facebook can require a high number of requests to get full coverage of all interactions.

Another interesting aspect with data from OSNs is the fact that it is humans that produce the data, in contrast to synthetic data. Using this data enables research areas that were hard to realize just a few years ago, e.g., big-scale user interaction analysis [6, 7] and the creation of Social Interaction Networks (SIN) graphs [8]. A SIN graph shows the interactions between users in various communities and can for instance represent interactions of all users on one newsgroup or relating to a specific topic. This allows, for instance, researchers to develop novel applications related to social sciences.

In December 2009 Google introduced “Personal search for everyone”, enabling custom ranking by using 57 signals. This method of personal ranking is nothing unique for Google. Today most search engines and even OSNs are ranking the content based on personal information. For instance, Facebook have algorithms to personal your news feed and prioritizing posts that you most likely will interact on. This prioritization of content poses new problems, as the ranking algorithms tend to just show content in the direction of the user’s sympathies and interest. It is, therefore, hard to get a diverse picture and the opposing point of view, essential for democracy.
Pariser calls this “The filter bubble” and addresses it in detail in the book with the same title [9]. We argue that it is possible to address the issues of the filter bubble and to reintroduce diversity in the online world by letting the users manually configure the ranking method [10].

With the high number of users in OSNs, awareness of privacy related issues is also of importance. However, users of OSNs tend to be naïve in what information to reveal (on OSNs), as related to what users tend to reveal on other places (online) [11, 12]. As a consequence, the awareness of the privacy related threats needs to be addressed in a clear and structured way to aid users and raise the awareness.

In Chapter 2, the background and motivation are presented as well as the state of the art work. Chapter 3 presents the Aim & Scope together with the Research Questions. In Chapter 4, the contributions are presented and the results are discussed and concluded in Section 4.2 and 4.3 respectively. Finally, the proposed Future Work is presented in Section 4.4 and the publications are then published in Chapter 5–9.
Background

The computer era enables communication in new ways. The early ways of social communications using computers included bulletin board systems (BBS), USENET, America Online and CompuServe. The ancestor to the Internet as we know it, ARPANET which is considered the first packet switched computer network, was designed to enable easy civilian and military communication, mostly in the form of email. In the early stages of the Internet, social communication were mainly organized as chat rooms or simple webforums. It was first in the late 1990’s that social media sites emerged in the same form as today, enabling users to maintain a profile and create a community (much like today’s “friend list”). One of the first OSN sites was SixDegrees.com, 1997 to 2001 that was followed by Friendster and Myspace. In Sweden, a very large social network site was LunarStorm, active between 2001 and 2007. Early pages of social networking such as Friendster, Myspace and the Swedish page Lunarstorm have all retired and one of their successors today is Facebook. Social Media or OSNs provide means to enable users to connect with friends and share information, i.e., a digital way to mimic the real world communication. This is often made in form of a web page, but is also supplemented with a smartphone application. The information shared were intentionally only to be available for the closed group of the users’ friends (or network of friends).

Private information that is withheld can express an individual’s aim to uphold their privacy. Privacy is a way to limit the information of an individual and thereby express them selectively. The boundaries of users’ privacy vary with the individual’s background and culture. Often the privacy is a way to protect private information, sensitive to the user. There are a few different privacy threats including identity theft, surveillance, and online victimization.
2. Background

2.1 Related Work

There has, to the best of our knowledge, been a lack of analysis of OSNs with the aspect of interestingness, validity and usefulness of data gathered. The study of Peters et al. [13] summarizes previous studies made to apply metrics in order to manage social media. The work by Peters et al. differs from our aim since the authors try to address the issue of getting higher ranking for marketing purposes, while our work emphasizes calculating the interestingness as described by Geng et al. [14].

Many studies exist that either directly or indirectly cover the challenge of crawling various OSNs. The studies conducted by Mislove et al. [15] are, the largest OSN crawling study available. From four popular OSNs; Flickr, Youtube, LiveJournal and Orkut, 11.3 M users and 328 M links were collected. Moreover, indirect studies of OSN crawling are presented in the studies by Wilson et al. [6] and Crnovrsanin et al. [7], where the authors transverse user profiles from Facebook. They collected roughly 70% of user profiles from various regional networks at high speed (averaging 10 MB/s) with quite limited resources. However, this study was conducted in spring 2008 and since then Facebook redesigned their site and it is no longer possible to crawl user profiles. More recently, a study by Buccafurri et al. [16] discussed different methods to transverse the social network in a crawling perspective. Still, the restriction on crawling users profiles is not an issue in this work, since we gather data from public groups only. As such, our work has substantial data to crawl and our challenge differs from Buccafurri et al. [16].

Analysis of user interactions on OSNs has been a topic for several years. Garton et al. [17] identified the connection of people via computer networks as social networks in 1997. The area of various types of OSN are comprehensively described in [18]. Interesting studies include the studies by Grabowicz et al. [19] where the authors apply and evaluate social theories on OSNs. Also the studies by Ferrara et al. [20] are interesting as it maps topology models on various social networks.

Studies to classify data include Linguistic Inquiry and Word Count (LIWC) [21], which is a transparent text analysis program that counts words in psychologically meaningful categories. With LIWC it is possible to show attentional focus, emotionality, social relationships, thinking styles, and
2.2 Terminology

Data collection addresses how data can be crawled in a systematic fashion. It is important to address means of identifying data to be collected as interesting in the collection process. Interestingness is a term commonly used in the context of data mining, as a measure of identifying and ranking data. A pattern is said to be interesting if it is valuable and the mining resources can be motivated. Commonly, the interestingness measure is used to emphasize conciseness, coverages, reliability, peculiarity, diversity, novelty, surprisingness, utility, and actionability [14]. These nine criteria are traditionally used to determine if data is of interest. Conciseness describes the ability for a limited set to still map and represent a full dataset. Given just a fraction of records, but with high coverage, it is possible to work with a large subset of the dataset. The dataset is reliable if it has a high ratio of a given relationship in applicable cases. A pattern is peculiar if it is far from other discovered patterns. Elements have high diversity if they differ significantly when compared. Novelty describes how much new information a record adds to the complete dataset. It is virtually impossible to actually predict novelty as there is no way to know what a record adds before mining it. Surprisingness indicates a pattern that was unexpected and unpredictable. With utility we investigate how much use a pattern is given a specific goal. Actionability is the means of a pattern in respect of how much it can contribute in terms of future decision making in the specific
2. Background

domain. A pattern’s interestingness is especially important to consider when it comes to collecting data.

Efficiency is a term used in this thesis and with efficiency in this scope we address ways of using as low amount of resources as possible. We argue that software is efficient if it uses significantly lower resources than other similar software.

When users communicate with each other it is called that they are interacting. In the context of OSNs, social interactions are often in a simple form, i.e., a user can often just click a button in order to interact with another user. This simple interaction can be used to indicate sympathy to a post or an ability to share one user’s text with another community.

In this thesis we are addressing OSNs with a focus on Facebook. In the scope of Facebook there is some terminology that needs explanation and clarification. Each user on Facebook has a number of friends. Users create a mutual agreement of relationship. When a user writes something on her own profile, it is called that she posts on her wall. A user can also follow a newsgroup. On Facebook these groups are called pages, and when one follows a page it is said that the user likes that page. It is also possible for a user to post on her friend’s wall or on a page’s wall. A user does not need to like a page to be able to post to it. However, some pages have restricted their page, in which only page owners and selected users are allowed to post. The main page of Facebook is called the news feed or sometimes simply just the feed. This feed contains a subset of posts from the users’ friends. It also contains a limited subset of posts from the pages the user likes. On shown posts, each user have the ability to either like, give a “thumbs up”, comment, write a small comment or share the post with the users’ friends. It is also possible to like comments, which we are calling a comment-like.
3.1 Aim

This thesis aims to investigate interactions in OSNs and identify means of ranking posts made in public groups. Currently, methods to access a complete dataset of interactions in OSNs is lacking. Interactions and produced data need to be collected in a structured and efficient way. The nature of social media interactions follows a constantly growing pattern that requires selection mechanisms to find interesting data.

3.2 Scope

The studies conducted in this thesis are performed on Facebook, which is the largest OSN. The data collection process covers publicly available pages within a specific range of Facebook. The studies of social interactions are in the early stage and the conducted studies identify general interaction patterns.

3.3 Research Questions

The main question we explore in this thesis is: How can data be crawled from Facebook in a systematic and resource efficient way? While investigating this question other challenges have risen. First, users’ privacy must be considered. Second, if available resources are not sufficient for full retrieval it is of importance to perform prioritization, i.e., only crawl data that are of use to the current application. The main research question has been elaborated in four research questions covered in this thesis:

\footnote{as all the interactions corresponding to a specific post}
3. **Approach**

RQ I. *How can data from Facebook be collected with regards to depth, i.e., covering all interactions in a given domain*\(^2\)?

It is of interest to collect the data from OSNs, for analysis purpose. Most OSN sites of today have an API providing the ability to build tools to access information from the site. However, these APIs often provides just a sparse interface to the data and requires additional effort to connect the data and make it useful. We are interested in how a tool extracting data from, e.g., Facebook’s public pages must be designed to access data with aspect of covering all interactions.

RQ II. *How can prioritization be used to improve the data-collection process with regards to maximizing interaction coverage with respect to available resources?*

In OSNs like Facebook, Twitter and Google+ new data is created all the time. All this information is probably not equally useful and by crawling a selection of the data we can maintain the essence of interesting interactions.

RQ III. *Which privacy threats exist in OSNs and what measures can users take to protect their privacy?*

With the use of OSNs comes the potential threat of user privacy. Users of OSNs are often publishing information concerning themselves or people in close relation to the users. Users share various types of information of different level of sensivity; ranging from just sharing a general link or funny picture to information such as checking-in at places. It is of importance to identify potential threats and find ways to protect the privacy of the user by making it possible to “lock down” the information so only the intended recipient or recipients can access the information.

RQ IV. *How can user content and interactions on the collected data in OSNs be valuable?*

There exist a challenge in crawling and collection of interactions from OSNs. But once that information have been collected there must exist valuable use of the data. What type of applications can the crawled interactions be used for.

\(^2\)e.g. page
3.4 Research Methodology

The studies in this thesis are conducted in both quantitative and qualitative form. Quantitative research is conducted with a focused description and with a conclusive research. In quantitative research only measurable data is observed. In contrast, qualitative research have a broad description and with exploratory results. Qualitative research focuses mainly on verbal data rather than measurements. Gathered information is analysed in an interpretative manner, impressionistic, subjective or even diagnostic way.

As this work started with a broad question related to data collection, it could be argued that the work presented in this thesis is in the form of exploratory research. Further, applied research that require flexibility when approaching the problem is often referred to as exploratory research [26].

Case studies have been used in some articles. A case study is a type of observational research where observations are made of a phenomenon without interfering. The observations from a case study are conducted as an in-depth study of a particular situation. One problem with case studies is that it is not possible to fully answer a question, as it is not possible to know when all subjects are evaluated. Instead, a case study will give indications and allow further elaborations. On the other hand, one of the advantages with case studies is the fact that researchers are allowed to take new directions based on the study.

The tools developed to address the problems in this thesis regarding data collection and organization are implemented and evaluated by prototyping. The developed crawler is built to be resilient to failures and adaptable to external issues. The developed crawler and the tools supporting it are acting as a foundation for further studies for the research group at Blekinge Institute of Technology and University of California Davis and the objective is to share the tools and data with other researchers.

For the quantitative parts of the research, statistical methods have been used, including statistical tests of similarity and correlation. The results and conclusions are presented and evaluated based on significance with one tailed confidence interval. The datasets have been selected using random sampling of non-synthetic data.
3. Approach

3.4.1 Validity Threats

With exploratory research there is always a validity concern of the drawn conclusions, as the problem definition is allowed to change during the study. Actions have been taken by both manual and automatic verification of the results, in order to avoid this validity threat. E.g., the crawled data have been evaluated both against available data on Facebook’s webpage and against the data accessible via the API.

Further, as the studies conducted in this thesis are based on a self-developed tool, this may pose a validity threat as the results reflects the data collected by our own tool. To minimize this issue actions have been taken. For instance, the study in Section 9.7 is made on a randomly sampled dataset to minimize bias results. In addition, the gathered data have manually been verified to be accurate and complete.
The work presented in this thesis addresses means of getting a complete dataset of user interactions from Facebook. During the last two years our crawler have been crawling data enabling research with a comprehensive dataset. Currently data produced by 280 million Facebook users have been crawled. Covering 38 million posts with 340 million corresponding comments and 4 billion likes.

4.1 Contributions

This thesis covers the following contributions.

First, to address the challenge of collecting data from OSNs a study of designing a crawler capable of covering all interactions in a given page is presented in this thesis. The work presented in Chapter 7 acts as a detailed pre-study for the novel crawler presented in Chapter 9. The crawler is not just novel in the way it crawls posts to the full extent of all interactions, it is also efficient as it is built as a distributed system. This distributed system, with one main server and multiple active clients responsible for the interaction with Facebook, enables high crawling rate and support for additional clients whenever the system requires more capacity. This relates and addresses RQ I, in which we present how an efficient crawler can be implemented and evaluated.

Second, in Chapter 9, we argue for viable metrics how to evaluate the value of posts and corresponding interactions. This work is based on traditional data mining theories applied to the context of OSNs. We present eight novel metrics to support ranking of posts, of which we evaluate and argue that it is possible to rank posts based on a few basic parameters. The
4. Results

Figure 4.1: Interactions around the contents shared on several Facebook public pages in a period of three weeks. The depicted users have interacted with other users on at least four communities.

results suggested in this chapter answers RQ II.

Third, in Chapter 5, a study of potential user privacy threats, within OSNs are presented. The study discusses six major threats to the user’s privacy; OSN information leakage, friend-in-the-middle, trojan application, public information harvesting, social bot and friend-in-the-middle trojan application. We also conduct a proof-of-concept showing how public information harvesting can be used to create interaction profiles and how to profile users. This study is not only of description and demonstration purpose, we also show how users can protect themselves to the presented threats. This chapter addresses RQ III.

Finally, we present a Social Interaction Network (SIN), a way of representing social interactions in OSNs. With SIN it is possible to follow users activity among different groups and see how opinion moves. In addition,
4.1. Contributions

Figure 4.2: Snapshot of the webpage SINCERE, showing the search result for “Heard some of the LAPD”. The first post and its corresponding comments are visible. The comments shown are clustered in two opinion groups, where the left group are from negative users and the right group are from positive users.

SIN also enables studies of social interactions and visualizations. Figure 4.1 illustrate such a visualization showing how interactions around posts and comments of several public pages on Facebook are related. In Figure 4.1, the relationship between various media pages and the first three weeks of the occupy movement\(^1\) is shown. For illustrative purpose, have users interacting on less than four different communities been removed. This work is fully described in Chapter 6. This chapter is related to RQ IV, showing the use of gathered social interactions. Also, in Chapter 8 we present the use of user interactions for opinion classification and grouping. This work is conducted by looking at the corresponding like-graph of comments related to a post. Both Chapter 6 and 8 contributes and answers RQ IV, as we both

\(^1\)The occupy movement is a protest against social and economic inequality
visualize social interactions and show means of using them. In addition, a framework to make the crawled data available and searchable in the form of a webpage has been developed. Figure 4.2 shows a demonstration of the social search webpage SINCERE; where the user is able to search text from the crawled posts. One of the goals of SINCERE is to diversify information and tackle the filter bubble [9], allowing the user to manually control the search ranking. Currently SINCERE supports ranking by content, number of likes, number of shares and number of comments made on the post. It also supports two types of entropy ranking methods: user entropy and post entropy. Entropy in this context reflects on the level of information novelty and diversity. The comments corresponding to the search result are clustered in two columns based on the users’ opinions classification and grouping, presented in Chapter 8. In Figure 4.2 the comments from users identified as negative are to the left and from positive users are to the right.

4.2 Discussion

Currently no methods exist to access the data corresponding to the complete interactions around posts\(^2\) from OSNs. There are even indicators that the OSN providers themselves do not have easy access to this data and even if the data exists it is hard to extract it. For instance, Facebook have powerful tools to select information and advertisements for its users. However, methods for extraction of the complete interactions are not available, through the API.

This work is limited to cover interactions around open pages on Facebook. Currently there is a gap as it is not possible to get interactions from a particular user in a specific timespan. The work presented in this thesis bridges this gap and enables researchers access to social interaction data.

As Facebook can register everything of the users’ actions; we are able to novelty collect the users’ actions in public pages. This collected information are organized within the SINCERE framework and made available. The way the data is structured and organized enables the research community to study patterns and behaviors of users. Do note that due to concern of individuals’ privacy no studies of single user behavior are conducted. With

\(^2\)complete interactions refers to all actions users have taken on a specific posts, including; likes, comments, shares and likes on comments
4.3 Conclusion

This thesis investigates how data from OSNs can be gathered and utilized. Different approaches to gather interactions from Facebook are investigated. First, a simple approach with a single client that gathers data in sequence is evaluated. Second, the crawling is extended to a distributed system with high error tolerance. Third, a study of how to improve the efficiency of data collecting process, with respect of gathering as many interactions as possible, is conducted. The crawled data is made available through a novel framework; both in the form of theoretical design guidelines and the openly available APIs of SINCERE and as SINs. Enabling future research in the field of computer science but also in social sciences, where the vast number of interactions between different users and communities could be further studied. The presented findings on prioritization of posts based on interestingness shows that it is possible to reduce the crawling time by 48.5\%, while still covering 99.5\% of all interactions. Methods of making use of the gathered data are also presented in the form of user-like, similar to SIN, to classify different opinions in user interactions.

The thesis further addresses different privacy threats that users of OSNs are exposed to. One of these threats is previously undocumented, user profiling based on the activity in publicly open groups. It is proven that with limited resources it is possible to profile users within an OSN through

our data it is possible to make the data available, as shown in our webpage illustrated in Figure 4.2 with means of introducing diversity in the presented results, i.e., to burst The Filter Bubble [9]. It is also possible to create interaction networks and graphs illustrated in Figure 4.1.

The findings of reduced crawling time by prioritization of highly interesting posts should also be investigated further. Prioritization has the advantage of putting stronger emphasis on information with higher interestingness, while disregarding the less interesting items. The likely disadvantage is that some of the disregarded items assessed as not interesting may in fact carry information of high interestingness. Furthermore, studies of identifying and further evaluating means of prioritizing posts are needed. Only a few of the proposed means to prioritize posts identified in Section 9.5 are fully evaluated and this requires more work.
the activity in open groups and then build a social interaction graph of
their interactions. Any user within the OSN is vulnerable to this threat,
independent on their privacy settings. Finally, we suggest a number of
different protection mechanisms against the threats identified.

4.4 Future Work

The developed crawler is currently capable of collecting data on Facebook.
It would be interesting to enhance the crawler to cover other OSNs as well.
In addition, the current stage of the crawler requires manual input of pages.
Therefore, work to extend the crawler to automatically discover pages to
crawl will be conducted.

Finally, studies to further investigate the shared interactions on the
content are an interesting field with unlimited opportunities for future
studies. Interesting studies include, but are not limited to: a study of
time distribution of interactions per communities and gender. A study
to determine tendencies and trends of community route path, i.e., how
users tend to move between different communities. This study also aims to
identify influential users, including the trend of user intensity and the top
users with most activity. Based of the comprehensive dataset, studies to
validate social science and humanity research based on social interactions
are also interesting.

4.5 References

statisticbrain.com/facebook-statistics/.

Mazaid. Opening Closed Regimes: What Was the Role of Social Media
php/2011/09/11/opening-closed-regimes-what-was-the-role-
of-social-media-during-the-arab-spring/.

[3] H. H. Khondker. “Role of the New Media in the Arab Spring”. En-
14747731.2011.621287.


4. Results


4.5. References


Privacy Threats Related to User Profiling in Online Social Networks

Fredrik Erlandsson, Martin Boldt, Henric Johnson

Abstract

The popularity of Online Social Networks (OSNs) has increased the visibility of users’ profiles and interactions performed between users. In this paper we structure different privacy threats related to OSNs and describe six different types of privacy threats. One of these threats, named public information harvesting, is previously not documented so we therefore present it in further detail by also presenting the results from a proof-of-concept implementation of that threat. The basis of the attack is gathering of user interactions from various open groups on Facebook which then is transformed into a social interaction graph. Since the data gathered from the OSN originates from open groups it could be executed by any third-party connected to the Internet independently of the users’ privacy settings. In addition to presenting the different privacy threats we also we propose a range of different protection techniques.

5.1 Introduction

In the beginning of 2012 Facebook had about 800 million users and the company was valued to over 100 billion dollars which to large extent originate from advertisement and user profiling possibilities based on user interaction. Besides Facebook there are a number of different Online Social Networks (OSNs) that has reached a considerable user-base, e.g. Google+, Twitter and LinkedIn.

It is therefore important to address the privacy implications of how the published information within OSNs is handled. Information that is published
by users within a limited group, or perhaps shared with a single user is often of a nature that can cause significant inconvenience, or even harm to concerned users. As OSNs grow in size the methods and knowledge among its users about how to configure privacy settings is crucial. In this paper we list different privacy threats within OSNs together with potential protection mechanisms. In addition to this we also add a new privacy threat that originates from scraping publicly available information which is published in open groups within the OSN.

OSNs like Facebook, Google+, and Twitter all provide open interfaces (i.e. APIs) for third-party applications to interact with the OSN by accessing and publishing data. This is very convenient for the user as it opens up possibilities for value increasing applications to interact directly with the social network. Consequently, there are more than 500,000 third-party applications, such as online games that interact and coexist with Facebook [1].

What people does not reflect upon is the fact that most of these applications have the abilities to interact with the OSN on behalf of the user, which also includes the possibility to gather information that the user posted as private correspondence. Add to this that users on OSNs share information that could be harmful for the user itself, or even the user’s friends. As an effect Trojan applications that use deceptive and covert behavior can gather such sensitive information from users. However, Trojan applications can also retrieve information among a user’s friends including their posts, which threaten the privacy of the OSN users.

5.2 Privacy Threats

In this section we will present six different types of privacy threats illustrated in Fig. 5.1. All of these threats result in user information leakage from the OSN to third parties. These privacy threats exist because social information about OSN users has a value, and can be refined into revenues within the context of targeted advertisements etc.

5.2.1 OSN Information Leakage

The first type of privacy threat, illustrated in case (a) in Fig. 5.1, is based on that the owner of a OSN, e.g. Facebook or Google, continuously gather detailed information regarding users activities within the OSN. This is
probably the most obvious privacy threat and as such it is well known within research community and it is also the threat that OSN users first come to reflect upon [2, 3]. We therefore expect OSN users to understand that information they share within the OSN, e.g. user profile content, messages, and photos, can be mined, refined and sold by the owner of the OSN. Exactly how the OSN owner is allowed to use and benefit from this information is regulated within policy documents, e.g. the statement of rights and responsibilities [4] and the data use or privacy policy [5] for Facebook. A problem is that a large extent of OSN users don’t reflect upon how their interaction within OSNs affect their privacy, which could be a threat to their privacy [6]. As a natural consequence these users do not bother to investigate the content of the OSN policy documents.

There is also a risk that the OSN infrastructure get compromised, giving third parties unauthorized access to sensitive information [7].

5.2.2 Friend-in-the-Middle Threat

Case (b) in Fig. 5.1 shows a type of privacy threat where user information is leaked through a trusted friend within the OSN. Because of this threat the OSN infrastructure often provide users the possibility limit their posts and information spread to smaller group, which (if used correctly) could be used as one method for avoiding public scrutinization. Unfortunately a chain is not stronger than its weakest link, which goes for friendships within OSNs as well. A large portion of OSN users act irresponsible by more or less allowing anybody to establish a friendship, which not only affect the user but potentially also that particular user’s friends.

One must also consider the current state of social gaming, where users require a certain number of friends in order to achieve certain tasks (level up) [8]. This tend to cloud users’ judgements regarding whom they are accepting as friends, as they instead focus on the primary task ahead, i.e. leveling up.

5.2.3 Trojan Application

The third type of privacy threat is associated with Trojan applications leaking information about its OSN users to third parties, see Fig. 5.1 case (c) [9]. The user is deceived to install a Trojan application which claims
5. Privacy Threats Related to User Profiling in Online Social Networks

Figure 5.1: Six different privacy threats within OSNs, where (a) is leakage from the OSN infrastructure to a third-party, (b) a friend-in-the-middle threat, (c) a Trojan application, (d) is public information harvesting, (e) is a socialbot, and finally (f) that represent a friend-in-the-middle Trojan application.

to provide some desired functionality, but also hides unwanted and shady behavior, and as a result leak valuable information.

5.2.4 Public Information Harvesting

Case (d) in Fig. 5.1 illustrates a new type of threat that we present in this paper, and as such it is previously unknown within both academia and among OSN users. The basis of the threat is that third parties collect user information published in open groups within OSNs like Facebook. Such open groups exist in the boundary between the OSN and the publicly available Internet. Since the information is gathered from open OSN groups there is no need for using covert or deceiving methods when collecting the information. It is simply a matter of scraping the information available on these web pages, which can be done by anyone connected to the Internet. Using the harvested information it is possible for third parties such as profit-driven companies or national security agencies to create social interaction graphs, which details how users interact among a certain topic, e.g. the Occupy Wall street movement. This privacy threat is described further in Section 5.3.
5.2. Privacy Threats

5.2.5 Socialbot

Recently automated software programs, called socialbots, have been seen influencing OSN users [10]. These socialbots are designed to control OSN accounts, by autonomously performing basic tasks such as posting messages and sending friend requests. Socialbots are not applications within the OSN itself, but rather software programs that impersonate the human beings behind user accounts by imitating human behavior towards the OSN, and as such the socialbots fool both the OSN infrastructure itself and the users populating it. Socialbots with these features have been seen infiltrating private and trusted areas shared by Friend relationships in Facebook, and as a consequence harvesting sensitive data from the concerned user accounts.

The threat from socialbots increase since many users are irresponsible when accepting new friend requests from unknown users. In a practical demonstration a socialbot were accepted as friend by OSN users at a rate of 19.3% out of 4493 requested users during the initialization phase and by 59.1% during the socialbot’s propagation phase [11]. Given this high acceptance rate regarding unknown users’ friend requests it is questionable what the effect of privacy settings that limit information access to friends, or friends-of-friends within a OSN really have in practice. If a user’s friend is routinely accepting friend-requests from unknown sources, this friend is a privacy threat, even though this might be unintentional, to both himself and his friends. With respect to our privacy we have therefore come to a situation where we no longer can fully trust the integrity of our friends within OSNs.

5.2.6 Friend-in-the-Middle Trojan Application

This type of threat is indirectly affecting a user when one of the user’s friends add a deceptive Trojan application. The effects on the user and the user’s friends privacy is similar to Trojan application threat described previously. As such, a user’s privacy is dependent not only on his/her own ability and judgement, but also on his/her friends competences, or even weakest friend in this regard.
5. Privacy Threats Related to User Profiling in Online Social Networks

Figure 5.2: (a) and (b) show the interactions done through comments and likes on posts shared on various “Occupy WS” groups. (a) shows interactions before a pepper spray incident at UC Davis, while (b) shows interactions a few days after the incident. Different colors represent different groups; Occupy UC Davis - magenta, Occupy Wallstreet - lilac, Occupy Los Angeles - light blue and Occupy Sacramento - light green.

5.3 Proof-of-Concept

The threat we describe as public information harvesting is based on that users within Facebook can interact in open groups that are publicly available from the Internet. User interaction within these groups is in the form of “Likes”\(^1\) on the group itself, comments within the group, or “Likes” on other users comments. By systematically gathering this public information it is possible to create interaction profiles identifying and profiling users based on the interactions made, i.e., through social interaction graphs as shown in Fig. 5.2.

\(^1\)“Like” is a term found in Facebook where an user can show that they agree or in other way would like to show that they share the same thought as the message, this is called +1 in Google+.
5.3. Proof-of-Concept

5.3.1 Gathering of Information from Open Groups

Facebook provide different methods for third-party application interaction, for instance using the Graph API [12]. The use of this API is straightforward, in a few hours we built an application acting as a data extraction tool that gathered information as an authenticated user on Facebook. Then we created a dummy-user without any interactions or affiliates to begin with. Next our newly created dummy-user accepted our application with just basic permissions. It was then through this dummy-user’s application we gathered data from various open groups on Facebook. However, it is important to stress that the content of open groups are freely available on the Internet so there is no requirement of using a dummy-user to extract this information, we only used it due to convenience reasons.

The information gathered have traditionally been seen by research community as simple post and user information. We have however seen that the information gathered follows such a structured form that different users’ interactions can be combined and form a social interaction graph. Any third-party can gather this user information independent on the user’s privacy settings without their knowledge.

5.3.2 Creating a Social Interaction Graph

From the information gathered in the previous step we created a social interaction graph shown (Fig. 5.2). The figure shows the interactions between different networks before (Fig. 5.2 (a)) and after (Fig. 5.2 (b)) the Pepper-spray incident that happened in Davis, CA. This pepper-spray incident resulted in not only more intense interactions, but also that users involved in their representative community started to interact with other “Occupy” groups.

When looking at the created social interaction graph we can conclude that even if a user have strict privacy settings the user’s actions are hard to hide. We were able to gather not just the name of the users, but also the profile ID making it possible to find out more information about the human behind the user account. The users in todays OSNs must understand that no matter how strict they are trying to protect their user profile with policies, they are still at risk of being profiled based on their behaviors in various groups.
5.3.3 Privacy Implications

Since public information harvesting can be carried out by basically anyone it definitely pose a threat to user privacy. One such example is countries where the regime is interested in targeting and monitoring citizens engaged in various issues that are uncomfortable for the regime. For users living in countries that respect human rights the threat might come from corporations and advertisers to larger extent.

5.4 Protection Mechanisms

In this section we suggest different protection mechanisms against the threats described in section 5.2. This list of protection mechanisms is with no means complete. Using encryption for instance it would be possible to address several of these threats if the OSNs could act and facilitate a public-key infrastructure (PKI), but due to the space-limitation we exclude that in this paper.

5.4.1 Information Leakage from the OSN

Since it is impossible to reach absolute security in any system it is important to inform the OSN users in an adequate manner regarding how their information is handled. Here public discussions to raise user awareness is an important component. It should also be possible to benefit from existing techniques to increase transparency of the OSN policies towards the users, e.g. “Privacy Simplified” [13] that help summarize privacy policies using standardized icons. In addition to improving user awareness it is of paramount importance that the OSN infrastructure is properly secured, and that a continuous security process is established.

5.4.2 Trojan Applications

To improve the protection against Trojan applications, case (c) and (f), we suggest the use of an application certification and reputation program, which just recently has been announced by Facebook under the name “App Center” [14]. We suggest that a more privacy-driven application certification program is added to this initiative, where not only the overall application quality is evaluated, but also the privacy-implications of the data gathered. Combined with a privacy policy of what data the application will retrieve
and how the application will handle this information would make a valuable addition. It is also preferable that interested users should have the possibility to see a audit trail of the interactions the application has carried out on behalf of the user [15].

5.4.3 Questionable Friends

The most important issue to focus on is the lack of user awareness about the problem shown in case (b) and (e), which could be addressed through end-user education. Instructing the users about socialbots, Trojan applications, and the implications of the “Friends-of-Friends” privacy setting. Users should also be instructed to keep their friend-list up to date as far as possible. Also by using various algorithms like the one presented by Fire et al. in [16] the number of questionable friends can be limited.

5.4.4 Social Interaction Profiling

To protect against social interaction profiling, case (d) in Fig.5.1, we suggest the use of pseudonyms or virtual profiles [17]. However, by hiding the real identity of the end-user, for instance using anonymity techniques, will also remove one fundamental value of the OSN, i.e., that the OSN transcends the real world since each user accounts (more or less) corresponds a human being. By using pseudonyms it is possible for a user to interact under separate pseudonyms in different open groups, which renders it impossible to make connections between different groups at least.

5.5 Conclusion

In this paper we present different privacy threats in OSNs. One of these threats is previously undocumented and we therefore describe this threat in more detail together with the results from our own proof-of-concept implementation, which includes the resulting social interaction graph that could be used for user profiling. The proof-of-concept shows that with limited resources it is possible to profile users within an OSN through open groups and then build a social interaction graph of their interactions. Any user within the OSN is vulnerable to this threat, independent on their privacy settings. Finally we suggest a number of different protection mechanisms against the threats identified.
5. Privacy Threats Related to User Profiling in Online Social Networks

5.6 References


5.6. References


SIN: A Platform to Make Interactions in Social Networks Accessible

Roozbeh Nia, Fredrik Erlandsson, Prantik Bhattacharyya, Mohammad Rezaur Rahman, Henric Johnson, S. Felix Wu

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 Facebook’s 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.

6.1 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 a 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 levels 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 create 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 one would 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 a 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
Figure 6.1: 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.

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.

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 6.1 shows the influence of news agencies on social interactions on some of the occupying movement communities on Facebook. Furthermore, figure 6.2 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 6.2, we talk about the related work in the area of social interactions network. We then describe the idea of social interactions network in section 6.3. We give a few example applications for our API in section 6.4. Section 6.5 describes the details of our proposed API. In section 6.6 we discuss the security, privacy, and implementation challenges. We evaluate our API in sections 6.7. Finally, section 6.8 talks about our future plans.
6.2 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].
6.3 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.

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
6.3. Social Interactions Network

Figure 6.3: (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.

Figure 6.3 shows the social interactions network around the contents shared on UC Davis’s Facebook page. Figure 6.3 (a) shows the interactions before the pepper spray incident [17] at UC Davis, while Figure 6.3 (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.

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 6.1 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 6.3 shows the social interactions network around the contents shared on UC Davis’s Facebook page. Figure 6.3 (a) shows the interactions before the pepper spray incident [17] at UC Davis, while Figure 6.3 (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.
6. SIN: A Platform to Make Interactions in Social Networks Accessible

Table 6.1: Shows the different social interactions networks formed around the contents shared on different public pages on Facebook.

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

6.4 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 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 6.4 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.

### 6.5 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.

**Definitions**

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.
Figure 6.4: 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.

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.
Figure 6.5: 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.

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 6.5 shows a complete interactions network around a single post made by a page on Facebook.

$\texttt{facebook} \rightarrow \texttt{api(‘/faith/{Post-ID}/comments’, [$\texttt{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.
6. SIN: A Platform to Make Interactions in Social Networks Accessible

Figure 6.6: 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.

on the given post. The higher the number of comments the stronger the link from the user to the middle of the star, which represents the originator of the post. Figure 6.6 shows the interactions done through comments on the same post as above. Fig 6.6(a) shows everyone who has left a comment on the post. Fig 6.6(b) shows only users that have left more than three comments.

$facebook\rightarrow api('/faith/{Post-ID}/shares', [\$limit])$ - 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.

$facebook\rightarrow api('/faith/{Post-ID}/likes', [\$limit])$ - This call is similar to the previous call, except it only returns the graph created by likes
Figure 6.7: 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.

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 6.7 shows the interactions networked returned by this API call for a public post made on the “Against Stop Online Piracy Act SOPA” page.

$\texttt{facebook}$→api(‘/faith/{Community-ID}’, [{$\texttt{limit}$}]) - This call will return the whole Social Network of all interactions around all the posts in the given community. For 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: 6.8 shows the entire network formed around the community page “Against Stop Online Piracy Act SOPA”.

$\texttt{facebook}$→api(‘/faith/{Community-ID}/comments’, [{$\texttt{limit}$}]) - 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
Figure 6.8: 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.

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.

$facebook\rightarrow\text{api}('fa\text{ith/}\{\text{Community-ID}\}/\text{shares}', [\$limit])$ - This call is similar to \text{api}('fa\text{ith/}\text{Community-ID}'), but here we only consider the sharing of the \text{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.

$facebook\rightarrow\text{api}('fa\text{ith/}\{\text{Community-ID}\}/\text{likes}', [\$limit])$ - This call is similar to \text{api}('fa\text{ith/}\text{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, comment, or a share) and the number of interactions between users. The graph returned is a directed Graph as opposed to Facebook’s 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.

6.6 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 effectiveness on preserving the original graph properties during the anonymization step [20–22]. 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.

6.7 Evaluation

We have taken many steps in order to deal with software failure while generating data from Facebook. There are timeout 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.

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 6.2 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.
6.7. Evaluation

Table 6.2: 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.

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

Table 6.3: Shows how much time we saved during the second phase of our crawler by using a parallel approach with 10 threads.

<table>
<thead>
<tr>
<th>Community</th>
<th>Run Type</th>
<th>Average Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU Community (3041)</td>
<td>Sequential</td>
<td>12'460</td>
</tr>
<tr>
<td>EU Community (3041)</td>
<td>Parallel (10 threads)</td>
<td>1'166</td>
</tr>
<tr>
<td>Milwaukee Bucks (5400)</td>
<td>Sequential</td>
<td>79'897</td>
</tr>
<tr>
<td>Milwaukee Bucks (5400)</td>
<td>Parallel (10 threads)</td>
<td>5'189</td>
</tr>
<tr>
<td>New York Knicks (66020)</td>
<td>Sequential</td>
<td>976'864</td>
</tr>
<tr>
<td>New York Knicks (66020)</td>
<td>Parallel (10 threads)</td>
<td>65'563</td>
</tr>
<tr>
<td>Jay Leno (41152)</td>
<td>Sequential</td>
<td>179'636</td>
</tr>
<tr>
<td>Jay Leno (41152)</td>
<td>Parallel (10 threads)</td>
<td>16'320</td>
</tr>
</tbody>
</table>

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 6.3 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.

6.8 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.

6.9 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.
6.10 References


Making social interactions accessible in online social networks

Fredrik Erlandsson, Roozbeh Nia, Henric Johnson, S. Felix Wu

7.1 Introduction

Recently, online social networks, OSNs, have gained significant popularity and are among the most popular ways to use the Internet. Additionally, researchers have become more interested in using the social interaction networks, SINs[1], in order to further enhance and personalize their services[2]. OSNs are also redefining roles within the publishing industry, allowing publishers and authors to reach and engage with readers directly[3]. However, SINs are not very easily available as of today through the current APIs provided by most OSNs. Such applications would therefore spend tremendous amount of time trying to gather the required SINs for their services. Therefore, our research problem is how we can design a system that makes social interactions in OSNs accessible. This also refers to the problem of how to crawl OSNs in a structured way, which is the focus of this short paper.

The nature of OSNs and the amount of information available makes the problem of what to crawl interesting. To narrow down the scope of the proposed research, we are focusing on the interactions in OSNs. By doing this, we noticed that there exist a gap and segregation between content and social graph. To simply provide social informatics for social computing applications, we have developed a crawler that serves as a bridge between the content and social graph in the online world, by not only providing which users have interacted with each other but around exactly which content these interactions have occurred.

Privacy of the user is a major concern when it comes to all online social interactions and crawling as discussed in [4]. We are treating the crawled
Making social interactions accessible in online social networks

data with high respect to the integrity of the people behind the users.

7.2 Related Work

Despite the huge number of social network publications, few have been dedicated to the data collection process. Chau et al. [5] briefly describe using a parallel crawler running breadth-first search, BFS to crawl eBay profiles quickly. The measurement conducted by Mislove et al. [6] is, to the best of our knowledge the largest OSN crawling study ever published. From four popular OSNs, Flickr, Youtube, LiveJournal, and Orkut, 11.3M users and 328M links are collected. Their analysis confirms known properties of OSNs, such as a power-law degree distribution, a densely connected core, strongly correlated in-degree and out-degree, and small average path length.

Other studies on OSN crawlers include [7, 8]. Gjoka et al. [7] proposed two new unbiased strategies: Metropolis-Hasting random walk (MHRW) and a re-weighted random walk (RWRW). Where Catanese et al. [8] described the detailed implementation of a social network crawler. It used the BFS and uniform sampling as the crawling strategies to run the crawler on Facebook, and then compared the two strategies.

7.3 A Platform to Make Interactions Accessible

We have designed and developed a system that is able to crawl open data. Initially, we will focus on the Facebook Graph API to crawl all content that is viewable to users; such as posts, comments, likes on posts and comments, and shares of posts.

7.3.1 Design

We have designed our crawler to operate in two stages. Stage one uses the Facebook’s unique identifier of a public community (page or a group) to find the id of all posts, messages, photos, and links posted on the given community by admins and members. For readability, a post will refer to anything shared on a community on Facebook in this paper. Stage two is a bit more complicated; for each post gathered in stage one we send at least three to four separate requests (assuming that there are no "likes" on comments), one for the post itself, one for the "likes" on the post (if
there exist any), one to get information on who have shared the post and finally one to get all comments (if there exist any). If one of the responses is paginated we have to make consecutive requests to gather the complete view. This also means that for posts with a lot of interactions we have to make multiple requests to the graph. For instance, we have crawled posts with hundreds or thousands of comments each with a few likes, where we have to make a request for each comment to get its likes. To scope the huge number of requests and the requirement to be efficient, our crawler is built as a distributed service much like discussed in [5]. Figure 7.1 shows a basic sketch how the controller and the crawling agents are connected.

### 7.3.2 Statistics

Over the last eight months our tool have gathered a bit over 150GB of structured data, including: 93 million unique Facebook users, 14 million posts, 126 million comments and over 800 million likes.

### 7.4 Challenges and Requirements

Our crawler highly depends on Facebook’s API, and therefore, bugs in Facebook’s API will cause problems that we have no control over. Also, resource limitations has forced us to be picky about which communities to crawl. Given enough resources, our crawler can be modified to automatically crawl all public communities on Facebook and other OSNs given an initial set of seeds.
7.4.1 Requirements

Our crawler tool, from a high level, is simply a black box that takes the identifier of a Facebook community as input and outputs a stream of documents. In addition to capturing the response of API requests, our crawler has to satisfy the following requirements:

**Real-time.** The information and interactions on Facebook public communities is extremely time-sensitive. In most cases, it is very important to crawl and parse a given post in a community on Facebook online. A few important questions that arise due to the nature of how the interactions around posts evolve are 1) “Which posts do we have to re-crawl to get the most updated information” and 2) “When would be the best time to re-crawl these posts”.

**Coverage.** It is important and desirable to be able to crawl each and every post thoroughly and completely. However, if resources do not allow this, it is more desirable to get all the data from a limited set of posts, rather than less data from a larger set of posts.

**Scale.** As of today there are over a billion users and millions of public communities on Facebook[9]. There are over 2.7 billion likes and comments posted on Facebook on a daily basis as of February 1st[10].

**Data Quality.** The crawler should output good quality and uncorrupted data. Therefore, it needs to be able to detect failures in Facebook’s current API and be able to restart from exactly where it stops when a failure occurs.

7.5 Applications of SINs

There are a vast number of applications where SINs can be used, here we give a brief description of two we have used to evaluate our dataset.

**Dynamic News Feed:** People spend hours on Facebook every day. However, they are only bound to see the posts shared by their immediate friends and pages they have liked. Using social interactions, it is possible to identify the type 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 news feed rather than the current
one where users see the same posts over and over again throughout the day. We can identify which posts the user would be interested in using social interactions but has not interacted with yet. Therefore, the user will only see posts that he/she has not seen before and the content is socially related to what he/she likes.

**Social Search:** Social Search[11] 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 pages 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 can 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 SIN to find related content.

### 7.6 Conclusion

We have shown means of building an extensive tool to gather data from public communities on OSNs. Our distributed crawler satisfies all of our requirements in order to retrieve the complete set of non-corrupted data, including all the content shared and all the user interactions around them. We discuss various applications and how they can benefit from leveraging SINs in order to further personalize their services. Finally, we have given a short description of how to design a data-mining tool for OSNs that can be used to gather data.

### 7.7 References


The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

Teng Wang, Keith C. Wang, Fredrik Erlandsson, S. Felix Wu, Robert Faris

Abstract

With the popularity of social media in recent years, it has been a critical topic for social network designer to understand the factors that influence user continued participation in online newsgroups. Our study examined how feedback with different opinions are associated with participants’ lifetime in online newsgroups. Firstly, we proposed a new method of classifying different opinions among user interaction contents. Generally, we leveraged user behavior information in online newsgroup to estimate their opinions and evaluated our classification results based on linguistic features. In addition, we also implemented this opinion classification method into our SINCERE system as a real-time service. Based on this opinion classification tool, we used survival analysis to examine how others’ feedback with different opinions influence user continued participation. In our experiment, we analyzed more than 88,770 interactions in official Occupy LA Facebook page. Our final result showed that not only the feedback with the same opinions, but also those with different opinions can motivate user continued participation in online newsgroup. Furthermore, an interaction of feedback with both the same and different opinions can boost user continued participation to the greatest extent. This finding forms the basis to understand better how to improve online service in social media.
8. The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

8.1 Introduction

Social media has influenced people’s lifestyle from many aspects. It not only changes the way people collect news and information, but even reforms the way people communicate with each other. One of the most popular uses of social media is to support online newsgroups [1]. They allow people to seek latest news and exchange opinions on a wide variety of topics from entertainment, education to religions and politics. Despite the popularity of online newsgroups, it is very difficult to maintain them for a long time. Member participation in online newsgroups is often sparse and uneven [2]. In this paper, we examine the factors that influence user continued participation in online newsgroups. To be specific, our work focuses on the effect of feedback with different opinions.

In 2008, S.L. Johnson [3] gave a detailed definition of online groups from the views of group membership and interaction. Besides the criteria mentioned by S.L. Johnson, online newsgroups, as a special case of online groups also has its own characteristics: members in online newsgroups are less of social component in real life and most of them are strangers with few off-line communication. It is just these characteristics that make online newsgroups ideal resources for researchers to examine user influence. Firstly, because most of the members in newsgroups are strangers in real life, they would be more open to share their opinions online while people in private friendship group may have more concerns. Additionally, in online newsgroup, because most of people’s interaction happens in online environment, the offline influence will have very little effect on the analysis result. In all, user interaction data in online newsgroup can give us more comprehensive information of their mutual influence pattern.

In general, there are mainly two challenges to examine the effect of feedback with different opinions on user continued participation. Firstly of all, we need to find out an effective and efficient method to classify user comments into different opinions. In this paper, instead of only focusing on corpus itself, we leverage user behavior information and build user-like graph to do opinion classification. Besides that, we also use linguistic analysis tool to evaluate the classification results and develop this method into a real-time service on our SINCERE system1. The second challenge of this topic is to recognize the influence of different feedback among those factors on user

1http://sincere.se/
continued participation. In this paper, we perform a large scale study of user interaction on Official Occupy LA Facebook page with totally 20,569 unique users, 56,937 comments, 31,833 posts and 66,758 likes. Based on this dataset, we use Cox proportional hazards model [4] in survival analysis to explore the relationship between user lifetime in online newsgroup and several explanatory variables. Our final result showed that the content of feedback in online newsgroup are significantly related to user continued participation: Not only the feedback with the same opinions, but also those with different opinions can motivate user continued participation. Furthermore, an interaction of feedback with both the same and different opinions can boost user continued participation to the greatest extent.

The rest of the paper is organized as follows. In section 8.2, we discuss related work on factors in user continued participation. Then we describe our opinion classification method and result evaluation in section 8.3. Based on this method, we design an experiment to analyze the influence of different feedback on user continued participation in section 8.4. Discussion and conclusions are talked about in section 8.5 and we also mention the limitations and future work in section 8.6.

8.2 Related Work

Moira et al. [5] grouped the theories of user online participation into three high level categories: social learning, distribution and feedback. In this paper, we focus our work on examining the influence of feedback on user continued participation. In previous work, theories of reciprocity [6] and reinforcement both proved that feedback from other users should predict long-term participation. As a careful analysis, the influence of feedback can come from the following three parts: the role of people who give the feedback, the amount of feedback and the content of feedback.

As to the role of people who give feedback, Steven Johnson [7] showed that interaction with online group leadership is associated with higher participation continuance and participation intensity. During the interaction with leadership in online group, the participant can feel psychological safety. Steven indicated that members of online groups with higher levels of psychological safety report higher levels of continued participation intentions.

Furthermore, people also found positive relationship between the amount
of received feedback and user continued participation. Previous work [2, 5] on newcomers in online newsgroups showed that newcomers will be more likely to post again if anyone responds to their initial post. Besides the work on newcomers in online groups, Y. Wang et al. [8] examined the factors that influence the continued participation of any member in online health support groups. Their results showed that the count of all comments in the threads in a week in which the user had posted is highly correlated ($r = .67$) with user’s continued participation in online health support group.

When it comes to the content of feedback, there are many different ideas. Some previous work focused on newcomers in online newsgroups [2, 9]. E. Joyce [2] indicated that length, tone, content and personal affirmation are not found to be significant predictors of long-term engagement for newcomers in online newsgroup. However, Y. Wang et al. [8] examined the influence of feedback on all the group members and found significant relationship between the content of feedback comments and user commitment in online health support groups. In this paper, our work also considers all the group members. But different from Y. Wang’s work which analyzed different types of social supports among user interaction contents, we focus on different feedback opinions in online newsgroups.

The influence of feedback on different opinions is widely discussed in democratic deliberation research on political communication area [10, 11] and this issue is still controversial now. Generally, there are two different perspectives in this area. Some people believed that expressions of disagreement may violate expected norms of politeness in social interactions [12]. Mutz [13] proposed that the negative effects of disagreement may make people avoid political discussion and deliberation. However, J. Stromer Galley et al. [10] indicated that expressions of disagreement do not generally harm participants’ future participation and an interaction of agreement and disagreement can even boost expected future participation in democratic deliberations. However, their work only focused on people discussion content on political topics and most of them used questionnaire or phone survey as their data collection method, which severely limits their experiment sample. This paper, to the best knowledge of us, is the first work to give a large scale analysis of the influence of feedback with different opinions on user continued participation in online newsgroup. We offer a new opinion classification algorithm to distinguish feedback comments into different opinions automatically, which allows us to do analysis on a much
larger dataset. Additionally, instead of only focusing on political topics where people’s stance is always sharp and in opposition to each other, our experiment broadens user interaction data to general discussion topics in online newsgroups.

8.3 Opinion Classification

In Natural Language Processing area, there are many work discussing opinion classification in online threaded discussion [14, 15]. In the latest research work, Rob Abbott et al. [16] identified disagreement in political blogs using only lexical features. However, their machine learning method need a complex training process and every training model is only valuable for its source corpus. Instead of only focusing on the corpus itself, we can also leverage user behavior information in online newsgroups to assist us on opinion classification. Some recent work showed that user behavior features can be used to capture contextual information present in textual features very accurately [17]. Taking the public newsgroup on Facebook website as an example, besides the text of user interaction under each post, the information of like behavior is also a valuable tool for us to recognize user’s opinion. In this paper, we leverage the user-like-graph under each post to classify user interaction content into different opinion based on its link structure.

8.3.1 Data Collection

Our dataset is crawled from public pages on Facebook. Those pages are open to the public and anyone with a Facebook account can post or comment to existing posts. To be specific, each public page is a tree-shaped structure in which multiple posts are organized in temporal ordering. Facebook users can share latest news and discuss their opinions on different topics in those pages, which makes them ideal examples of online newsgroups. Under each post, we can get all of the user interaction information, such as the content of comments, likes information and their time-stamps. In general, our method leverages user like information on each comment to do opinion classification, which can be applied to analyze user interaction information on any Facebook public page in real time.
8.3.2 User-like graph

As to each post in a Facebook public page, we define a user-like graph $G = (V, E)$, where $V$ is a set of nodes and $E$ is a set of edges among $V$. For simplicity, in this paper, we consider the user-like graph undirected graph. A node stands for a user who has liked a comment or whose comments were liked by others in this post. An edge $e$ stands for the connection between two users $u$ and $v$ in this user-like graph and its weight $w_{uv}$ equals to the number of likes they have clicked on each other’s comments in this post. Figure 8.1 shows two examples of user-like graph of the posts in official Occupy LA Facebook group.

8.3.3 Opinion Classification method

The general idea of our opinion classification method is as follows: As to each post, we firstly classify people into different groups based on its user-like graph. Then we collect the comments made by different groups of people as different opinions contents. Therefore, as the first step, we need to find a partition of the user-like graph such that edges between different groups have a very low weight (which means people in different clusters are holding different opinions from each other) and the edges within a group have high
weight (which means that people within the same cluster are holding similar opinions with each other). Obviously, this is a classic graph partitioning problem.

In order to make our output clusters reasonably large groups of nodes, we use the concept of Ratiocut [18, 19] to formulate our objective function. In user-like graph $G = (V, E)$, we denote a subset of vertices $A \subset V$ and $\overline{A}$ for the complement of $A$. In addition, for two set $A, B \subset V$, we defines

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

(8.1)

For a given number $k$ of subsets, we choose a partition $A_1, \ldots, A_k$ which minimizes

$$\text{Partition}(A_1, A_2, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|}$$

(8.2)

In the objective function 8.2, the size of a subset $A$ of a user-like graph is measured by its number of vertices, i.e. the amount of people in this subset. It will get a small value if the clusters $A_i$ are not too small. Therefore, this objective function can make our output clusters balanced, measured by both the connections between each cluster and the number of their vertices. Unfortunately, introducing balancing conditions makes our partition problem become NP hard. In order to solve this objective function, we choose to use spectral clustering algorithm [19], which is the most popular algorithm to solve relaxed versions of Ratiocut problem.

Based on our discussion above, we can now get a near-optimal partition of any user-like graph by spectral clustering algorithm when the number of cluster $k$ is fixed. Now we need to decide the optimal number of cluster $k$ for a user-like graph. In this paper, for simplicity, we only choose the cluster number from 1 to 2, i.e. we only want to decide if the comments under the same post can be clustered as two different opinions or people are just holding the same opinion in this post. In 2004, Newman and Girvan [20] proposed a modularity function which can directly measure the quality of a particular clustering of nodes in a graph. Their function $Q$ measures the fraction of the edges in the graph that connect nodes in the same group minus the expected value of the same quantity in a graph with the same partitioning result but random connections between the nodes. If the sum weight of edges in the same groups is the same with that got by random
connection, we will get $Q = 0$. And if the partitioning result has a strong community structure, the value of $Q$ will be very close to 1 [20].

In our method, after getting the partitioning result by spectral clustering algorithm with $k = 2$, we will use modularity $Q$ to check the clustering quality of our partitioning result and decide the number of clusters in this user-like graph from 1 to 2. Newman’s work showed that real-world unweighted networks with high community structure generally have $Q$ values within a range from 0.3 to 0.7. Figure 8.2 shows four examples of our clustering results and their corresponding $Q$ values. Each of these graphs represents the user-like structure of one of the posts in Occupy LA Facebook group. The two different colors (pink and blue) stands for the spectral clustering result when we fix $k = 2$. We can find that $Q$ value can be a good measurement for deciding the number of clusters in the user-like graph: The clustering results with $Q \geq 0.2$ show strong community structure, where $k$ remains to be 2, while results with $Q < 0.2$ converge at some special nodes, where $k$ is determined as 1. In this paper, we set $Q = 0.2$ as an threshold. Therefore, if the value of modularity measurement $Q$ is less than 0.2 when $k = 2$, we will consider the user-like graph as one cluster, or we accept its partitioning result and cluster the nodes into two different groups. As the last step, for each post, we collect the comments made by different clusters of people as different opinions contents.

### 8.3.4 Evaluation from Linguistic Features

In order to evaluate the effectiveness of our opinion classification method, we select three Facebook public pages as our dataset: OccupyLA\(^2\), Occupy Wall Street\(^3\) and Occupy Together\(^4\). Starting from Sept. 2011, the Occupy movement call for people to protest against social and economic inequality, which attracts a wide range of people all round the world. Online social networks, during this time, plays an important role by offering people an ideal platform to get latest news and share opinions. Among all the public newsgroups about occupy movement on Facebook website, the public pages of Occupy Wall Street and Occupy LA are the largest two groups, and OccupyTogether is a comprehensive public page where people share information and opinions on any occupy movement. Additionally, in order

---

\(^2\)http://www.facebook.com/occupyLA  
\(^3\)http://www.facebook.com/OccupyWallSt  
\(^4\)http://www.facebook.com/OccupyTogether
8.3. Opinion Classification

Figure 8.2: Examples of Clustering Results with Different Q Values

(a) $Q = 0.03. k = 1$

(b) $Q = 0.06. k = 1$

(c) $Q = 0.27. k = 2$

(d) $Q = 0.31. k = 2$

Figure 8.2: Examples of Clustering Results with Different Q Values

to examine user interaction content in each post, the number of comments in
the post that we analyze should reach a certain amount. In this experiment,
we select all the posts which contain more than 35 comments inside. Finally,
we get 205,198 comments among 1929 posts in these three public pages from
Sept. 2011 to Apr. 2012. Because our dataset is very large, it is impossible
for us to get the ground truth of the opinion classification result for each
post by employing workers to rate manually. In this paper, we evaluate our
opinion classification method from the linguistic features of user comments,
which is a totally different view from that with the structure of user-like
graph.

We use the Linguistic Inquiry and Word Count (LIWC) tool [21] to
study the linguistic characteristic of user comments. LIWC is a popular tool which calculates the frequency with which words in a text match each of 68 categories representing linguistic dimensions, psychological constructs and personal concerns \[8\]. Many previous work \[8, 22\] have shown that the categories in LIWC are effective in determining linguistic differences on attentional focus and emotionality of the relationship. In this paper, we consider 39 categories in 13 areas, which is shown in Table 8.1. Among these 39 categories, 32 of them belong to psychological processes and 7 of them belong to personal concerns area. The analysis results in \[8\] and \[16\] reveal that the LIWC scores on psychological process are very helpful on identifying people’s emotional attitudes, i.e. agreement and disagreement opinions, in their online comments. Besides that, \[17\] also showed the LIWC score of categories on personal concern is also effective on identifying people’s different references in online groups.

During the evaluation process, our theoretical basis is: Comments on different opinions have different characteristics on their linguistic features \[16\]. After using our opinion classification method to analyze those 1929 posts, 1216 posts are considered with two different opinions inside. As to each of these 1216 posts, we denote \(G\) as the set of comments of one post and \(P, Q \subset G\) as the two groups of comments on different opinions. Suppose there are \(m\) comments in set \(P\) and \(n\) comments in set \(Q\). The LIWC analysis result of one comment is denoted as \(S_{ij}\), where \(i\) denotes the ID of the comment in set \(P\) or \(Q\) and \(j\) denotes the serial number of the 39 categories. In set \(P\), we define the LIWC scores in the \(i\)th category as \(X_{pj} = [S_{1j}, S_{2j}, ..., S_{nj}]^T\) (\(j \in [1, 39]\)). Similarly, in set \(Q\), the LIWC scores in the \(i\)th category is defined as \(X_{qj} = [S_{1j}, S_{2j}, ..., S_{mj}]^T\) (\(j \in [1, 39]\)). Therefore, the LIWC result of these two groups of comments can be denoted as follows:

\[
L(P) = [X_{p1}, X_{p2}, ..., X_{p39}] \quad (8.3)
\]

\[
L(Q) = [X_{q1}, X_{q2}, ..., X_{q39}] \quad (8.4)
\]

Then as to each of the 39 LIWC categories, we use Welch’s t-test \[23\] to test whether the means of the two population \(X_{pj}\) and \(X_{qj}\) are different from each other. For LIWC category \(j\), we denote the p-value of their t test as \(p_j\) (\(j \in [1, 39]\)). So we get the result of linguistic comparison as

\[
\text{Compare}(P, Q) = [p_1, p_2, ..., p_{39}] \quad (8.5)
\]
Table 8.1: The 39 textual Categories in 13 areas used in our linguistic evaluation. Areas marked with \(^1\) are psychological processes, and areas marked with \(^2\) are personal concerns.

<table>
<thead>
<tr>
<th>Areas</th>
<th>Categories</th>
<th>Selected Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social(^1)</td>
<td>mate, talk, they, child</td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td>husband, aunt</td>
<td></td>
</tr>
<tr>
<td>Friends</td>
<td>friend, neighbor</td>
<td></td>
</tr>
<tr>
<td>Humans</td>
<td>adult, baby, boy</td>
<td></td>
</tr>
<tr>
<td>Affective(^1)</td>
<td>happy, cried, abandon</td>
<td></td>
</tr>
<tr>
<td>Pos. Emotion</td>
<td>love, nice, sweet</td>
<td></td>
</tr>
<tr>
<td>Neg. Emotion</td>
<td>hurt, ugly, nasty</td>
<td></td>
</tr>
<tr>
<td>Anxiety</td>
<td>fearful, nervous</td>
<td></td>
</tr>
<tr>
<td>Anger</td>
<td>hate, kill, annoyed</td>
<td></td>
</tr>
<tr>
<td>Sadness</td>
<td>crying, grief, sad</td>
<td></td>
</tr>
<tr>
<td>Cognitive(^1)</td>
<td>cause, know, ought</td>
<td></td>
</tr>
<tr>
<td>Insight</td>
<td>think, know, consider</td>
<td></td>
</tr>
<tr>
<td>Causation</td>
<td>because, effect, hence</td>
<td></td>
</tr>
<tr>
<td>Discrepancy</td>
<td>should, would, could</td>
<td></td>
</tr>
<tr>
<td>Tentative</td>
<td>maybe, perhaps, guess</td>
<td></td>
</tr>
<tr>
<td>Certainty</td>
<td>always, never</td>
<td></td>
</tr>
<tr>
<td>Inhibition</td>
<td>block, constrain, stop</td>
<td></td>
</tr>
<tr>
<td>Inclusive</td>
<td>and, with, include</td>
<td></td>
</tr>
<tr>
<td>Exclusive</td>
<td>but, without, exclude</td>
<td></td>
</tr>
<tr>
<td>Perceptual(^1)</td>
<td>heard, feeling</td>
<td></td>
</tr>
<tr>
<td>See</td>
<td>View, saw, seen</td>
<td></td>
</tr>
<tr>
<td>Hear</td>
<td>Listen, hearing</td>
<td></td>
</tr>
<tr>
<td>Feel</td>
<td>Feels, touch</td>
<td></td>
</tr>
<tr>
<td>Biological(^1)</td>
<td>eat, blood, pain</td>
<td></td>
</tr>
<tr>
<td>Body</td>
<td>cheek, hands, spit</td>
<td></td>
</tr>
<tr>
<td>Health</td>
<td>clinic, flu, pill</td>
<td></td>
</tr>
<tr>
<td>Sexual</td>
<td>horny, love, incest</td>
<td></td>
</tr>
<tr>
<td>Ingestion</td>
<td>dish, eat, pizza</td>
<td></td>
</tr>
<tr>
<td>Relativity(^1)</td>
<td>area, bend, exit, stop</td>
<td></td>
</tr>
<tr>
<td>Motion</td>
<td>arrive, car, go</td>
<td></td>
</tr>
<tr>
<td>Space</td>
<td>down, in, thin</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>end, until, season</td>
<td></td>
</tr>
<tr>
<td>Work(^2)</td>
<td>job, majors, xerox</td>
<td></td>
</tr>
<tr>
<td>Achievement(^2)</td>
<td>earn, hero, win</td>
<td></td>
</tr>
<tr>
<td>Leisure(^2)</td>
<td>cook, chat, movie</td>
<td></td>
</tr>
<tr>
<td>Home(^2)</td>
<td>kitchen, family</td>
<td></td>
</tr>
<tr>
<td>Money(^2)</td>
<td>audit, cash, owe</td>
<td></td>
</tr>
<tr>
<td>Religion(^2)</td>
<td>altar, church, mosque</td>
<td></td>
</tr>
<tr>
<td>Death(^2)</td>
<td>bury, coffin, kill</td>
<td></td>
</tr>
</tbody>
</table>
Because the posts in our dataset cover diversified topics, we cannot limit their different linguistic features show in only one particular item of those 39 categories. Therefore, as the final step, if any of these 39 p-values is less than the predetermined significance level $\alpha(= 0.05)$, we will conclude the two groups of comments got by our opinion classification method reveal different characteristics on their linguistic features, which indicates that our opinion classification result is acceptable for this post. Table 8.2 shows our evaluation results of all the 1216 posts in the three Facebook public pages. The item Total denotes the amount of posts with more than 35 comments in each online newsgroup. The item Graph denotes the amount of posts which are recognized as two opinion groups inside by our opinion classification method. The item Linguistic denotes the amount of posts within Graph which reveal different linguistic features. From the result we can see that 959 of those 1216 posts reveal different linguistic features, which achieves an accuracy of 78.9%.

### 8.3.5 System Development

Besides theoretical evaluation, we also implemented this opinion classification method into our SINCERE system (Social Interactive Networking and Conversation Entropy Ranking Engine) as a real-time service. SINCERE system is a diversified search engine based on user social informatics. Its database offers all the interactions (such as likes, shares, comments, timestamps) of 1391 Facebook public pages.

Figure 8.3 shows a screen shot of our opinion classification service on SINCERE. This system can automatically classify the comments of this post into different opinions (one or two opinions) and show the result in a pull-down menu. If there are only one opinion group recognized, the system will list all of the comments in one list. Otherwise, it will show the different groups of comments on two parallel columns, which is the same style with the example in Figure 8.3. From the content of those classified comments in our example, we can find that the opinion classification result is very
8.4 The Influence of Feedback with Different Opinions on User Continued Participation

In section 8.3, we introduced a new opinion classification method in online newsgroup and evaluated its effectiveness from linguistic features. In this section, we use this method as a tool to analyze the influence of feedback with different opinions on user continued participation in online newsgroup. The dataset we use in this experiment is crawled from the official Occupy LA Facebook public page. We collected all the posts, comments and like information on this page from Sep. 2011 to Apr. 2012. During this period of time, there were in total of 20,569 users who posted 56,937 comments belonging to 31,833 posts. Additionally, there are also 66,758 likes among all...

Figure 8.3: Screen-Shot of Our Opinion Classification Service on SINCERE System effective: Under the post where people are discussing the recent actions of Los Angles Police, the comments on the left (marked as green color) express support and kindness, while most of the comments on the right (marked as pink color) show skeptical attitude.
of these comments. In order to analyze the influence of feedback on online newsgroup participants, the user we examine should have enough amount of activity record in this group. Therefore, in this experiment, we are only interested in those users who have made more than 20 comments in Occupy LA public page, which includes 622 users in total.

We use Cox proportional hazard (PH) model [24] in survival analysis [4] to explore the relationship between user lifetime in this newsgroup and several explanatory variables on the influence of feedback comments. Survival analysis is a main method to examine and model the time it takes for some special events to occur [24]. In our experiment, the specified event is defined as the end of the user’s active lifetime on this page. In previous work, this technique has been widely used in medical science, sociology and engineering [9]. As the most widely used method of survival analysis, Cox regression can provide estimated coefficients for each covariate and allow the assessment of the impact of multiple covariates in the same model.

8.4.1 Experiment Design

8.4.1.1 Dependent Variable

- **Lifetime**: Because we are interested people’s active lifetime on online newsgroups, user’s first and last comment time may not be ideal indicators for the actual user lifetime in online newsgroup. [25] gave a definition on the lifetime of IRC channels based on their level of activity. In our experiment, we include user comment frequency into the definition of user lifetime. In our dataset, the total time duration is 220 days. Firstly, we divide this period of time into 22 time-blocks with the same 10-day interval. Then as to each participant in this newsgroup, we considered him/her to start his/her activity when he/she makes more than 3 comments in two consecutive time-blocks. And we consider him/her to left this newsgroup at the day after when he/she gives no comments in its following two consecutive time-blocks. Figure 8.4 shows the distribution of user lifetime of those selected 622 participants in Occupy Los Angeles public page. Their online lifetime ranges from 202 days to 0 days. In addition, because people whose last comment is found within the last time-block may still be participating in this Occupy LA group, we treat them as right censored in the survival analysis.
8.4. The Influence of Feedback with Different Opinions on User Continued Participation

8.4.1.2 Control Variables

- **OriginalPostWriter**: Among all the 31,833 posts in Occupy Los Angeles public page, most of them are written by the official maintainers of this public homepage named *OccupyLA*. However, there are also many posts written by normal users. Considering people who write original posts on this page may have different participating enthusiasm than people who just take part in discussions started by others, for each of those 622 members we are interested in, we define a control variable `OriginalPostWriter` as the percentage of his/her original posts among all of his comments in Occupy Los Angeles public page.

- **NonDiscussionPostsInvolved (NonDiscussion)**: Among all the posts written by the official maintainer of Occupy Los Angeles group, we also split them into two parts: one is the posts with more than 35 comments inside and another is those with less than 35 comments. The posts in the first part are considered as discussion posts. As to the posts in the second part, we use them to define a control variable `NonDiscussionPostsInvolve`. It is the number of an individual’s comments in non-discussion posts divided by the amount of his/her comments in all posts.

- **ReceivedCommentsPerActivity (ComPerAct)**: In each discussion post in Occupy LA group, there is not exactly reply during user interaction: people just make comments one after another along the timeline of each post. So we define the following comments within
The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups

Table 8.3: Descriptive Statistics for Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>Std.Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OriginalPostWriter</td>
<td>0</td>
<td>1</td>
<td>0.320</td>
<td>0.318</td>
</tr>
<tr>
<td>NonDiscussion</td>
<td>0</td>
<td>1</td>
<td>0.415</td>
<td>0.237</td>
</tr>
<tr>
<td>ComPerAct</td>
<td>0</td>
<td>57.625</td>
<td>5.679</td>
<td>6.900</td>
</tr>
<tr>
<td>SamePercent</td>
<td>0</td>
<td>1</td>
<td>0.375</td>
<td>0.283</td>
</tr>
<tr>
<td>DifferentPercent</td>
<td>0</td>
<td>0.8</td>
<td>0.110</td>
<td>0.133</td>
</tr>
</tbody>
</table>

three hours after the individual makes a comment as his/her received replies. We assume this people read all the following comments within three hours after his/her comment and regard them as feedback for his/her activity. Based on this definition, we calculated the average number of received replies during the three hours after the individual makes a comment.

8.4.1.3 Independent Variables

- **SameOpinionsPercentage(SamePercent):** This variable measures the percentage of the replies in the same opinion with this person among all of the replies this person received. To be specific, during the three hours after this person makes a comment, all the received comments can be classified into three groups: comments with the same opinion with him/her in this post (these comments are made by people who are in the same opinion group in this post); comment with different opinion with him/her in this post (these comments are made by people who are in different opinion group in this post); comments with unclear opinion (these comment are made by people who neither clicked like on other’s comments nor are liked by others).

- **DifferentOpinionsPercentage(DifferentPercent):** This variable measures the percentage of the replies in different opinion with this person among all of the replies this person received.

Table 8.3 shows the descriptive statistics of all these variables. Based on the definitions above, we standardize all the control and independent variables with a mean of zero and standard deviation of one and use Cox regression model to analyze the relationship between the user lifetime in Occupy LA group and these variables.
8.4. The Influence of Feedback with Different Opinions on User Continued Participation

8.4.2 Experiment Result

Results of Cox regression model are shown in Table 8.4. The exponential coefficient indicates the direction of the effect of variables: when exp(coef) is smaller than 1, it represents a positive relationship between the variable and the lifetime. For example, because the exp(coef) of OriginalPostWriter is 0.641 which is less than 1, we can say that when all other variables are in average values, the more original posts one user writes, the more lifetime he/she will have in this newsgroup. Std. Err. indicates its standard error.

Table 8.4: Results of Cox Regression Model

<table>
<thead>
<tr>
<th>Control/Independent Variable</th>
<th>exp(coef)</th>
<th>Std. Err.</th>
</tr>
</thead>
<tbody>
<tr>
<td>OriginalPostWriter</td>
<td>0.652***</td>
<td>0.123</td>
</tr>
<tr>
<td>NonDiscussionPostsInvolved</td>
<td>0.723**</td>
<td>0.108</td>
</tr>
<tr>
<td>ComPerAct</td>
<td>0.770*</td>
<td>0.120</td>
</tr>
<tr>
<td>SamePercent</td>
<td>0.813**</td>
<td>0.074</td>
</tr>
<tr>
<td>DifferentPercent</td>
<td>0.875**</td>
<td>0.055</td>
</tr>
<tr>
<td>SamePercent X DifferentPercent</td>
<td>0.761**</td>
<td>0.087</td>
</tr>
<tr>
<td>ComPerAct X SamePercent</td>
<td>1.324***</td>
<td>0.071</td>
</tr>
<tr>
<td>ComPerAct X DifferentPercent</td>
<td>1.140*</td>
<td>0.062</td>
</tr>
</tbody>
</table>

*: p<0.05, **: p<0.01, ***: p<0.001

Among the results of two control variables, the exp(coef) value (0.652) of OriginalPostWriter tells us that when all other variables are in average values, members who posted an average of one standard deviation (0.320) more original posts were 35% more likely to remain in the group. Similarly, the exp(coef) value of NonDiscussionPostsInvolved indicates that members who involves one standard deviation more in Non-Discussion Posts were 28% more likely to remain in the group. And those who received a standard deviation more feedback after his/her comment revealed to be 23% more likely to remain in the group.

In addition, both of the independent variables, SamePercent and DifferentPercent show significant influence on user survival rate in social newsgroup. Members who received a standard deviation more feedback on the same opinion with him/her were 19% more likely to remain in the group and those who received a standard deviation more feedback on different opinions also showed improved preference to remain in the group, with the rate of 13%. In other words, both positive and negative feedbacks from others can motivate user longtime participation in social newsgroups, and positive feedbacks has a slightly more driving effect on it.

\[ 35\% = ((1 - 0.652) \times 100\%) \]
Furthermore, we also consider the interaction between independent variables and those with the control variable ComPerAct. Firstly, when controlling all of the control variables, the two independent variables SamePercent and DifferentPercent revealed super-linear positive influence on the user lifetime: not only each of them reveals positive influence on the lifetime, their interaction (SamePercent * DifferentPercent) shows significant positive relationship with user longtime participation as well. In other words, compared with people received an average number of positive feedback and negative feedback, members who received a standard deviation more feedback with both positive and negative feedback revealed to be 46% more likely to remain in the group, which is much more higher than the result of their linear combination 29%. What is more, we also find that the interaction between ComPerAct and SamePercent/DifferentPercent shows negative influence on the user lifetime. Taking the interaction ComPerAct * DifferentPercent as an example, it indicates that the common influence of ComPerAct and DifferentPercent 23% is not as great as that from their linear combination 33%. This may be the result of duplication influence of them on user lifetime.

8.5 Discussion and Conclusions

In this paper, we built user-like graph to classify different opinions of user interaction content in online newsgroup. And then we use Cox regression model to evaluate the influence of feedback with different opinions on user lifetime in official Occupy LA Facebook group. From the results shown above on different variables, we can get many important conclusions which can help designers of social network system to get a deeper understanding of user behavior and improve their online service.

Among the three control variables, firstly, the results show that members who started more original posts revealed a longer lifetime in this newsgroup. Therefore, the designers of Online social network can motivate users longtime participation by offering more opportunities for normal users to post their own news and become a discussion starter. Secondly, we find those who have

\[6\text{46\%} = (1-0.813*0.875*0.761)*100\%\]
\[7\text{29\%} = (1-0.813*0.875)*100\%\]
\[8\text{23\%} = (1-0.77*0.875*1.140)*100\%\]
\[9\text{33\%} = (1-0.77*0.875)*100\%\]
more comments in Non-discussion posts preferred to stay in this newsgroup longer. Our explanation for this result is that although Non-discussion posts do not have enough comments to host a discussion environment, the information it offered is also very important. D. Fisher et al. [26] indicates that the topic of forum is one of the factors which we can use to predict user engagement. Therefore, this result tells website designers that despite the importance of discussion posts with many comments and people involved, they should not neglect the information offered by Non-discussion posts. Last but not the least, the result of control variable ComPerAct shows that the more feedback one individual receives after his/her comment, the more likely he/she will remain in this newsgroup.

When controlling all the control variables, our final result indicates that not only the number of replies, but also their content has a significant influence on user’s commitment to online newsgroup. To be specific, our conclusion is that not only the feedback with the same opinions, but also those with different opinions can motivate user continued participation. Furthermore, an interaction of feedback with both the same and different opinions can boost user continued participation to the greatest extent. Based on our conclusion, we think that although feedback with different opinions may result in an unpleasant interaction, they help to form a comprehensive and healthy discussion environment. In psychological area, De Dreu et. al. [11] indicates that conflicts in group discussion can increase creativity and divergent thinking. Therefore, we believe that when people are involved in a discussion with different perspectives, their understanding of certain topics may be improved, which will increase people’s evaluation on this online newsgroup and motivate their future participation. This result tells website designers that they can also motivate user continued participation by digging into user interaction contents. For example, website designers can highlight or send notices to a user when feedback from different opinions show up in his/her involved discussion posts.

8.6 Limitations and Future Work

One of the limitations in this paper comes from our opinion classification method. Firstly, we modeled the user-like graph as an undirected graph which makes no discrimination between the writer of one comment and
the likers of that comments. However, the degree of their preference on certain opinions may different from each other. Future work can model it as a directed graph so as to get a more accurate model on user opinions. Secondly, in this paper, we only choose the number of opinion groups in one post between 1 to 2 and classify the person in user-like graph into either one of the opinion groups. However, user interaction in online newsgroup is a much more complex scenario: Different from those content in debate forum [16] where people’s opinions form apparently two parties, the discussion in online newsgroup may consist of many different opinions, each of which starts from a different view and is not necessarily opposed to others. Therefore, another future work can consider more than two opinion-groups among user interaction content and assign them various degrees as feedback comments instead of simply agreement and disagreement.

Furthermore, when we analyze the influence of feedback on user continued participation, we average the effect of the feedback comments during the whole user’s lifetime in online newsgroup. However, the influence of feedback may change during user participation in one group and different factors of feedback, such as the role of speaker, the amount of feedback and the content of feedback, may have various dynamic influence pattern during user participation. Future work can analyze how those different factors are changing their influence during user lifetime in online newsgroup.

8.7 Acknowledgment

We want to thank Mohammad Rezaur Rahman and Roozbeh Nia for their help on data collection. We are also very grateful to Ran Cao in Statistics Department of Michigan State University for her insightful comments on the survival analysis part. 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.

8.8 References

8.8. References


8. **The Influence of Feedback with Different Opinions on User Continued Participation in Online Newsgroups**


[23] B. L. Welch. “The generalization of student’s problem when several different population variances are involved”. In: *Biometrika* 34.1/2 (1947), pp. 28–35.


Interaction metrics to support crawling prioritization in online social networks

Fredrik Erlandsson, Martin Boldt, Henric Johnson, S. Felix Wu

Abstract

The size and current growth of online social networks (OSNs) are creating ‘big data’ challenges when it comes to data gathering and analysis. In this work we present a novel agent-based crawler that have gathered 263 million unique Facebook users interacting through 308 million comments and 3.5 billion likes. Further, we investigate the growth of Facebook communities and how to efficiently make use of a limited set of resources while crawling a continuously growing dataset. Gathering all content in real-time is simply not possible due to the vast amount of data in combination with restrictions in Facebook’s API. To address this limitation we have investigated methods to estimate the interestingness of content within OSNs. Using the interestingness metric we then filter and prioritize what content to crawl and in which order. We verify our method by crawling 176 randomly sampled Facebook pages, containing 1.3 million posts and 135 million interactions. The results show that it is possible to reduce roughly 70% of the content and still capture 99.9% of the interactions, and at the same time reduce the crawling time by 35%.

9.1 Introduction

Online social networks (OSNs) and social media, such as Facebook, Twitter, Instagram and Google+, are attracting increasingly more interest from Internet users. Facebook alone increased its number of users with 22% between 2012 and 2013, and in January 2014 there were 1.31 billion active users [1]. OSNs influence peoples daily routines and everyday behaviors,
9. Interaction metrics to support crawling prioritization in online social networks

for instance 48% of US adults between 18 and 34 check Facebook the first thing when they wake up [1]. Apart from changing the way people interact and communicate, OSNs also provides novel means of news aggregation. Today it is possible to maintain social awareness by following certain people and newsgroups within OSNs. The growing use of OSNs poses interesting ‘big data’ challenges regarding storage, management and analysis of users’ online activities. In January 2014, Facebook stated that they are storing data of the magnitude of exabytes (10^{18} bytes), and this number is steadily growing with roughly 9 million messages sent every hour. Except the obvious challenges of storage and processing all this data, two aspects are addressed in Section 9.5. Namely methods to predict interestingness and also how the data can be used in the most efficient way with aspect of user interactions. We are addressing means to find data with high use for various application areas and research problems. This article addresses the challenge of how to efficiently make use of a limited set of resources while crawling a continuously growing dataset of public data from Facebook.

One interesting aspect with data from OSNs is the fact that it is humans that produce the data, in contrast to synthetic data. Using this data enables research areas that were hard to realize just a few years ago, e.g., big-scale user interaction analysis [2, 3] and the creation of Social Interaction Networks (SIN) graphs [4]. A SIN graph shows the interactions between users in various communities and can for instance represent interactions of all users on one newsgroup or relating to a specific topic. This allow, for instance, researchers to enable novel applications related to social sciences.

In addition we have developed and evaluated a crawler that uses a distributed approach to efficiently crawl data from Facebook using their public API [5]. The crawler is, to the best of our knowledge, the only crawler capable of gathering Facebook data in depth by covering virtually all interactions within posts. However, the amount of Facebook data increases faster than the aggregated crawler performance. Therefor, some form of resource prioritization is required. That is, how to use a limited set of resources to cover most of the interactions. In order to manage this we have developed methods for estimating the interestingness of posts with regards to how much added value it provides compared to the already collected data. By applying ranking based on interestingness among post from the same page it is possible to create a prioritized list of posts to crawl. Using this approach it should, in theory, be possible to crawl virtually all interactions.
9.1.1 Challenges

In this paper we address how to prioritize among posts to crawl in order to cover as much of users interactions as possible with the available resources. This leads to the following questions: What properties of a single post can be used to measure interestingness? How can we estimate the interestingness of a post with regards to how much added value it provides to already existing data? We also address the issue of crawling and the challenge to further implement a crawler that not only crawl data in breadth, i.e., just getting the message of the post and some basic meta-data like number of comments, shares and likes. The crawler should also have capabilities to efficiency crawl posts in depth, in which all interactions around a post are gathered and still being efficient. For this we are forced to come up with a priority schema on which posts to crawl related to the questions above.

9.1.2 Contributions

We make three key contributions. First, in Section 9.4 we outline the design and development of an extended crawler that is capable to crawling all interactions from a given Facebook page. This crawler differs from previous developed crawlers in that it is capable of crawling posts in depth. Second, we have identified and described different measures that can be used to classify posts as being interesting or not related to various application areas. This work is presented in Section 9.5, and the proposed method is evaluated in Section 9.6. Finally, we use the classification of posts to increase the performance of our crawler in Section 9.7. Based on this experiment we present a significant decrease in crawling time, while still covering the vast majority, in excess of 99%, of the unique users and the interactions on posts.

9.2 Related Work

There have, to the best of our knowledge, been a lack of analysis of online social networks with the aspect of interestingness, validity and usefulness of data gathered. The study of Peters et al. [6] summarizes previous studies made to apply metrics in order to manage social media. The work by Peters et al. differs from our aim since the authors tries to address the issue of
Many studies exist that either directly or indirectly cover the challenge of crawling various OSNs. The studies conducted by Mislove et al. [8] are, to the best of our knowledge, the largest OSN crawling study available. From four popular OSNs; (Flickr, Youtube, LiveJournal and Orkut), 11.3M users and 328 M links were collected. Their analysis confirms known properties of OSNs, such as a power-law degree distribution, a densely connected core, strongly correlated in-degree and out-degree graphs and short average path length. Moreover, indirect studies of OSN crawling is present in the studies by Wilson et al., Crnovrsanin et al. [2, 3], where the authors transverse user profiles from Facebook. Wilson et al., Crnovrsanin et al. did collect roughly 70% of user profiles from various regional networks at high speed (averaging 10 MB/s) with quite limited resources. However, this study were conducted in spring 2008 and since then Facebook redesigned their site so it is no longer possible to crawl user profiles. More recently a study by Buccafurri et al. [9] discusses different methods to transverse the social network in a crawling perspective. Still, the restriction on crawling user’s profiles in not an issue in this work since we gather data from public groups only. As such, our work have substantial data to crawl and our challenge differs from Buccafurri et al.

Analysis of user interactions on OSNs have been a topic for several years. Garton et al. [10] identified the connection of people via computer networks as social networks in 1997. With studies from Grabowicz et al. [11] where the authors apply and evaluate social theories on OSNs. Also the studies by Ferrara et al. [12] are interesting as a study to map topology models on various social networks.

Studies to classify data include Linguistic Inquiry and Word Count (LIWC) [13] as a transparent text analysis program that counts words in psychologically meaningful categories. With LIWC it is possible to show attentional focus, emotionality, social relationships, thinking styles, and individual differences from just small sample of text. Diversity introduced by Bhattacharyya et al. [14] is an other usable factor to use for classifying data. The diversity factor is based on the relationship distance between two users.
9.3 Terminology and Application

9.3.1 Facebook Terminology

Within the Facebook context there are some terminology that needs explanation and clarification. Each user on Facebook has a number of friends, with whom a mutual agreement of relationship is established. Each user has a homepage called a profile that includes a wall on which it is possible to write, or post, content in the form of text messages, photos etc. Apart from users’ profiles and walls, the content within Facebook is divided into different groups, called pages, concerning certain interests. Users can add content to pages through posts and they can also show interest in pages by liking them. After liking a page future additions to these pages will be sent automatically to the user. However, a user do not need to like a page to be able to post to it, but some pages are restricted so that only page owners and selected users are allowed to post within them. The main page of Facebook is called the news feed or simply the feed. This feed contains a subset of both posts from friends as well as pages followed. For each post in the feed a user can either like, comment or share posts with friends. Finally, it is also possible to like other users’ comments, which we are referring to as commentLikes.

9.3.2 Application of Crawled Data

The crawled content depends on the application and how the data and SIN will be used. There are two distinct types of applications, recall and precision-based, that are based on different strategies for data collection.

- First we have the type of applications that are highly dependent of recall data or breadth data, which is possible to crawl in close to real-time. This involves for instance dynamic news feeds, i.e., pages where new and emerging topics are presented.

- The second type of applications are dependent of precision, i.e., deep data without the demand of real time coverage. This involves for instance Friend Suggestion [15] and Opinion Classifying [16], which could be accomplished by the use of SINs [4].

Generally it is harder to gather precision data in a timely fashion compared to recall data. In that regard our findings benefit the use of precision-based applications to a larger extent by enabling the generation of SINs. In
Figure 9.1, we show a real world example of such an interaction network, where each node represents a user and the size of the node shows the number of interactions related to that particular user, e.g., the large circle in the middle of Figure 9.1a represents the page owner. The edges in the graph represent comments (in purple) and the corresponding likes on these comments (in magenta). The SIN was generated by crawling the 20 most popular posts, with regards to number of comments, between Nov. 2007 and Aug. 2013 from the Facebook page called ‘(RED)’. This page have manually been selected as it is a page with over 1.5 million ‘likers’ with open post rights and are reasonable active with own posts. To increase visibility the likes on the post are not shown as the graph otherwise would have been lavished with such interactions. In Figure 9.1a using a clockwise orientation a somewhat separate interaction cluster is shown around eight o’clock (in magenta). In this interaction cluster the users interact among each other while the users surrounding the main influential user does not interact outside this cluster.

Figure 9.1b, depicts a SIN based on the same dataset as Figure 9.1a but here all nodes with eccentricity less than three have been removed to enhance cluster visibility, i.e., interactions with the post author. Eccentricity is a measure of distance and are calculated with respect to the post author [17]. There are several clusters of users liking just certain users’ comments, and we can assume that these users are already friends or following each other. The cluster identified from Figure 9.1a (in magenta) is even more visible in Figure 9.1b, in which all interacting users are shown in a direct circle to the post author.

9.4 A Platform for Making Interactions Accessible

In this section we describe a platform for making interactions accessible, which includes the agent-based crawler, an API for publishing the crawled data, and final applications of the collected data. The design, workings and crawling strategy of the crawler is outlined together with summary statistics regarding previously crawled Facebook pages. Also, various restrictions imposed by the Facebook API is discussed in relation to how it influence the crawling strategy.

The crawling technology is designed around a central agent controller that delegates tasks to the crawling agents. Each crawling agent makes use of a two-stage crawling strategy. All crawled data is returned to the
Figure 9.1: A social interaction network (SIN) generated from 20 popular Facebook posts. Nodes represent users, e.g. the post owner in the middle, where size indicates number of interactions and edges represent comments (in purple) and corresponding likes (in magenta). In (b) all nodes with eccentricity less than three, i.e., interactions with the post author, has been removed to enhance cluster visibility.

controller for parsing and database handling. This overall design makes the crawler modular and scalable. The agents communicate with the controller via a REST\textsuperscript{1} like interface, sharing JSON encoded data.

\subsection*{9.4.1 Agent Controller}

The agent controller is implemented in PHP and is responsible for which content to crawl. Currently there are a couple of hundred crawling agents operating under one controller, the exact number varies depending on the job at hand. The agent controller makes use of a queue to keep track of which content to crawl and when to crawl it. It is also possible for the controller to prioritize this queue favoring more important content, based on various metrics such as time aspects or interestingness.

\footnote{Representational state transfer}
9. Interaction metrics to support crawling prioritization in online social networks

9.4.2 Crawling Agents

The crawling agents are implemented in PHP and they are responsible for gathering either summary statistics regarding a specific Facebook page or the complete content from within a specific post, i.e., likes, comments, and commentLikes. Each agent is built as a loop where the agent calls the controller for a new batch of tasks which include, page_ids or post_ids. Based of the type of task the agent uses stage_one or stage_two to crawl the page or post respectively, as follows.

9.4.3 Crawling Strategy

The crawling strategy is divided into the following two stages. Stage_one is operating on a certain page specified by the controller, collecting summary statistics and meta-data related to that particular page. Stage_two is operating on a given post, collecting all interactions within that post. In Algorithm 1, we show an overview of a crawl of an entire page. The calls to stage_one and stage_two represents calls to Algorithm 2 and 3 respectively.

Algorithm 1 crawl of a single page

Require: page_id

\[\text{data} \leftarrow \text{stage_one}(\text{page_id})\]

\textbf{for} post_id in data \textbf{do}

\hspace{1em}writeFile(stage_two(post_id))

\textbf{end for}

\textbf{return} data

9.4.3.1 Stage_one Crawling

This stage is using the Facebook’s unique identifier of a public community (page or a group) to find the id of all posts, messages, photos, and links posted on the given community by admins and members, shown in Algorithm 2. The function makeFacebookRequest() represent a call to Facebook and the return is empty when there are no more data to request. This is a straightforward process that have to consider Facebook’s pagination [18] of API requests. Facebook have a limit on how many entities to be returned via their graph-API. This limit is by default set to 25 for most of Facebook’s internal calls. We have made a global increase so all calls we make to Facebook requests 200 entities. The higher this limit is configured, the more
likely a failure might occur on Facebook’s servers; since, every request has a limited time to be completed. For each failure we will need to re-crawl the number of posts equal to the limit that we have set. We have identified 200 to be the limit for our use cases with concern of the issues described above. A stage one crawl will simply access the feed connection on the community-id of the community we are interested in and continue to read the next page until we reaches the last page. This will give us a complete list of posts in a particular community.

**Algorithm 2 stage_one**

<table>
<thead>
<tr>
<th>Require: page_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>repeat</td>
</tr>
<tr>
<td>postData ← makeFacebookRequest (page_id/feed)</td>
</tr>
<tr>
<td>data ← data + postData</td>
</tr>
<tr>
<td>until postData is empty</td>
</tr>
<tr>
<td>return data</td>
</tr>
</tbody>
</table>

### 9.4.3.2 Stage_two Crawling

With the stage_two crawling we are gathering all social interactions on a post and we have to make the following requests for the posts gathered in Stage_one Crawling, as shown in Algorithm 3. First we gather the post itself, in which the post contains basic information like author, type and the message. In applicable cases we also get; link to the posted photo, or a video. In the first request, we also get a preview of comments and likes on the post, but this is not a complete view. In order to get all likes we iterate through the like handle (considering pagination here as well). To get all comments on a post we have to iterate through the comments handler and since each comment can have commentLikes we have to iterate through the likes handle for each comment as well.

For each post gathered in Stage_one Crawling we send at least three to four separate requests (assuming that there are no likes on comments), one for the post itself, one for the likes on the post (if there exist any), one to get information on who have shared the post and finally one to get all comments (if there exist any). If one of the responses is paginated we have to make consecutive requests to gather the complete view. This also means that for posts with a lot of interactions we have to make multiple requests to the graph. For instance, we have crawled posts with thousands of comments
9. Interaction metrics to support crawling prioritization in online social networks

Table 9.1: Distribution of posts based on both the number of comments and the time duration between post creation and time of the last comment. The table consists of interactions from 176 randomly selected pages covering 602,783 posts and 31,819,229 comments.

<table>
<thead>
<tr>
<th># of comments</th>
<th>0.25h</th>
<th>0.5h</th>
<th>1h</th>
<th>2h</th>
<th>4h</th>
<th>12h</th>
<th>24h</th>
<th>7d</th>
<th>1m</th>
<th>6m</th>
<th>&gt;6m</th>
<th>total posts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>count</td>
<td>%</td>
<td>count</td>
<td>%</td>
<td>count</td>
<td>%</td>
<td>count</td>
<td>%</td>
<td>count</td>
<td>%</td>
<td>count</td>
<td>count</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>16.7</td>
<td>10.5</td>
<td>11.6</td>
<td>11.7</td>
<td>11.0</td>
<td>9.5</td>
<td>1.4</td>
<td>0.7</td>
<td>0.2</td>
<td>0.2</td>
<td>151,882</td>
</tr>
<tr>
<td>2–5</td>
<td>2</td>
<td>7.7</td>
<td>8.4</td>
<td>11.4</td>
<td>12.7</td>
<td>12.3</td>
<td>18.0</td>
<td>12.8</td>
<td>12.8</td>
<td>2.3</td>
<td>1.2</td>
<td>209,822</td>
</tr>
<tr>
<td>6–20</td>
<td>1.4</td>
<td>3.2</td>
<td>6.9</td>
<td>9.7</td>
<td>10.9</td>
<td>18.7</td>
<td>15.6</td>
<td>21.6</td>
<td>6.9</td>
<td>4.0</td>
<td>1.1</td>
<td>113,259</td>
</tr>
<tr>
<td>21–50</td>
<td>0.1</td>
<td>0.1</td>
<td>0.6</td>
<td>1.6</td>
<td>2.8</td>
<td>11.7</td>
<td>18.4</td>
<td>38.1</td>
<td>15.1</td>
<td>9.1</td>
<td>2.3</td>
<td>52,048</td>
</tr>
<tr>
<td>51–100</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1</td>
<td>0.4</td>
<td>4.7</td>
<td>15.1</td>
<td>47.3</td>
<td>18.0</td>
<td>11.2</td>
<td>3.3</td>
<td>30,000</td>
</tr>
<tr>
<td>101–250</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.2</td>
<td>1.3</td>
<td>7.3</td>
<td>49.2</td>
<td>22.4</td>
<td>14.9</td>
<td>4.7</td>
<td>22,605</td>
</tr>
<tr>
<td>251–500</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>2.1</td>
<td>43.0</td>
<td>26.8</td>
<td>19.7</td>
<td>7.8</td>
<td>10,592</td>
</tr>
<tr>
<td>501–1000</td>
<td>0.4</td>
<td>0.7</td>
<td>1.1</td>
<td>0.5</td>
<td>0.3</td>
<td>0.4</td>
<td>0.7</td>
<td>29.3</td>
<td>27.2</td>
<td>27.4</td>
<td>12.1</td>
<td>7,038</td>
</tr>
<tr>
<td>1001–10000</td>
<td>0.5</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.4</td>
<td>1.3</td>
<td>0.7</td>
<td>13.6</td>
<td>27.2</td>
<td>38.0</td>
<td>17.4</td>
<td>5,489</td>
</tr>
<tr>
<td>&gt; 10000</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>2.1</td>
<td>18.8</td>
<td>12.5</td>
<td>41.7</td>
<td>25.0</td>
<td>48</td>
</tr>
</tbody>
</table>

Each with a few likes, where we have to make a request for each comment to get its likes.

**Algorithm 3** stage_two

**Require:** post_id

\[
\text{post data} \leftarrow \text{makeFacebookRequest (/post_id)} \\
\text{data} \leftarrow \text{data} + \text{post data} \\
\text{repeat} \\
\quad \text{like} \leftarrow \text{makeFacebookRequest (/post_id/likes)} \\
\quad \text{data} \leftarrow \text{data} + \text{like} \\
\text{until like is empty} \\
\text{repeat} \\
\quad \text{comment} \leftarrow \text{makeFacebookRequest (/post_id/comments)} \\
\quad \text{data} \leftarrow \text{data} + \text{comment} \\
\quad \text{if comment has likes then} \\
\quad \quad \text{repeat} \\
\quad \quad \quad \text{commentLike} \leftarrow \text{makeFacebookRequest (/comment_id/likes)} \\
\quad \quad \quad \text{data} \leftarrow \text{data} + \text{commentLike} \\
\quad \quad \text{until commentLike is empty} \\
\quad \text{end if} \\
\text{until comment is empty} \\
\text{return data}
\]
9.4.3.3 Post Re-crawling

Since the content in OSNs constantly evolve due to the interaction from its users, it is not feasible to make use of traditional web crawling techniques that are designed for gathering static (or semi-static) web pages. The addition, modification and removal of content within OSNs highlights the need to re-crawl content when it has been expanded or modified. Apart from crawling previously unseen content efficiently, the crawler also needs to detect when to re-crawl already gathered content in order to cover modifications of the content. This problem often correlates with the popularity of posts, since such posts more frequently receive additional interactions from users in the form of likes, comments and shares. Unfortunately it is not possible to get notifications from Facebook’s API concerning post modifications. As a result the only viable solution for gathering the modifications is to repeatedly carry out full stage_one re-crawls to determine whether posts have changed or not. This repeated re-crawl gets more computationally expensive as the number of pages and posts increase. It is therefore important to identify heuristic solutions that could assist in deciding if, when and how often to re-crawl content. In the end, it is a trade-off between capturing complete but outdated subsets of OSNs or to capture slightly incomplete subsets in a timely fashion.

Re-crawling estimation can make use of aggregated statistics of previously crawled posts to predict if and when re-crawling is needed. An analysis of a random sample of 176 crawled pages is presented in Table 9.1, in which each row represent an interval of comments and the columns represent the time interval for the last comment of a specific post. For example, posts with a range of 6–20 comments gets its last comment between four to twelve hours after the post is created with a rate of 18.7%. With this information we argue that the controller can make decisions if a post should be crawled in the first time or if the controller should wait i.e., covering all comments with higher probability. For instance, if a post have 200 comments and the age is just 10 hours (meta-data from the stage_one crawling) the ratio of posts is just 1.2% with the same distribution, or 1.5% looking at the cumulative post ratio of posts with 101–250 comments within 12 hours. Based on this information it would be better for the controller to wait an additional week before crawling the post, this will ensure a higher probability to cover all comments (58.0%). As another example, if a post have seven comments and the age of the post is just four hours there is 32.1% probability that
9. Interaction metrics to support crawling prioritization in online social networks

the post will not get any more comments. Another interesting observation from Table 9.1 is the distribution of number of comments. A vast majority (78.8%) of the posts have 20 or less comments and 96.2% of the posts have 250 or less comments. For posts with 21–50 comments, 73.4% of the last comments are made within a week from the post creation time.

An additional approach that helps us decide if and when a re-crawl is needed of a page or post is by studying the way SIN subsets form around a particular page or post. These patterns, together with how the user previously have interacted with similar posts or pages can assist in predicting the amount of future interactions. For instance, psychology studies show that people tend to interact when they agree with the content, but they would not initiate an opposing point of view [19]. Although, once an opposing point of view has been posted, others are likely to follow.

9.4.4 Organization of Crawled Data

When the crawling agents have returned the crawling results to the controller, the controller is responsible for storing the crawled data from posts and organizing the posts of interesting pages into the crawling queue. The controller also parses and organizes the data from crawled posts into a database. With the current crawling rate there are a few things to consider. We have written the parser in PHP and being a soft typed language it is probably not the best suitable language for big data parsing and analysis. However, the advantage of using the same language for parsing crawled data and the controller logic is the fast integration time. It is possible to directly call the parsing related functions without an external API directly from the controller. Secondly, since we like to insert data into a database we have seen that the choose of database and storage engine is of importance. We started out with a small PostgesQL database but as the dataset significantly grew the issue with a slow insert rate of PostgeSQL became a real issue. We were actually crawling data at higher speed than the database were able to write. Given that we made a decision to move towards MariaDB (a high performance fork of MySQL) together with TokuDB as storage engine. TokuDB is a storage engine built with insert performance in mind and claims up to 20X insert speed compared to InnoDB, which is the default storage engine of MySQL. We also used a high performing server with 48 GB of ram and a RAID-1 configuration of two SSD disks to improve the write speed to the database.
9.4. A Platform for Making Interactions Accessible

The parsing is made with respect of the output from Stage_two Crawling, and we parse the full post, likes, comments and commentLikes into respectively tables. We also extract all users interacting on each post and add users to the database. The database is organized with the following tables: user (storing user-id and name), application (storing information of various Facebook applications that are active in the posts), post (containing the post message and various extra data), page (the crawled pages and pages acting on crawled posts), comment (comments on posts) and like (containing both likes on posts and commentLikes).

Regarding parsing and organizing data from a re-crawl, we are currently adding the data into the database as a union, i.e., adding and updating the new and updated data and ignoring removed data. In addition, we also store the output from each Stage_two Crawling as a separate file, enabling us to re-parse and reorganize the data.

9.4.5 Privacy Issues Related to Crawling and Facebook Credentials

In this work, we have crawled public pages on Facebook in order to create SINs that represent the underlaying relationship between users within the Facebook community. The crawled data was published in public pages, which means it was available for anyone with a web browser to consume, even without a Facebook account. However, a Facebook account is required to be able to post new content since this requires a logged in account. The crawler requires a Facebook user account, but only for accessing Facebook's graph API.

As privacy always is a concern for the user, we have addressed the proposed threats in [20] by never disclosing user id or the name of the user. Also, the data we have gathered for this work is not exposed to user profiling or user behavior mining.

9.4.6 Facebook API Restrictions

By building the crawler as a distributed system, much like discussed by Chau et al. [21] we manage to fulfill the demand of a high crawling rate. The fact that Facebook only allows 600 API request per 600 seconds per registered application also ratifies the distributed design. We have one controller that is keeping track on current status and what data (in our
case which post or page) to crawl next. The proposed controller supports multiple crawling agents. Each agent runs independently and can have its own Facebook application-id and user-id. In order for the agent to not hit the limited requests from Facebook we have found that it is possible to reuse the same application-id for ten to fifteen agents. This number is based more on network latency than crawling constraints as more network latency affect the crawling time. With low latency to Facebook’s data center this number should be decreased and vice versa. Running more agents with the same application-id will hit Facebook’s 600/600 limits and force us to wait up to 600 seconds before the agent can continue. Currently we have just over one hundred active agents doing the actual crawling of data from Facebook and committing the data to our controller.

9.4.7 Application Programming Interface

The resulting raw data from the crawling is made available trough an API\(^2\), which allow third parties to use it for research purposes or application development. In addition to the raw data, we also produce social interaction networks (SINs) in real time around different content shared on OSNs. This functionality allows third parties to only worry about how to fit such social informatics within their purposes, instead of spending precious time gathering the raw data and producing the SINs themselves. Additionally, it is possible to more efficiently retrieve interaction based data using our API compared to Facebook’s internal API.

9.4.8 Crawling Statistics

During 2013 our crawler have gathered in excess of 200 GB of structured data, including; 225 million unique Facebook users, 33 million posts, 205 million comments and over 2.8 billion likes. The crawling agents used have been distributed around the world in both North America, Asia and Europe. Figure 9.2a shows the crawling time per post for stage\_one crawling, for a random sample of 10 000 pages, omitting pages with less than 250 posts. The mean crawling time for stage\_one crawling is 0.053 seconds per post. Figure 9.2b shows the crawling time per post for stage\_two crawling from a random sample of 1 000 000 posts. The median crawling time for this data set is 0.986 seconds per post, but there are also posts that require considerably

\(^2\)http://api.sincere.se
longer crawling time due to much content within them. The box plot on the top of both of the figures in Figure 9.2 illustrates the median, upper and lower fence together with entries outside these fences. The mean crawling time of trivial posts that could be captured within one single request is 0.23 seconds ($M = 0.225 \pm 0.196$). Where trivial posts are considered posts that only require one Facebook request in stage_two.

Based on the crawling statistics in Figure 9.2 and the fact that we currently have 100 million posts in the crawling queue, the estimated crawling time is roughly three years. If one takes into account the addition of new posts and that several posts will require re-crawls it is likely that the total crawling time will increase further, unless more resources are added and measures are implemented to increase efficiency.

### 9.4.9 Generalizability of Crawler

In this paper we describe how the crawler is used for collecting data from publicly available Facebook pages on the Internet. However, the crawler could easily be configured to also crawl private communities as long as the user credentials used are granted access to the content.

Apart from crawling Facebook communities it is also possible to modify specific parts of the crawler to cover other OSNs, such as Google+. This will however require some extra work as Google+ is organizing the data around open groups differently than Facebook. For instance, the verbs to identify
likes are called called plusoners and comments called replies on Google+.

9.5 Interestingness of Posts

In order to create a priority schema of interaction data, we must find a suitable evaluation method and metric to use. Interestingness, is a term commonly used in the context of data mining, as a measure to identify and rank data. Commonly the interestingness measure is used to emphasize conciseness, coverages, reliability, peculiarity, diversity, novelty, surprisingness, utility, and actionability [7]. These nine criteria are traditionally used to determine if data is of interest. Conciseness describes the ability for a limited set to still map and represent a full dataset. Given just a fraction of records, but with high coverage, it is possible to work with a large subset of the dataset. The dataset is reliable if it has a high ratio of given relationship in applicable cases. A pattern is peculiar if it is far from other discovered patterns. Elements have high diversity if they differ significantly when compared. Novelty describes how much new information a record adds to the complete dataset. It is virtually impossible to actually predict novelty as there are no way to know what a record adds before mining it. Surprisingness indicates a pattern that was unexpected and unpredictable. With utility we investigate how much use a pattern is given a specific goal. Actionability is the means of a pattern in respect of how much it can contribute in terms of future decision making in the specific domain.

This paper addresses the interestingness of posts in OSNs and also a way to define the value of posts. We argue that the more interesting a post is, the more valuable it is. The use of resources can be motivated by the value the post provides. In addition, the time spent on crawling a post is motivated by the value the post adds to the dataset.

We argue that posts globally are non-commensurable as they originate from different domains. Even for posts from the same domain (i.e., from the same page or by the same author) it is hard to compare the level of interestingness.

9.5.1 Meta Data Metrics

The following meta data metrics can be used when classifying posts’ interestingness.
Table 9.2: Matrix of interesting measures and the discussed metrics applicable for OSNs. X denotes a definite match of interesting measures and O denotes a possible identified match.

<table>
<thead>
<tr>
<th></th>
<th>Conciseness</th>
<th>Coverage</th>
<th>Reliability</th>
<th>Peculiarity</th>
<th>Diversity</th>
<th>Novelty</th>
<th>Surprisingness</th>
<th>Utility</th>
<th>Actionability</th>
</tr>
</thead>
<tbody>
<tr>
<td># of users</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td># of interactions</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Post lifetime</td>
<td>X</td>
<td>X</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Social capital</td>
<td>X</td>
<td></td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Keyword/topic</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Linguistic emotion</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Diversity</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Compress ratio</td>
<td>X</td>
<td></td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

* a we expect all the identified metrics can be useful, in different contexts

### 9.5.1.1 Unique Number of Users Interacting on the Post

We can calculate the total number of distinct users interacting directly to the posts, either by liking the post itself, making comments or liking some of the comments. When looking at this we also could give different score to the interacting users based on the type of interaction. We argue that there are two types of interactions, direct and indirect. A direct interaction is interactions directed to the post, including likes, comments and shares. Where an indirect interaction happens on relation to the post. For instance, when you like on a comment to a post.

### 9.5.1.2 Number of Interactions on a Post

By looking at the interaction metrics on a post we can determine the interestingness based on the sum of the number of likes, comments, commentLikes and shares. This is close to what we are looking at in 9.5.1.1, but here we are not identifying the number of users. Instead we look at the quantitative measures of the post; how many likes, comments and commentLikes the post have. Even here we can use a different score of the type of interaction and comments can be seen as a more valuable interaction than a like.
9.5.1.3 Post Lifetime

Given the delta time between when a post was created and the last modification on this post we can calculate the lifetime of the post. As seen in Table 9.1 the posts with 20 comments or less represents 78.8% of the population. 67.9% of the posts with 20 comments or less have a lifetime of less than twelve hours. For 73.7% of posts with more than 500 comments (which constitutes 2.9% of the population) the lifetime is more than one week. Challenges here are the fact that posts might get a very long lifetime only with a single like or comment. This can however be solved by normalization of the comment time.

9.5.2 Content Based Metrics

In addition to the meta data metrics, shown in Section 9.5.1, it is also possible to prioritize posts based on content based properties and the content itself. Below we list some five such metrics.

9.5.2.1 Amount of social capital

By looking at graphs of all posts or groups we can get an indication of a set of posts that would produce high social capital either using structural holes or weak ties. The graphs here could be based on likes or overlapping based on intersecting users.

9.5.2.2 Keyword and Topic Patterns

By mapping and identifying keywords and topics based on post content, it is possible to prioritize posts with matching pattern(s). This will lead to a keyword score showing the summarized score for each keyword the message contains.

9.5.2.3 Linguistic Emotion Profiles

It is possible to prioritize posts that fit well within certain linguistic profiles. Given a profile of linguistic feature range we can pick posts that the aggregated Linguistic Inquiry and Word Count (LIWC) profile [13] can be linked to. LIWC is a popular tool that calculates the frequency of words in a text to match each of the 68 categories representing linguistic dimensions, psychological constructs and personal concerns [22].
9.5.2.4 Diversity Based

When defining the prioritization queue we can define either a target similarity or diversity. Here we can use a combination of *Amount of social capital* and *Keyword and Topic Patterns* to define similarity or diversity. We can also look at the work by Bhattacharyya et al. [14] to classify the diversity.

9.5.2.5 Compress Ratio of Posts

Given two posts, or a group of posts, we can compare them by the easiness level in social compression. Here compression means the ability in analyzing some high level properties to reduce the amount of comments to a much smaller level, much like the work by Cilibrasi et al. [23]. In other words, if some posts are highly compressable, then even if the posts contain a lot of comment or interesting information, they can be very useful especially for search functions.

9.5.3 Summary

Based on the metrics discussed in the previous subsections we can evaluate a single post and also groups of posts and by this get an estimate of the interestingness. The method of grouping by *Meta Data Metrics* are shown to best be suited for comparing posts against each other as we are investigating metrics like number of users and lifetime. Where as the *Content Based Metrics* method suits the classification among groups of posts better.

In Table 9.2 we show a matrix of interestingness measures with the identified metrics available on posts in OSNs. From the table we see that the utility can be matched to all metrics discussed above. It is just a matter of the application of OSN data.

9.6 Evaluation of Interestingness

Based on our dataset we are interested in investigating the possibility to predict which posts that have the potential to generate the highest number of interactions and the most number of unique users. We argue that higher number of interactions and unique users lead to higher interestingness. In the scope of crawling, it is of interest to concentrate the (crawling) resources to a subset of posts while still covering as many interactions and users as possible.
9. Interaction metrics to support crawling prioritization in online social networks

Figure 9.3: Shows the number of interactions and the number of unique users of the same dataset sorted by the number of comments (green), number of likes (blue) and the post lifetime (red) by descending order. The dataset covers posts from the Facebook page called ‘(RED)’ and spans between 2007–11–07 and 2013–08–02.

9.6.1 Similarity Plots

Figure 9.3, illustrates how the total number of interactions and number of unique users are growing respectively, in terms of the number of posts. Each line in the two plots represent the internal sorting of the posts. We show sorting based on number of comments (green), number of likes (blue) and post lifetime (red). Both of the plots use the same dataset, posts from the Facebook page called ‘(RED)’. Also, in this work we threat all types of interactions the same. We do not apply weighting although one can say that a comment is a much stronger interaction than a like and a commentLike is even weaker. It should also be noted that a post made recently have a short lifetime and a post made earlier potentially could have a longer lifetime since it is calculated as a delta between the creation time of the post and the time of the last comment. That explain why we see the separation between the red, the green and blue lines. The dashed line shows where the derivate approaches zero and we only see a minor increase of interactions or unique users. In other words the posts above the line could be considered to be ignored in order to improve performance as they slightly supply new
9.6. Evaluation of Interestingness

Figure 9.3a shows the increase of the number of total interactions in relation to the number of comments, the number of likes and the lifetime of the post. There is only a minor difference if sorting are done by the number of comments or number of likes since the lines follows each other. However, considering the lifetime plot (red line), which is slightly different shaped. This can be explained with how the lifetime are calculated, i.e., the time of the latest comment minus the creation time of the post. This gives a poor representation of the interaction rate over time in a post. The same behavior can be seen in Figure 9.3b, which illustrates the unique number of users instead of total number of interactions added per analysed post. By using number of likes (blue line) in Figure 9.3a, it is possible to capture approximately 500,000 by crawling the most liked 1000 posts. Crawling an additionally 1,500 posts would not add much in terms of number of interactions collected.

9.6.2 Statistical Similarity Test

It is of interest to test for statistical similarity between the three parameters; likes, comments and lifetime. A relevant question is; Does it matter which of the three we use when it comes to prioritizing posts.

Figure 9.4 depicts a correlation plot of how each of the factor relates to each other with regards to the number of interactions (from Figure 9.3a). Although it is quite hard to see in the plot, most of the points are at 500,000 interactions where the three perspectives gets closer to the same number of interactions i.e., closer to the dashed line.

The Kruskal-Wallis similarity test [24] is used to investigate the statistical behavior and to compare the three datasets as we can not assume normality over the datasets. The output from the test shows $\chi^2(2) = 539.3 \ p < 0.05$, rejecting the null hypothesis of dissimilarity, the factors are statistically similar when it comes to predicting and indicating the number of interactions.

9.6.3 Similarity Test on Multiple Pages

Determining that the number of interactions and number of distinct users follows either the number of likes, number of comments or the post lifetime for the page ‘(RED)’ is an interesting finding. It is also of interest to see if
this is valid for all pages on Facebook. For this we have randomly sampled 176 pages on Facebook, crawled and parsed and analysed them with our crawler in the same way as for the ‘(RED)’ page.

Figure 9.5 shows the resulting confidence interval at the 99% level for statistical significant similarity (also using the Kruskal-Wallis similarity test) for the 176 different pages investigated in relation to the total number of posts each investigated page contains. Red shows the acceptance of similarity and blue shows no statistical similarity. The lower interval for rejections and upper limits for similarities are removed as it moves towards zero for rejections and close to infinity for acceptance. Figure 9.5a shows the number of posts in perspective of number of interactions and Figure 9.5b shows the number of posts in perspective of number of unique users. We show a strong statistical similarity of the factors from both the perspective of number of interactions and the unique number of users if we have pages with more
9.6. Evaluation of Interestingness

Figure 9.5: Mean and upper/lower confidence intervals (99%) for the number of posts needed in a page to produce statistical similarity of the factors presented in Figure 9.3. Red shows the statistical significant similarity identified and blue shows where there is no statistical significant similarity presented in perspective of number of interactions in (a) and number of users in (b).

than 9011 or 9320 posts respectively. For large pages we argue that we get the same number of interactions or number of unique users based on either of the three factors (comments, likes and lifetime) chosen. For smaller pages (2774 and 5725 respectively) there are no statistical similarity between the different factors. For pages with posts between 2774–9011 (Figure 9.5a) and 5725–9320 (Figure 9.5b), no assumption of statistical similarity can be made.

9.6.4 Prioritizing Posts on Open Pages

By using meta data metrics like number of comments and number of likes, we argue that it follows the intuitive idea of number of interactions. Considering a post with the most number of interactions to be interesting, there is no statistical difference if either the number of comments or likes on a particular post is used. As it have been shown to predict the post with the highest number of interactions. This finding can be used in different applications. For example our application with an efficient crawler we can prioritize which posts to spend crawling resources on for a full crawl (a stage_two crawl) using the data collected from the much cheaper stage_one crawling. Obviously the best result can be expected by combining the number of comments and the number of likes. But we have shown that for pages with a high number of post there is a statistical similarity between the factors. Therefore we can expect the same result with just one of the factors.
9.7 Discussion

The information we get from the posts from the stage_one crawl is metadata like creation time, author, number of comments, likes and modification time. Given this information it is only possible to use a few of the proposed metrics from Section 9.5. For instance, is not possible to know the Unique Number of Users Interacting on the Post discussed in Section 9.5.1.1, the Compress Ratio of Posts or Amount of social capital as these methods require information that only are available when the post have been crawled completely. The same applies to the interestingness score based on the methods in Section 9.5.2, Content Based Metrics.

Considering Number of Interactions on a Post we have shown in Section 9.6 that we get an estimate of the number of interactions by looking at either the number of likes, comments or shares. This way we can get an estimate of which of two posts that will have the highest number of interactions.

Evaluating the dataset with regards to the meta data metrics such as number of comments or number of likes we can prioritize what to crawl for the next available agent. We can also use the lifetime of the post to see how long it has been between the post were made and the last comment made on that particular post.

What we can see in Figure 9.6 is the fact that most of the investigated pages have an exponential shape of the number of interactions in relation to the number of posts. We can say that when the derivate of interactions per posts goes to zero we will not gain many new interactions per new post we crawl. In other words when we are able to detect that the number of interactions no longer significantly increases by crawling more posts we can stop crawling that page and use the resources to crawl posts from other pages. Therefor, we have a cut-off point where we just marginally gain more interactions. Investigating where this point is is not a straightforward task as this is dependant on external factors from the pages. In Figure 9.7, we see how many posts the had to be crawled in order to cover 99% of the total number of interactions within the 176 randomly selected pages. The median at 35.7% of crawled posts per page gives us a good indication that we can save a significant amount of resources by reducing the number of posts crawled per page while still covering close to all interactions.
Figure 9.6: Number of interactions per posts, shown for 176 different pages, each color represent one page.

Figure 9.7: Cut-off percentage to cover 99\% of all interactions per page.
9. Interaction metrics to support crawling prioritization in online social networks

Figure 9.8: Proposed savings in number of posts (in black) and total crawl time (in red), based on skipping less than 99% of the interactions on the page ‘(RED)’. Note that this is specific to just this page.

9.7.1 Crawling Time

By applying prioritization of which posts to crawl based on the findings in Section 9.6 we can reduce the crawling time of the page ‘(RED)’ from 31 510 seconds to 16 226 seconds, a saving of 48.5%, if we stop crawling at the intersection of the solid and dashed line in perspective of the number of likes on a post blue in Figure 9.3b. The cost of this reduction is missing 0.5% of the total number of unique users.

The same applies in the perspective of the number of interactions. We can reduce the crawling time to 16 522 seconds or by 48.5% while only missing 0.5% (2 597) of the total number of interactions (509 935) on the page ‘(RED)’. This can also be seen in Figure 9.8, where we see that by accepting a loss of 0.5% of the total number of interactions we only need to cover 17% of the total number of the posts of the ‘(RED)’ page. By doing this we can save close to 50% of the total crawling time according to the red line in Figure 9.8.
9.8 Conclusion

In this paper we present a novel crawler designed as a distributed system, capable of mining all interactions from posts in open pages on Facebook. This crawler have, over a period of twelve months gathered 263 million unique Facebook users interacting on 36 million posts through 308 million comments and 3.5 billion likes.

We also present eight methods for applying interestingness metrics to posts in OSNs to support ranking. Methods to predict the total number of interactions and unique users interactions on posts are compared and evaluated. For large pages we argue that we get the same number of interactions or number of unique users based on either of the three factors (comments, likes and lifetime) chosen. This have been verified against a sample of 176 randomly selected pages from our crawled dataset.

The results from the statistical tests of correlation between the number of users and the total number of interactions are applied to our crawler and based on this we can reduce the crawling time by up to 48.5%, while still covering 99.5% of all interactions.

We present an estimation of post lifetime related to number of comments, see Table 9.1. This distribution is based on the random sample of 176 pages that consists of 602783 posts and 32 million comments.

9.9 References


9. Interaction metrics to support crawling prioritization in online social networks


9.9. References


9. Interaction metrics to support crawling prioritization in online social networks


ABSTRACT
The use of online social networks poses interesting big data challenges. With limited resources it is important to evaluate and prioritize interesting data. This thesis addresses the following aspects of social network analysis: efficient data collection, social interaction evaluation and user privacy concerns.

It is possible to collect data from most online social networks via their open APIs. However, a systematic and efficient collection of online social networks data is still challenging. Results in this thesis suggest that the collection time can be reduced to 48% by prioritizing the collection of posts.

Evaluation of social interactions requires data that covers all the interactions in a given domain. This has previously been difficult to do. In this thesis we propose a tool that is capable of extracting all social interactions from Facebook. With the extracted data it is for instance possible to illustrate interactions between different users that do not necessarily have to be connected. Methods using the same data to identify and cluster different opinions in online communities have been developed and evaluated.

The privacy of the content produced and the end-users’ private information provided in social networks is important to protect. Users should be aware of the privacy-related consequence of posting in online social networks in terms of privacy. Therefore, mitigating privacy risks contributes to a secure environment and methods to protect user privacy are presented.

The proposed tool has, over the period of 20 months, collected 38 million posts from public pages on Facebook which include, 4 billion likes and 340 million comments from 280 million users. The data collection is, to the best of our knowledge, the largest research dataset of social interactions on Facebook, enabling research in the area of social network analysis.