Measuring User Influence in Github: The Million Follower Fallacy

Ali Sajedi Badashian
University of Alberta
Edmonton, Canada
alisajedi@ualberta.ca

Eleni Stroulia
University of Alberta
Edmonton, Canada
stroulia@ualberta.ca

ABSTRACT

Influence in social networks has been extensively studied for collaborative-filtering recommendations and marketing purposes. We are interested in the notion of influence in Software Social Networks (SSNs); more specifically, we want to answer the following questions: 1) What does "influence" mean in SSNs? Given the variety of types of interactions supported in these networks and the abundance of centrality-type metrics, what is the nature of the influence captured by these matrics? 2) Are there silos of influence in these platforms or does influence span across thematic boundaries?

To investigate these two questions, we first conducted an in-depth comparison of three influence metrics, number of followers, number of forked projects, and number of project watchers in GitHub[1] (the largest code-sharing and version-control system). Next, we examined how the influence of the top software engineering people in GitHub is spread over different programming languages.

Our results indicate (a) that the three influence metrics capture two major characteristics: popularity and content value (code reusability) and (b) that the influence of influentials is spread over more than one programming language, but there is no specific trend toward any two programming languages.

CCS Concepts

•**Software and its engineering** → **Software libraries and repositories;** General programming languages; •**Human-centered computing** → **Social network analysis;** Collaborative and social computing; •**Information systems** → Crowdsourcing;

Keywords

Mining software repositories, Software Social Networks, Programming languages, Social influence, Github

[1]http://www.github.com

CSI-SE '2016 Austin, TX, USA

ACM ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

1. INTRODUCTION

Recognizing the scope and magnitude of the influence of individuals in social platforms is important for value-added services on these platforms. For example, businesses may want to target their advertising to people with influence who may serve as "evangelists" for the advertised product or service to the community at large [16]. "Influencer marketing" as a means of advertising and marketing facilitates transfer of information regarding your products and items from your company to the customers [8]. It is even observed that voting patterns are influenced by news in the Facebook feeds of the influential users [6]. This is why the notion of influence is now being widely studied in social networks [11, 9, 1, 28]. Most of this work is generic and does not attempt to recognize influentials in a specific discipline. We are interested in the study of influence in Software Social Networks (SSNs), and more specifically in GitHub, and how it is different/similar from regular social networks. We believe that focused studies of influence in different social platform are useful for domain-specific services. We observe that, currently, SSNs are expanding their service offerings. For example, Stack Overflow[2] launched Stack Overflow Careers in 2011, and Linkedin[3] provides job-search functionalities. Knowing the real influential aspects of a user in specific software development communities and environments can help in utilizing those people. There are many SSNs that propose professional recommendations and advertisements, based on technical assets of their users, their connections and influence. For example, Stack Overflow recommends questions and also jobs for its users based on their achievements in different areas and their influence on the community (i.e., the up/down votes to their posts). This shows that the software community needs more attention of the study of influence and influentials.

It is difficult to identify "influentials" because there are too many traces of work and communication in the SSNs. In fact, identifying the right practices, strategies and structures that are unique to a specific aspect of influence are challenging. As a result, researchers use a range of metrics to identify influencers [11]. Cha et al [9], in their highly cited "Million follower fallacy" paper, demonstrated that, in Twitter, users with many followers are not necessarily influential in terms of being retweeted or mentioned, but the most influential users can hold significant influence over a variety of topics. Dubois and Gaffney [11] studied influence in political parties and reported similar findings: "number

[2]http://www.stackoverflow.com
[3]http://www.linkedin.com

of followers" and "eigenvector centrality" measure traditional political status, while measures considering message quality (e.g., number of tweets containing a specific keyword) are indicators of content value.

In this paper, we report on a study of influence on GitHub, parallel to the studies of Cha et al. and Dubois and Gaffeny, focusing on identification and comparison of influence metrics in GitHub. Furthermore, aiming to understanding whether influence is thematic, we analyze the influence of GitHub members over different languages. Our work is driven by the following high level research questions:

- What does influence mean in GitHub? Since various types of interactions between the users are supported in this network, what is the interpretation of various influence metrics derived from abundant centrality-type measures?

- Are there high reputed users as silos of influence with extended influence over different communities? Or the users' influence is focused across a specific programming language as the thematic boundaries?

Our study revealed the following two interesting findings.

1. The numbers of "followers" a GitHub member has indicates his/her popularity and fame. On the other hand, the number of repositories the developer owns that have been "forked" is an indicator of the value of the content produced by the developer. The number of GitHub members "watching" the developer's GitHub activities captures a notion of influence between the above two indicators.

2. Influence is spread across different programming languages. In other words, influential GitHub members exert their influence in more than one programming languages. However, we found no specific trend over any two specific languages.

The rest of this paper is organized as follows.

2. LITERATURE REVIEW

2.1 Motivations Behind Studying Influence

Studying influentials is important in marketing [9]. Market researchers are interested in who is best in a network to spread an idea [11]. This concept has an important role in dissemination of information like opinion propagation and viral marketing [13]. Influential people have great value in business, from their power of content creation to convincing their fellows in a variety of fields to reacting to the events and news [2]. Inspired by Katz and Lazarsfeld's idea [15] in targeting the most influential people to save the cost, companies find it beneficial in spreading a product or technology.

According to the "two-step flow hypothesis" (multi-step flow model), new ideas flow from the mass media to influentials that, in turn, pass these ideas on to others that are more passive in terms of information seeking" [14].

We specifically consider influence as the ability to cause desirable and measurable actions and outcomes in people [3, 16, 23, 15].

2.2 Influence in Social Networks

There have been a lot of studies targeting influence and influential people in social networks [28, 15, 20, 9, 11, 25, 7, 30]; Sun and Vincent [28] define influential users based on the relationship between online posts. They proposed a graphical model to represent relationship between online posts on a topic. With this information, their model identified most influential users based on the relationship among postings. Patil et al. [20] studied the trusted and untrusted relationships (friend of friend and enemy of friend) in Epinions , the general customer review website. They found that there is strong agreement between collective opinions of a user's friends/trustees and the user's regarding another user.

2.3 Implications of the Contributions of the Developers in SSNs

Focusing on software engineering area, according to Storey et al. [26] a paradigm shift in social software engineering is emerging. This includes active engagement in online software development which leads to high communication and fast diffusion of technologies in the communities. In fact, social media has changed the way developers collaborate [27]. Noting this, Storey et al. mentioned the roles of social media in collaborative activities at the team, project and community levels. They are not only forums, wikis and social networks that reflect these team-based activities, but also IDEs utilized mechanisms that integrate with online version control systems.

Dabbish et al. believed that a software project's success is influenced by the visibility of its developers' activities through motivating others [10]. In fact, developers choose about projects to contribute by considering the projects with frequent contributions or with contributions from developers with high status [29, 26].

According to Marlow et al. [19], in software social networks like GitHub, the transparency of the contributions makes the users able to construct impressions of one's work. These impressions lead to future interactions and trust and utilization of someone's source code [26]. Pham et al. investigated this with GitHub users and concluded that this trust difference is evident in the differences how the project managers behave regarding the contributors of a developer they know and the developers they do not know. The later ones usually undergo more accurate examinations [21].

Marlow and Dabbish [18] interviewed a group of GitHub users (employers and job seekers) and stated that employers obtain clues from developers' GitHub profiles (including activities, skills and interactions). The interesting fact was that these clues were stated more important than the developers' resumes. The real fact in open source software engineering projects is that developers participation in the projects has some "value" more than financial assets [18, 26]. The developers show their proficiency level in these contributions and spread their work record by the social clues available in GitHub. This is why Frèdèric Harper [12] –the senior technical evangelist at Mozilla[4]– mentions how to obtain influence and personal branding using online software social networks like GitHub and Stack Overflow. Growing the network and becoming more visible is accessible through participation in these online repositories. Then measures like reputation in Stack Overflow or number of followers in

[4]http://www.mozilla.org

GitHub reveals kind of overall evaluation for the developers that they maintain and expand while performing their daily tasks.

2.4 Metrics of Influence

Blincoe, et al. [5] studied the “following” as a metric of influence in Github and found that influentials guide their followers to new projects. They believe that regarding influencing others, popularity can be more important than contribution in projects. In another study, considering cross references (links) between projects, Blincoe, et al. [4] identified software ecosystems. Considering these references as technical dependencies between Github projects, they argued that the User’s social influence aligns with these technical dependencies within ecosystems. They found that popular ecosystems are mostly centered around tool support for software development. This is true for the “number of forked projects” popularity measure in our study.

Generally speaking, there are different measures and interpretations of influence in social networks. Cha et al. [9] considered three influence parameters; number of followers, mentions and retweets of a user. They compared these metrics against each other and found that these metrics describe the influencer differently; while being a user highly followed shows that he is a popular user (like public figures or news sources), high level of being mentioned in other tweets is an indicator of name value. High levels of retweeting (people retweet one’s tweets) on the other hand, shows content value (e.g., content aggregation services or businessmen). As other metrics of influence, Kelman [17] defined compliance, identification and internalization as three different possibilities of how users react to an influencing attempt. Isabel [1] extended this study to include two other indicators *neglect* and disagreement. This type of influence could be considered when the influence is propagated by messages or communication content.

We are interested in studying influence in software social networks. In our previous work [24], we found that there is moderately high correlation (above 0.5) between developers’ popularity, development and management characteristics in GitHub. This study aims for a deeper understanding toward popularity and influence metrics, to obtain a better indication for the available measures in software social networks as a social network. We attempt to study influence metrics in Github by replicating two famous studies in this area [9, 11] regarding regular social networks and adding a variety of other analyses including influence analysis over different languages.

3. THE DATA SET

We used a MySQL database dump [22] (with a size of 25GB) containing information of 6,205,555 GitHub users and the meta-data regarding their social and development contributions. We consider a Github account a user. In fact, this can be a user, a company or an open source library/platform. Since for some accounts, it is not exactly known if they are a company or a person, we prefer not to do personal judgments on filtering the accounts.

Then, for each user, u, we computed the cardinalities of three different sets: (a) the number of the user’s followers, $u.follow$; (b) the number of distinct GitHub members who forked any one of the projects, of which u is the owner, $u.fork$; and (c) the number of GitHub members who watch any one of the projects of which u is the owner, $u.watch$.

It is important to note here that we can compute all three measures only for the subset of the GitHub members who participate to at least one project; the $u.fork$ and $u.watch$ measures are not defined for user who do not have any projects associated with them. This subset includes about half of the GitHub membership, namely 3,524,603 users. A second interesting observation regarding this set is that, even though the measures are defined for all its members, in a very large number of cases (one or more of) these metrics are 0, since many GitHub users have no followers and do not own projects “interesting” enough to watch or “useful” enough to fork.

We are motivated to consider these three metrics inspired by Cha et al.’s study [9]. Like many other networks, the $u.follow$ measure is hypothesized to be measure of popularity. $u.fork$ is assumed to be related to usage and can be considered equivalent of retweet in Cha et al.’s study. The $u.watch$ metric can express interest and be considered equivalent to mention in Twitter.

4. METHODOLOGY AND RESULTS

The methodology of our study is the same as that of Cha et al. [9] and Dubois et al. [11] and it involves three steps.
Step1: We first analyzed the pairwise correlation of the three measures, using the Spearman correlation. The purpose of that step was to examine whether or not these three measures capture the same notion of “influence”.
Step2: Next, we performed an in-depth qualitative analysis of the similarities and differences of the top 30 influential GitHub users according to each of the three measures. Our objective was to try to tease apart the differences in the profiles and befaviours of the three sets of influentials, corresponding to the three above measures.
Step3: Finally, considering the top 10 programming languages in GitHub, we calculated the $u.fork$ and $u.watch$ metrics over each language for all users. Then, calculated and sorted the best ranks of each user based on each sub-metric ($u.fork$ and $u.watch$ language based measures). After that, using the Spearman correlation, analyzed the pairwise correlation between the influence of the user over the first language of choice of each user and the next choices. This is to find out if a user’s influence is spread between different languages or it is focused on one language. This is motivated by Cha et al.’s study where they found that influentials can hold significant influence over a variety of topics [9] as we mentioned before.

4.1 Correlations Bewteen Influence Measures

Table 1 reports the pairwise Spearman correlation between the three influence measures. The $u.follow$ measure is only weakly correlated with the two other measures, while the $u.watch$ measure is moderately correlated with $u.fork$.

Table 1: Spearman’s rank correlation between influence metrics: P-value<0.01 (Significant at the 99% level)

Metrics	Correlation
$u.follow$ vs $u.watch$	0.38**
$u.follow$ vs $u.fork$	0.36**
$u.watch$ vs $u.fork$	0.55**

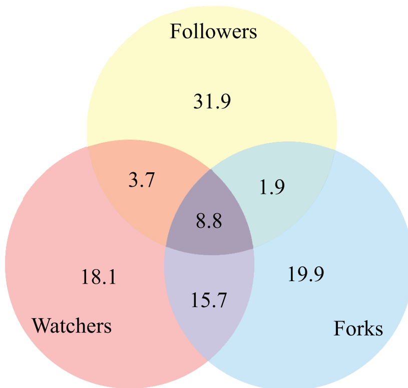

Figure 1: The top-100 influentials across measures, shown in a venn diagram: the values are normalized so that the total is 100%

Then, in order to eliminate the possibility that these correlations are due to the large numbers of GitHub members with one or more "zero" values, we performed a more focused analysis, examining only the relatively influential users. We sorted the developers based on each of the three measures and we identified three different sets of users for each measure: the top 10^{th} GitHub users, the users in the top 1^{st} percentile, and the top 100 users. We then calculated the pairwise Spearman correlations between these sets, which are reported in Table 4.1. None of these correlations is significant, which indicates that the three measures are completely independent and capture different notions of "influence". There is only a weak correlation between *u.follow* with *u.watch* and *u.fork*. But *u.watch* and *u.fork* are more connected measures.

The three top 100 lists include 216 distinct accounts (that can be either individuals, companies or libraries) as shown in Figure 1. Note that the values depicted in the figure are percentages over these 216 people. Interestingly, there were only marginal overlap between *u.follow* and two other top lists (around 11% and 12%). The *u.watch* and *u.watch* lists however are sharing some more people (still just around 24%). This validates the results of the first step on *u.follow* measure being separated from the two others, but *u.watch* and *u.fork* being moderately connected.

4.2 Looking into The Profiles of Influentials

Then, in order to gain an insight on the nature of each metric, we took a deeper look on the top users' profiles on each category manually. We started investigating 30 top accounts in the three lists by checking their GitHub profiles. Then we looked on their roles and appearance on the web (i.e., their personal page, company contributions, Wikipedia and social networks like Twitter). These roles are the ones that are directly mentioned in at least one of the resources in the web. Then, we classified these roles in seven categories as it is shown in Table 3.

All the accounts in the top 30 list based on the *u.follow* influence metric are people. Many of them are public figures; Linus Torvalds, Chris Wanstrath and PJ Hyett (Cofounders of GitHub). There are also other enterprise roles (e.g., David Heinemeier Hansson; Founder and CTO at BaseCamp) or mixed roles (e.g., John Resig; lead developer of jQuery JavaScript library; Entrepreneur and Dean of Computer Science at Khan Academy). There is no account tagged with "ORG" tag in this category. We found that the focus is on the public figure in this category and conclude that this metric measures popularity the most. These are famous and popular people in software engineering (not companies or organizations), but due to the nature of GitHub, only the roles of active developers are highly being followed and more managerial roles are not seen here.

Considering the top 30 accounts based on *u.forks*, however, completely different results were obtained. They contain more accounts tagged with "ORG" than "USR". 25 accounts in the top 30 based on *u.fork* are organizations, usually big software companies (i.e., Facebook, Twitter, GitHub, Mozilla, Heroku, Apache, Google, Udacity, etc.), and 13 of these 25 are specifically libraries and frameworks (e.g., jQuery, Bootstrap and Angular-ui). On the other hand, only 3 of the top 30 in this category are developers or engineers, and, we argue that this measure does not indicate personal preferences. Since forking means code usage and content spreading, it is showing "content value". In fact, they are influentials in that they produce reusable code which is adopted by communities of developers.

Similarly, less than half of the top 30 accounts based on *u.watch* measure are tagged with "ORG" which are big companies (e.g., Google, Facebook, Github, Adobe, Yahoo, etc.) or libraries and frameworks (e.g., jQuery and elasticsearch). However, in this category, there are also developers (e.g., Kenneth Reitz who is Python Overlord and Evangelist at Heroku or Sindre Sorhus who is a Github developer) and some entrepreneurs (e.g., James Halliday; founder of Substack, or, Victor Felder, Co-Founder of Kwak.io). Comparing with *u.fork*, there are less ORG and libraries in this group, but more engineers and developers that shows a semi-personal process. We interpret this group as a combination of the two others so that it indicates both popularity and code reuse. We can relate this group generally to "interest" of the users toward a project.

These results are similar to Cha, et al.'s study: "The most connected users are not necessarily the most influentials" [9]. As a result, we found similarities and differences with that study as are indicated in Table 4. In Twitter, the three metrics indicate three different measures (that respectively account for public figures, celebrities and content value). However, in Github, due to the technical aspects of the network, the measures are limited to two; "number of followers" accounts for popularity and fame, and, "number of forked projects" shows code usability. "number of watched projects" is somewhere in between the two, but more connected to "number of forked projects" and can be interpreted as "interest".

Like Twitter, having lots of followers doesn't necessarily mean being actively influencer in action (although it can be in some cases, due to the technical nature of GitHub), but there are better measures for real influence which are number of forked projects and number of watched projects influence metrics.

4.3 Influence over different languages

Since we are studying 10 different languages separately, we focused on the top 1% followed users (37,022 users) to avoid too many zero values. We want to see if users are influential in one, focused language or they are influential in several

Table 2: Spearman's rank correlation between influence metrics for top users:
**** P-value<0.01; significant at the 99% level**
*** P-value<0.05; significant at the 95% level**
- P-value>0.05; non-significant results

	Top 10%			Top 1%			Top 100 users		
	based on number of followers			based on number of watchers			based on number of forks		
u.follow vs *u.watch*	0.49 **	0.40 **	0.49 **	0.52 **	0.23 **	0.33 **	0.23 *	0.17 -	0.39 **
u.follow vs *u.fork*	0.46 **	0.39 **	0.40 **	0.52 **	0.26 **	0.21 **	0.27 **	0.26 **	0.13 -
u.watch vs *u.fork*	0.75 **	0.70 **	0.67 **	0.91 **	0.67 **	0.64 **	0.86 **	0.63 **	0.30 **

Table 3: Different roles the top users (based on each measure) have

Role / Based on measure	Engineer, developer or designer	CEO, founder, co-founder, inventor or entrepreneur	Speaker	Author or writer	Education; professor	Open source library, framework, platform, etc.	Organization
u.follow	**25**	**12**	**4**	**7**	2	0	0
u.watch	**11**	**5**	0	0	1	**8**	**18**
u.fork	3	2	0	1	2	**13**	**25**

languages. First, we obtained the top 10 programming languages (see Table 5). The projects of these languages are around 85% of the total projects in GitHub with a specified, dominant language. Then, since number of followers is not project dependent, we ignored this metric and obtained the users' number of watched and forked projects influence measures over each one of these languages.

Figure 2 shows the number of languages the users have watcher or forked project in. Regarding both number of forked and watched projects, most users are influential over less than four different languages. However, interestingly there are a relatively small number of users who are influential over all the 10 considered languages. In average, each user has "forked projects" over 1.7 different languages (median=2) and "watched projects" over 2.5 different programming languages (median=3).

We found that in average, each user has almost one more watched project than forked project (by another developer in the community). There is a strong alignment (between 0.64 and 0.84) between number of forked projects and number of watched projects within each language. This alignment is higher than the general results of Table 4.1. This comparison reminds us the nature of these two metrics that are directly based on programming language. Also the fact that in many cases in real world, developers' projects are first being watched, then forked after a while. This way, they may choose which projects they need to fork by the notifications they get from the projects they are watching.

An important outcome of the multi-language traces of impact hypothesizes an extended notion of influence over more than one focused language.

So we looked for the correlation between number of forked (or watched) projects measure over different languages and observed that the measures in no two languages are correlated (correlation < 0.3). This means that there is no trend (or pattern) over any two specific languages. But this does not reject the hypothesis that the users' influence is spread over different languages. In order to check this broadness of influence of the users over different languages (without limiting it to a specific language for all the users), we obtained the rank of the user in all the 10 languages of any individual (regarding number of watched/forked projects) separately.

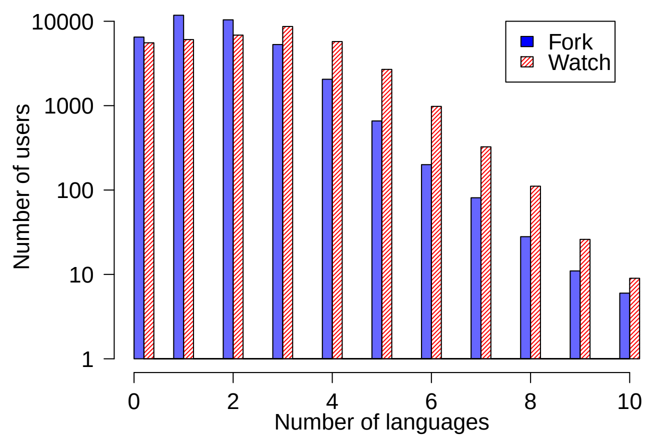

Figure 2: breadth of influence over different programming languages

Then, disregarding the language, we prepared a list of all the 10 ranks of each user (if the user has an influence over that language). Then, obtained the correlation between the first choice of the users and the next choices (i.e., the first two ranks he obtained over the 10 languages). And, we found that there is a moderate correlation between the first two choices (0.51 regarding number of forked projects and 0.42 regarding number of watched projects measure). For number of forked projects, there is also weak correlation between the first choice of each user and the next choices (until the 8th language) as well.

We elaborate on this intricate reasoning that the users' influence is spread over at least two different languages (not as a trend between two specific languages, but for each user between two desired languages). As a higher lever reasoning, we conclude that influential people, influence others no matter which programming language they are working focused in. In other words, as a character of the user, influence is spread over different programming languages.

We also argue that number of forked projects influence is more spread than number of watched projects measure (although people have higher number of watched projects than

Table 4: **Comparing influence metrics in Twitter and GitHub**

Twitter[9]			GitHub		
Metric	Top users in the list	Indication	Metric	Top users in the list	Indication
follow	public figures and news sources	popularity; people who get attention by one-to-one interactions; audiences are directly related to these influentials	*u.follow*	public but professional figures	popularity and fame in software engineering
mention	celebrities	name value	*u.watch*	a combination of public professional people and big companies or owners of libraries and frameworks	interest; both popularity and code usability
retweet	content aggregation services, news sources and businessmen	content value	*u.fork*	big companies, owners of practical libraries and frameworks	code usability (content value)

Table 5: Top 10 programming languages considered

Language	# of projects
JavaScript	2,681,228
Java	1,150,381
Ruby	1,279,018
Python	1,112,088
PHP	987,731
CSS	588,552
C	587,579
C++	523,479
Objective-C	471,241
C#	339,779

forked ones). This shows that number of forked projects influence measure, as the real engaging influence is not language dependent, and, is rather developer dependent.

5. CONCLUSIONS AND FUTURE WORK

We compared the insights on three influence metrics in GitHub; number of followers, watched projects and forked projects. We found that:

- People with many followers are not necessarily influential in terms of watched or forked projects. While number of followers is sign of popularity and fame in software engineering, number of forked projects influence measure is sign of code usability and content value. Comparing with Twitter, we resemble number of followers and number of forked projects in GitHub to follow and retweet respectively in Twitter. Number of watched projects is not equivalent with mention, but indicates both popularity and content value, with a focus on the later. We can interpret it as "interest" as well.

- As a characteristic of the developer, influence is spread over different languages. We found that the users' influence is spread over two or more different languages. But there is no specific trend for alignment between any two programming languages.

Both these results are mostly in parallel with the results of Cha et al. [9]. In Twitter, "having a million followers does not always mean much" [9]. Instead, they stated that having active audience who retweet or mention the user is more influential. In GitHub, we found many successful software engineering people with the most number of followers. Here, due to the nature of GitHub –which is technical and development based– being followed means something more than Twitter; The highly followed people are not politicians and athletes, but professional software engineering people who are more popular and esteemed by others (we validated this by both investigating the top 30 accounts and considering the weak correlation between *u.follow* and *u.fork*). So there is a little difference between the interpretation of "follow" based influence in GitHub and Twitter. Having said this, as the main conclusion, we found that like Twitter, number of followers is not the best way to identify the actively engaging influencers. There are other measures that measure real influence in terms of impact and influence (e.g., , number of forked projects and number of watched projects in GitHub).

Also Cha et al. mentioned that in Twitter, the influence of most influentials are obtained by continuous work over different topics. Interestingly, we observed that in Github, users are influential over 2 or more programming languages, even though being active in several programming languages (in GitHub) is much more difficult than being an active twitter in various topics (in Twitter).

As a future work, we envision considering more measures like starred projects, social influence (e.g., commenting) and bug report relations (e.g., bug reporter and fixer). In addition, there is a long road in inter-network analysis of the metrics to identify the similarities and differences between regular social networks and the SSNs.

6. REFERENCES

[1] I. Anger and C. Kittl. Measuring influence on twitter. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, i-KNOW '11, pages 31:1–31:4, New York, NY, USA, 2011. ACM.

[2] M. Aziz and M. Rafi. Identifying influential bloggers using blogs semantics. In *Proceedings of the 8th International Conference on Frontiers of Information Technology*, page 7. ACM, 2010.

[3] F. M. Bass. A new product growth for model consumer durables. *Manage. Sci.*, 50(12 Supplement):1825–1832, Dec. 2004.

[4] K. Blincoe, F. Harrison, and D. Damian. Ecosystems in github and a method for ecosyste m identification using reference coupling. In *Proceedings of the 12th Working Conference on Mining Software Repositories*, MSR '15, 2015.

[5] K. Blincoe, J. Sheoran, S. Goggins, E. Petakovic, and D. Damian. Understanding the popular users: Following, affiliation influence and leadership on github. *Information and Software Technology*, 70:30–39, 2016.

[6] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, and J. H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012.

[7] P. Bouros, D. Sacharidis, and N. Bikakis. Regionally influential users in location-aware social networks. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 501–504. ACM, 2014.

[8] D. Brown and N. Hayes. *Influencer Marketing: Who really influences your customers?* Routledge, 2008.

[9] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi. Measuring user influence in twitter: The million follower fallacy. *ICWSM*, 10(10-17):30, 2010.

[10] L. Dabbish, C. Stuart, J. Tsay, and J. Herbsleb. Social coding in github: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pages 1277–1286. ACM, 2012.

[11] E. Dubois and D. Gaffney. The multiple facets of influence identifying political influentials and opinion leaders on twitter. *American Behavioral Scientist*, 58(10):1260–1277, 2014.

[12] F. Harper. *Success in Programming: How to Gain Recognition, Power, and Influence Through Personal Branding*. Apress, 2015.

[13] G. Katsimpras, D. Vogiatzis, and G. Paliouras. Determining influential users with supervised random walks. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 787–792, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[14] E. Katz. The two-step flow of communication: An up-to-date report on an hypothesis. *Public opinion quarterly*, 21(1):61–78, 1957.

[15] E. Katz and P. F. Lazarsfeld. *Personal Influence, The part played by people in the flow of mass communications*. Transaction Publishers, 1955.

[16] E. Keller and J. Berry. *The influentials: One American in ten tells the other nine how to vote, where to eat, and what to buy*. Simon and Schuster, 2003.

[17] H. C. Kelman. Compliance, identification, and internalization: Three processes of attitude change. *Journal of conflict resolution*, pages 51–60, 1958.

[18] J. Marlow and L. Dabbish. Activity traces and signals in software developer recruitment and hiring. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 145–156. ACM, 2013.

[19] J. Marlow, L. Dabbish, and J. Herbsleb. Impression formation in online peer production: activity traces and personal profiles in github. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 117–128. ACM, 2013.

[20] A. Patil, G. Ghasemiesfeh, R. Ebrahimi, and J. Gao. Quantifying social influence in epinions. In *Social Computing (SocialCom), 2013 International Conference on*, pages 87–92. IEEE, 2013.

[21] R. Pham, L. Singer, O. Liskin, F. Figueira Filho, and K. Schneider. Creating a shared understanding of testing culture on a social coding site. In *Software Engineering (ICSE), 2013 35th International Conference on*, pages 112–121. IEEE, 2013.

[22] The GHTorrent Project. Mysql database dumps, data dump of 2015-04-01. "http://ghtorrent.org/downloads.html", Visited on 2014/08/20.

[23] E. M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[24] A. Sajedi Badashian, A. Esteki, A. Gholipour, A. Hindle, and E. Stroulia. Involvement, contribution and influence in github and stack overflow. In *Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, CASCON '14, pages 19–33, Riverton, NJ, USA, 2014. IBM Corp.

[25] A. Silva, S. Guimarães, W. Meira Jr, and M. Zaki. Profilerank: finding relevant content and influential users based on information diffusion. In *Proceedings of the 7th Workshop on Social Network Mining and Analysis*, page 2. ACM, 2013.

[26] M.-A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky. The (r) evolution of social media in software engineering. In *Proceedings of the on Future of Software Engineering*, pages 100–116. ACM, 2014.

[27] M.-A. Storey, C. Treude, A. van Deursen, and L.-T. Cheng. The impact of social media on software engineering practices and tools. In *Proceedings of the FSE/SDP workshop on Future of software engineering research*, pages 359–364. ACM, 2010.

[28] B. Sun and V. T. Ng. Identifying influential users by their postings in social networks. In *Proceedings of the 3rd International Workshop on Modeling Social Media*, MSM '12, pages 1–8, New York, NY, USA, 2012. ACM.

[29] C. Treude and M.-A. Storey. Awareness 2.0: staying aware of projects, developers and tasks using dashboards and feeds. In *Software Engineering, 2010 ACM/IEEE 32nd International Conference on*, volume 1, pages 365–374. IEEE, 2010.

[30] C. Zhou, P. Zhang, J. Guo, and L. Guo. An upper bound based greedy algorithm for mining top-k influential nodes in social networks. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 421–422. International World Wide Web Conferences Steering Committee, 2014.