Involvement, Contribution and Influence in Github and Stack Overflow

Ali Sajedi Badashian, Afsaneh Esteki, Ameneh Gholipour, Abram Hindle, Eleni Stroulia
Department of Computing Science, University of Alberta, Canada
{alisajedi, esteki, ameneh, abram.hindle, stroulia}@ualberta.ca

Abstract

Software developers are increasingly adopting social-media platforms to contribute to software development, learn and develop a reputation for themselves. *GitHub* supports version-controlled code sharing and social-networking functionalities and *Stack Overflow* is a social forum for question answering on programming topics. Motivated by the features' overlap of the two networks, we set out to mine and analyze and correlate the members' core contributions, editorial activities and influence in the two networks. We aim to better understand the similarities and differences of the members' contributions in the two platforms and their evolution over time. In this context, while studying the activities of different user groups, we conducted a three-step investigation of *GitHub* activity, *Stack Overflow* activity and inter-network activity over a five-year period. We report our findings on interesting membership and activity patterns within each platform and some relations between the two.

Keywords: Mining software repositories, Cross-network analysis, time series analysis, GitHub, StackOverflow

1 Introduction and Motivation

The software-engineering community is increasingly recognizing the significant evolution of programming practices, due to the rising adoption of social platforms for software development and knowledge exchange. Consequently, the scope of the mining-software-repositories research agenda is expanding to include, in addition to code repositories, social platforms like *Twitter* and *Stack overflow*. The general research question motivating these studies is understanding the nature of the individual developers' participation and contribution in these collaborative development.

Two among the most widely adopted and studied platforms are *GitHub*[1] and *Stack Overflow*[2]. *GitHub* is a collaborative development platform. Developers can participate in multiple projects to a different degree; for example, they may commit code and/or contribute to documentation, they may follow the activities of other developers, and they may watch the activities of projects of interest. *Stack Overflow* is a web site for asking and answering questions (Q&A) related to programming languages and tools. These two platforms serve different purposes: code sharing and collaborative development vs. information and knowledge exchange. At the same time, they both serve potentially the same community of developers for the same overall goal, *i.e.,* software development. Clearly, studying the behavior and activities of an individual developer in these two different platforms can help us glean valuable insights in the nature of their participation, contribution and influence in the global community.

To date, two studies have pursued such inter-network analysis, for two distinct purposes. Vasilescu et al. [9] studied how participation in

[1] https://github.com/

[2] http://stackoverflow.com/

the two networks impacts the developers' productivity in *GitHub*; they reported that it appears as though *Stack Overflow* participation reinforces *GitHub* productivity (albeit differently for novices and experts) and that active *GitHub* committers are also active answerers in *Stack Overflow*. Venkataramani et al. [10] developed a method for recommending *Stack Overflow* questions to developers, whose *GitHub* contributions constitute evidence of relevant expertise. In our work, we build and expand on this body of work by focusing on the question of "how different types of activity and productivity across the two platforms correlate" (essentially a variant of the first question above) but conducting a much broader investigation on a much larger data set. We are interested in examining the interplay between the actual *substantive contribution* of the developer to the platform's core objectives, the more general involvement of the developer with the community with "managerial" or "editorial" activities, and the *influence* and *recognition* that the developer has within and across the two platforms. This is the first step towards our long-term objective, which is to understand the factors that may positively influence productivity so that we can formulate and encourage new best practices for social software engineering.

In this paper, we first define three high-level indicators of developer's *development* contribution, general *management/editorial* activities, and *popularity* for each platform. Next, we study these indicators (and their correlations with each other) within and across the two platforms. Finally, focusing on the intersection of the two platforms' memberships, we study the their adoption patterns and the relative intensity of the users' activities on the two platforms. Our findings indicate that, when focusing on substantive development, the users' contributions in the two networks correspond only weakly. We find similar results to Vasilescu, but with looser connectivity, likely due to two important differences in our studies. First, our study examines a much larger data set; second, we adopt a different analysis methodology: while they examined smaller preselected user sub-groups, we examine (a) the complete set of developers who belong to both platforms, and (b) user groups with similar activity intensity in both platforms. More generally, examining the users' adoption patterns and involvement and contributions in the two platforms over time, we discovered several interesting similarities, as well as noticeable differences, implying the need for more work in this area.

The rest of this paper is organized as follows. Section 2 places our work in the context of previous work in the area. Section 3 describes our data set and Section 4 details our analysis methodology. Section 5 reviews our results and Section 6 discusses our inferences based on them. Finally, Section 7 summarizes our conclusions and outlines some avenues for future work.

2 Related Work

The amount of information available in software repositories is increasing on a daily basis, as more users adopt these platforms and engage in their respective communities. This increasingly rich data set has been the subject of substantial analysis through a variety of statistical and social (graph-based) analysis methods.

Social connectedness in developer communities: Thung et al. [8] analyzed characteristics of developer-developer and project-project relationship graphs in *GitHub*. They identified influential developers and projects using the pageRank algorithm, and they report that project networks are more interconnected than developer networks. In other words, although social coding enables substantial collaborations among developers, software development networks are fundamentally different from other social networks. In software-development networks, individuals are connected mostly through and around code (*i.e.,* projects, questions, etc.) while in typical social networks, connections are created directly between users (*i.e.,* "friending", "following" etc.).

Influential developers: Several studies have focused on influential people and how to identify them. Watts and Dodds [11], for example, believe that the influentials can be opinion leaders in a large group, and they can be targeted, with a reasonable cost, to persuade others to do some business-related actions. Likewise, in the context of software development, according to Lee et al. [6], there are some groups of extremely well-connected "rock-stars" developers [7]), whose activities influence others: other users use the activity of these rock-star users as guides to their projects. More recent studies correlate the susceptibility of a society to a spreading trend to two main factors [1]: the readiness of the society and the inter-personal relationships among users. New marketing strategies like collaborative filtering are, to a degree, the result of this modern view toward influence. Influential people, in the domain of social networks, were understood as simply those with more followers. However, more recent research by Cha et al. [1] on Twitter reveals that there are other important factors relevant to influence. They studied retweets, mentions and indegree (number of followers) as indicators of content, name value and popularity of a user respectively and reported that while the most followed users are public figures, the most mentioned users are mainly celebrities. They believed that indegree reveals only small part of one user's influence and there are some other influence indicators like *mentions* [1]. In this paper, we computed and compared these indicators in *GitHub*; our comparison revealed similar results for indegree and mentions, confirming that mentioning a user is different from following him/her.

Contribution across platforms: There have been few attempts to correlate activities of developers across *GitHub* and *Stack Overflow*. For example, Venkataramani et al. [10] compared code commits in *GitHub* –as an evidence of expertise– with the expertise needed to answer newly posted *Stack Overflow* questions. They developed a recommender system for the *Stack Overflow* questions that suggest a developer to answer the question. Their methodology follows that of Ghosh et al. [2], but spans across multiple networks. Ghosh et al. mined Twitter lists of millions of people to identify the expertise of the listed users. Twitter lists are like tags or contact groups in a cell phone and are used for individuals to categorize their connections. Ghosh et al. calculated the "value" of a user based on the list names in which he/she is included in. Finally, the only other study that tried to find the shared behavior of users in *GitHub* and *Stack Overflow* was done by Vasilescu et al. [9], who considered the overall activities of developers over the two platforms and analyzed the distribution of work units over time, focusing on functional interactions between commit and question/answer activities. They reported that active *GitHub* committers take the role of "teachers", contributing more answers than questions.

3 Data Preparation

In this work, we use the *GitHub* dataset released on October 2013 as a MySQL database dump in GHTorrent [4]. With a total size of 15.3 GB, it contains information about 2,437,234 users and their activities from December 2007 to October 2013, represented in 22 sql tables, which include data about users, commits, comments on commits, issues, pull requests, follower-following relations, and so on. With respect to *Stack Overflow*, we use the 20GB XML *Stack Overflow* dataset, released on September 2013, that contains information about 2,332,403 registered users from August 2008 until September 2013. For our comparison of the two networks, we consider the data for all users within the five-year period starting from September 1, 2008 and ending at August 31, 2013 for both networks. Note that this data set is substantially larger than the one of Vasilescu et al., who studied 50(10) months of *Stack Overflow* (*GitHub*) activity.

In order to compare activities of users in *GitHub* and *Stack Overflow*, one must identify those users who use both systems. The process of finding common contributors is called *identity merging* and is a big challenge since users may use different aliases in different repositories. We use Vasilescu's approach [9] for intersecting the two datasets, relying on email addresses. In the *GitHub* dataset, email addresses are public and available, while in the *Stack Overflow* dataset only the MD5 hashes of emails exist. Therefore we link a user from *GitHub* to one in *Stack Overflow* only if their computed MD5 hashes are identical. Using this approach 261,841 common users were identified, a set representing approximately 10.7% of *GitHub* users, and 11.2% of *Stack Overflow* users.

Unlike *Stack Overflow*, *GitHub* does not validate users' email addresses. This enables users to register with different logins or names, but the same email address. For example *John Smith* might register by *(John Smith, johnsmith@gmail.com)*, and *(JohnS, johnsmith@gmail.com)*. Such duplicate users are merged and their activities in both networks are aggregated. In total, 6466 duplicates out of 261,841 common users were found; in the end, after merging duplicates, the total number of users with a presence in both networks is 255,375.

We wrote Java programs for extracting the data out of data dumps, intersecting the metrics between the two networks, merging duplicate users and finding mentions to the user names. These programs are computationally intensive. So we parallelized some of them using MapReduce [5] (in a four node virtualized cluster in Apache Hadoop, each node with 4-core processors, 8 GB memory, 100 GB hard drive, and working on Ubuntu 12.04 OS). Importing the data sets and running the Java programs consumed a total time of more than 30 hours. We also used R 3.0.2 and SPSS 17.0 for our statistical analysis.

4 Activity Indicators

To analyse the developers practices in *GitHub* and *Stack Overflow* in a consistent manner, we first extracted a number of basic metrics, specific to *GitHub* and *Stack Overflow* activities. Next, these basic metrics were combined to define three higher-order indicators representing substantive contribution, managerial/editorial activity, and influence in each of the two platforms. The corresponding sets of these three higher-order metrics enable us to establish a "level playing field" over the two platforms, abstracting away the particular details of the activities they support.

4.1 *GitHub* Activity Metrics

The basic activity indicators for each developer on *GitHub* are shown below.
Commits: Number of commits made by the developer.
PullReqs: Number of times the developer notified others about his changes so that they can pull them if they want.
PullReqsHandled: Number of times the developer opened or closed pull requests; the person who acts on a pull request may be different from the one who issued this pull request.
ProjectsWatched: Number of projects the developer watches. Watching a project lets the user be notified about new commits, pull requests and issues in the project repository.
IssueComments: Number of comments the developer made on issues. Issues are, in effect, notifications between team members for tracking bugs or tasks.
IssuesReported: Number of issues the developer reported.
IssuesHandled: Number of times the developer pushed, forked or commented on a previously reported issue.
Followers: Number of people who follow the developer.
Mentions: Number of times the developer's name is mentioned in comments (*i.e.,* commit comments, pull request comments and issue comments). This parameter is counted by looking for "@username" patterns. However, skimming through the list of user names in *GitHub* reveals the fact that there are some misleading user names such as *@have*, *@Github* and *@c++*. On the other hand, the chance of using these words by a user in his/her comments is very high. For example we observed that *@c++* is used in several comments in order to refer to something about c++ language. To reduce the number of false positives, the user names that match English stop words, Programming languages and Reserved words in them are filtered out in the process of recognizing mentions.

Starting with the above basic metrics, we defined three higher-level indicators. The *development (DEV)*[3] indicator is defined as the sum of log(*Commits*), *PullReqs* and *PullReqsHandled* by the developer. These three types of activities are combined together since they all potentially bring about changes to the project repository. The commit activity is log-scaled because logically, a pull request is the result of several commits. This decision was validated through our inspection of the data, which showed that the average values for commit is about ten times greater than the two other metrics. The *management (MAN)* indicator captures the non-core contributions of the developer to the project, defined as sum of *IssuesReported*, *IssueComments*, *IssuesHandled* and *ProjectsWatched*. Finally, the *popularity (POP)* indicator aggregates the number of the developer's *Followers* and *Mentions* to his/her name, representing the overall recognition the developer enjoys in the community.

Note that for some of the derived metrics, we used log-scaling of one or more components (*i.e.,* simple metrics) for de-emphasizing on some simple metrics as well as normalizing their data. The distribution of data in these cases were log-normal in which the logarithm of the values is almost normal.

[3] Log-scaling is used whenever a metric is less meaningful than other metrics with which it is combined, or when its range is much higher than the other metrics.

4.2 *Stack Overflow* Activity Metrics

Similarly to our work with *GitHub*, we first established a number of basic *Stack Overflow* metrics for each *Stack Overflow* member, as described below.
Questions: Number of questions asked by the user.
Answers: Number of answers provided by the user.
UpVotes: Number of UpVotes cast by the user to posts of other users.
DownVotes: Number of DownVotes cast by the user to posts of other users.
QuestionsViewed: Number of times the user's questions were viewed by others.
Favorites: Number of times the user has received favorite (interesting) tags for his/her questions.
ProfileViews: Number of times the user's profile was visited by others.
PostScores: The aggregated score given to all posts of a user by other users.

Next, we computed a number of derived indicators, conceptually parallel to the ones we computed in *GitHub*. The *development (DEV)* indicator is computed as the sum of the user's *Questions* and *Answers*. The intuition is that active users usually ask more *Questions* and cast more *Answers*. The *Stack Overflow management (MAN)* indicator is defined as the sum of the *UpVotes* and *DownVotes* a user has cast on other posts. The *popularity (POP)*[3] indicator in *Stack Overflow* is defined as the number of *Favorites* plus (the logarithm of) the sum of *ProfileViews* and *PostScores* a user has received and log(*QuestionsViewed*).

Again, we used log-scaling for de-emphasizing as well as normalizing the data after looking for the values and distributions of the simple metrics.

5 Mining, Analysis and Findings

In this section, the findings of our analyses of each network are reported separately and then the cross-network analysis is discussed. Unless specified directly, all the correlations are Spearman correlation and the p-values are less than 0.01 (99% significance level).

To analyze the activity patterns of the users in the two networks, each user was ranked based on their *development* metric in *GitHub* and *Stack Overflow* separately. Then, in each network, we categorized the users to three groups: (a) the top 1000 *super-active* users; (b) the next 9000 *active* users; and (c) the rest of the users who are more *typical*. Orthogonal to these categories, we distinguish the *consistent* users who have been active (i.e., committing assets on *GitHub* and asking and answering questions on *Stack Overflow*) for at least 10 out of 60 months and exhibit a low variance of number of commits, questions/answers, *i.e.,* less than 25 over 60 months. There are 7485 and 20504 consistent users in *GitHub* and *Stack Overflow*respectively.

The analysis of the behavior of individual users as well as these four user groups is discussed in this section.

5.1 *GitHub* Findings

In order to develop some initial intuitions regarding the types and intensity of activity of the *GitHub* developers, the *Spearman* correlation between all pairs of (basic and derived) metrics was calculated. The values for correlation between derived metrics and all the others are shown in Table 1. A moderate correlation between Followers and all other simple activity metrics was discovered (*e.g.,* 0.4 with ProjectsWatched, 0.39 with PullReqsHandled and 0.38 with PullReqs. The other values are less than 0.35 and are not shown here due to space limitation). Intuitively, this correlation implies that as developers become increasingly engaged with the *GitHub* platform, *i.e.,* joining more projects, committing more code, and contributing to more issues, they accrue more followers. Interestingly, the correlation is weakest for Mentions, which leads us to infer that even though developers may be followed by many other community members they are not necessarily mentioned in the community discussions. This finding is consistent with similar findings in the Twitter community [1], and this phenomenon is discussed in Section 6.

As shown in Table 1, more popular users are engaged more in the commits, projects, issues and even comments. *Management*, in addition to its constituent metrics, correlates with *Commits*, *PullReqs* and *PullReqsHandled*. This finding indicates that the people engaged in monitoring activities, have probably more important commits in the projects than other users because they show a moderate correlation with *Commits* and higher correlations with *PullReqs* and *PullReqsHandled*. These two later activities indicate to other community members that the user in question has important updates for them to pull.

Table 1: *Spearman* correlation coefficient between *GitHub*'s metrics

	Followers	Mentions	ProjectsWatched	Commits	PullReqs	PullReqsHandled	IssueComments	IssuesHandled	IssuesReported	GH-Dev	GH-Man	GH-Pop
GH-Dev	0.53	0.17	0.47	0.86	0.77	0.79	0.60	0.56	0.59	1.00		
GH-Man	0.59	0.16	0.85	0.47	0.60	0.61	0.79	0.75	0.78	0.60	1.00	
GH-Pop	1.00	0.17	0.56	0.50	0.46	0.49	0.50	0.48	0.46	0.53	0.59	1.00

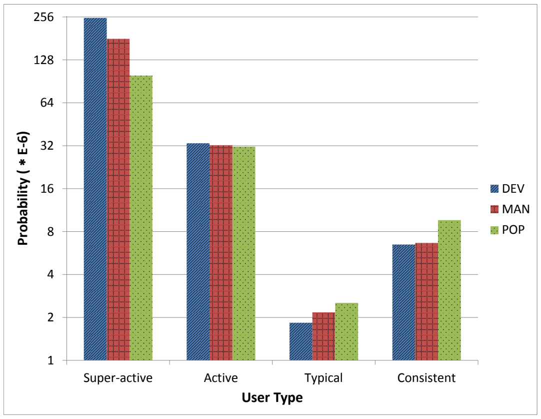


Figure 1: Density of derived metrics in *GitHub* for four *GitHub* user groups

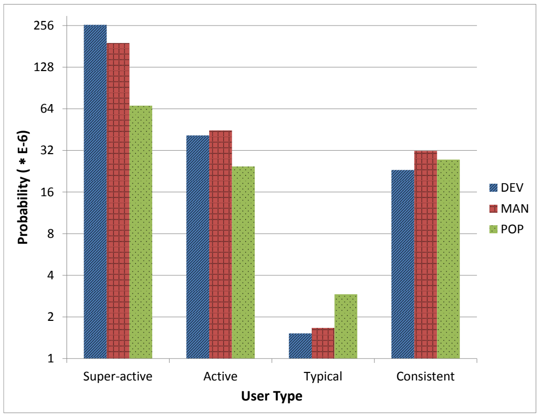


Figure 2: Density of derived metrics in *Stack Overflow* for four *Stack Overflow* user groups

Development is also correlated to the metrics related to issues, implying that active developers are relatively active in many activities, beyond just committing. The correlations also indicate relatively a strong relationship between the three derived metrics, which indicates that the more popular users contribute more to code (*development*) and also perform more monitoring tasks (*management*). Also the *management* and *development* activities of the users are correlated, which indicates that those who monitor issues and watch projects are also actively engaged in coding.

In the next step, we comparatively studied the behaviors of three distinct groups of *GitHub* developers: *Super-active*, *Active* and *Typical* users as defined before. In effect, this analysis was conceived to study how *"the behaviour of more active GitHub developers differs from that of the less active ones"*. Results of this analysis are shown in Figure 1. In this figure, the aggregated value of each one of the three *GitHub* metrics for all users is assumed to be one. As this figure shows, as an example, the average *development* value for each one of the super-active users is 2.52E-4 that is the density of *development* of a single super-active user. The log-scaled y-axis shows that each sample user is responsible for what fraction of contribution in the three derived metrics. The figure reflects the correlation results in the previous step. While the correlation results say that all the three derived metrics almost change together, Figure 1 indicates that for each user group (*i.e.,* super-active, active or typical users) the three derived metrics are almost in the same range. The users become popular while they write more code and monitor more projects. However, the lower levels of popularity can be attained with a little effort while achieving higher levels requires much more effort, and more specifically substantial development activity. Popularity, however, is not gained through *development* alone. There are other factors like their communications, commenting behavior or consistency of the activities of a user, as shown by the increased popularity of consistent users with respect to their level of activity.

5.2 *Stack Overflow* Findings

Similarly, the analysis of the *Stack Overflow* dataset was conducted in two steps. First, the correlation between all the (basic and derived) metrics was calculated for all *Stack Overflow* users. The values for correlation between derived metrics and all the others are shown in Table 2. Results show that almost all the metrics are highly correlated with each other. *Popularity*, in addition to its four constituent metrics, strongly correlates with *DownVotes*, *UpVotes*, *Questions* and *Answers* indicating that popular users are highly engaged in these activities, with UpVotes having higher correlation (to *Popularity*) than DownVotes, and Questions higher than Answers. While voting in general is an important contribution and results in the user gaining popularity, positive attitude (as evidenced in UpVotes) is more effective than the negative ones. Also while both question and answer posting are related to popularity, quite surprisingly, question asking is more important than answering. In other words, while popular users have a lot of questions and answers, popularity correlates more highly with questions than answers.

Development is strongly correlated with all the simple metrics indicating that active users are engaged in almost all activities instead of just asking / answering.

The correlations also indicate strong relations between the three derived metrics (stronger than the similar relations in *GitHub*). It shows that the more popular users have more posts (*development*) and also perform more monitoring tasks (*management*). The very strong correlation between *popularity* and *development* confirms our findings on posting and its effect on popularity (with emphasize on questions, as mentioned earlier). Also the *management* and *development* activities of the users are correlated implying that those who monitor posts are also actively engaged in the posts as well.

Next, the activity of three user groups of *Stack Overflow* developers are studied to compare the behavior of active *Stack Overflow* developers against others. The distribution of values of each one of the three *Stack Overflow* derived metrics for users of different groups are shown in Figure 2. Similar to *GitHub*, all the derived metrics change together, so we can infer that people get popular when they post their questions/answers, but achieving high levels of popularity needs much more posting efforts. Users can also gain popularity through other activities. However, we can see a 0.91 correlation between *popularity* and *development* as the strongest correlation. This emphasizes on the importance of posting on popularity of a user.

5.3 *GitHub* and *Stack Overflow*

In this section, we discuss our analysis of the two datasets cross-referenced, with the intent to examine if the types and levels of developer activity across the two networks correlate. Intuitively, one might assume that users active in one platform are also active in the other, as implied by Vasilescu's work [9]. This intuition is empirically examined in this section by analyzing the correlation between metrics in the two networks followed by linear-regression models.

To investigate this intuition, the correlations of all pairwise combinations of all basic metrics were computed. Unlike our original expectation, the correlation values indicate only weak relations. For example, the correlation between *Commits* in *GitHub* and *Questions* and *Answers* in *Stack Overflow* is 0.07 and 0.17 respectively. The greatest values belong to *IssueComments*, *IssuesHandled* and *IssuesReported*. These three metrics were correlated with *UpVotes*, *ProfileViews*, *Answers* and *post scores* with values between 0.24 to 0.28. Other values are less than 0.25 (due to space limitations and weakness of all the values, we suffice the most important values and skip the detailed values of other pairwise correlations). Compared with Vasilescu et al. [9], we found weaker relationship between committing and answering. Thus, we turned our attention to the higher-order metrics to look for potentially subtler relationships. The intuition is that even though the basic metrics do not correlate, it is still possible that combinations of these basic metrics (with a higher level logic) may actually correlate.

We paired all the higher-order metrics and evaluated the correlations. All the pairwise correlations are between 0.15 and 0.26, shown in Table 3. These results indicate that the corresponding *development*, *management* and *popularity* met-

Table 2: *Spearman* correlation coefficient between *Stack Overflow*'s metrics

	DownVotes	UpVotes	ProfileViews	Questions	Answers	Favorites	QuestionsViewed	PostScores	SO-DEV	SO-MAN	SO-POP
SO-Dev	0.57	0.82	0.87	0.78	0.87	0.63	0.77	0.88	1.00		
SO-Man	0.64	1.00	0.79	0.63	0.79	0.62	0.63	0.84	0.82	1.00	
SO-Pop	0.55	0.79	0.91	0.83	0.72	0.75	0.85	0.88	0.91	0.79	1.00

rics across the two networks are only weakly correlated. The best result is for *management* that is 0.26 indicating that people with substantial levels of non core-development, i.e., monitoring, activities in *GitHub* probably also monitor *Stack Overflow*. Similarly, there exists a slightly weaker correlation between *development* and *popularity*. We repeated the experiment after log-scaling the basic metrics and achieved almost the same results (all correlations smaller than 0.3) leading to the same weak relationships.

To tease out multi-variable relationships, linear regression models were produced between all the (basic and higher-order) metrics of one network as independent variables and each of the higher-order metrics of the other network as the dependent one. Each time running the ANOVA model, we excluded ineffective variables and re-executed the model. For all the cases the R-square was pretty low (usually between 0.001 and 0.06 and in rare cases up to 0.09); so no regression model can be fit between subsets of independent variables in *Stack Overflow* and higher-order metrics of *GitHub*, and vice versa. Although in a few cases it was observed a T-value >2 and sig <0.01, but due to low R-square (as mentioned above), the model can't fit. In other words, the T-value and significance in a few of the above experiments indicate that there are relations, but the very low R-square means that all the independent variables can change a small percentage of the dependent variable from the other network (up to 9% in the best case here) and the model doesn't fit generally. Again we repeated running the models with log-scaled metrics, yet no better results were obtained. These results endorse previous findings on weak connectivity between *Stack Overflow* and *GitHub* activities and that the activity in one network cannot predict the other in general.

Table 3: *Spearman* correlation coefficient between *Stack Overflow* and *GitHub* derived metrics

SO GH	SO-DEV	SO-MAN	SO-POP
GH-DEV	0.19	0.21	0.18
GH-MAN	0.22	0.26	0.23
GH-POP	0.15	0.18	0.16

These results indicate that while one cannot neglect the similarity between the users behaviours in *GitHub* and *Stack Overflow*, this similarity is not at all strong. In other words, *Stack Overflow* and *GitHub* are not completely unrelated, but the relations between them are weak. We will compare these findings with the results of Vasilescu et al. [9], in Section 6.

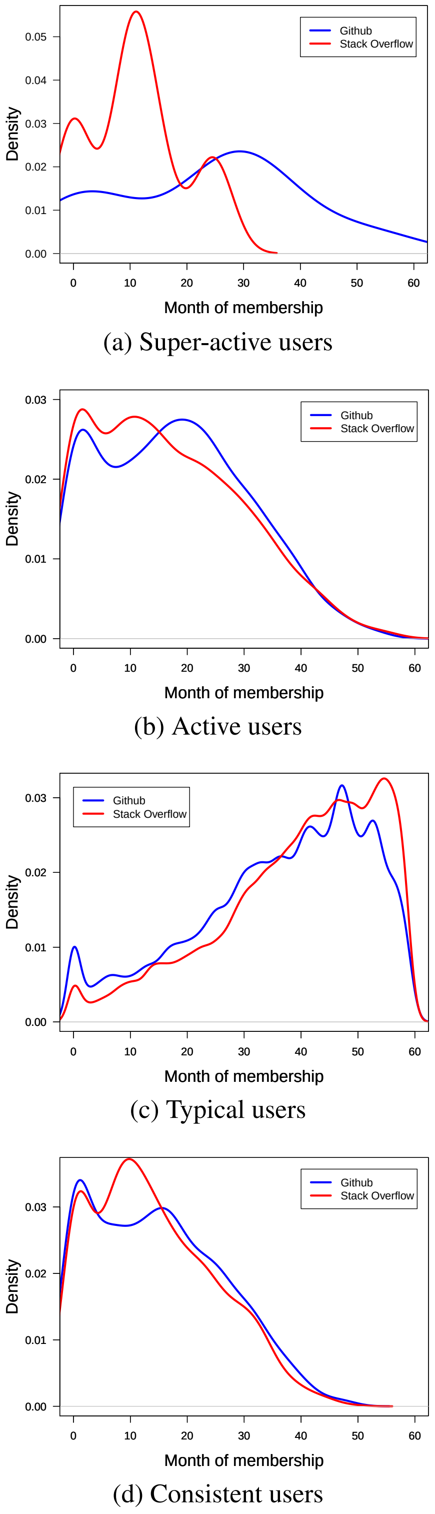
5.4 Findings on Similarly Active User Groups

To address the differences and similarities between *Stack Overflow* and *GitHub*, the behaviors of users with similar activity levels across the two networks are compared, *i.e.,* considered super-active users in *GitHub* who are also super-active users in *Stack Overflow*. We use this filtering to make the comparisons more precise between two equivalent user groups. Interestingly, there are only 16 *super-active users* in both networks; only 1.6% of super-active users in *GitHub* (out of 1000 super-actives in *GitHub*) are also super-active in *Stack Overflow*. Note that we want to preserve harmony with the definitions in previous sections. Although the overlap between super-active users of the two networks is not a large set, it is still an indicator of the very active users with the similar role in both networks and this is quite interesting for our purpose. We also do not want to enlarge the selection window to include high number of users with low level of activity. There are 852 users who are *active* in both *GitHub* and *Stack Overflow*, indicating that around 9.5% of active users in *GitHub* (out of 9000 actives in *GitHub*) are also active in *Stack Overflow*. There are 234,955 *typical* users in both networks: in this group and up to half of them are "idle", *i.e.,* people with none or only one commit, question, or answer. Finally, there are 1167 *consistent users* in both networks, indicating that around 15.6% of consistent users in *GitHub* are also consistent in *Stack Overflow*.Note that members of the other three groups who also satisfy the consistency criterion is also a member of this group. In fact, this group of users is considered in order to verify the analyses for the other groups; due to lack of activity for many of the typical users or small number of users in the super-active users, one may say that the results cannot be generalized for all users. So we considered this fourth group as a verification, sanity or endorsement to the measurements for the other user groups.

Note that there are about 20,000 users (less than 8% of all users) that lie outside the above categories, because they are ranked in one of these categories in one network but not the other. These users are not considered in the comparison of different user groups in this section, because we are interested in focusing on similar users as much as possible. Also note that the low rate of common super-active, active and consistent users (1.6%, 9.5% and 15.6% respectively) is evidence to non-zero but weak connectivity.

5.4.1 Membership

We first compared the length of each users' membership in *GitHub* and *Stack Overflow*, in terms of months (a number between 1 and 60). Having two numbers for each user, corresponding to the length of the user's membership in *GitHub* and *Stack Overflow* correspondingly, a Pearson-correlation test shows a 0.45 coefficient between all common users, indicating a moderate relationship between the two membership dates. The same correlation estimation was repeated between each of the four user groups separately; we found that for the super-active, active, typical and consistent users, the correlations are 0.28, 0.49, 0.44 and 0.41 respectively. The three later values are significant at the 0.01 level, but for the super-active users the results are not significant (p-value=0.29, that is greater than



(a) Super-active users

(b) Active users

(c) Typical users

(d) Consistent users

Figure 3: Distribution of "Month of membership" for different user groups

0.05; and the reason is sample size that is not enough).

For an alternative view of the users' adoption of the two platforms, we considered the initial-date-of-membership densities for the members of each user group in Figure 3. For all types of users, except for the super-active ones, the patterns are quite similar. While the two patterns for super-actives are somehow similar, the majority of super-active users started their activities early (mostly around month 10) in *Stack Overflow* and *GitHub* super-active users joined later (mostly around month 30). Unlike the typical users, most of the active and consistent users, however, joined the two networks in the first 30 months. In other words, the typical users are late adopters, starting mostly in the second half of the network's life.

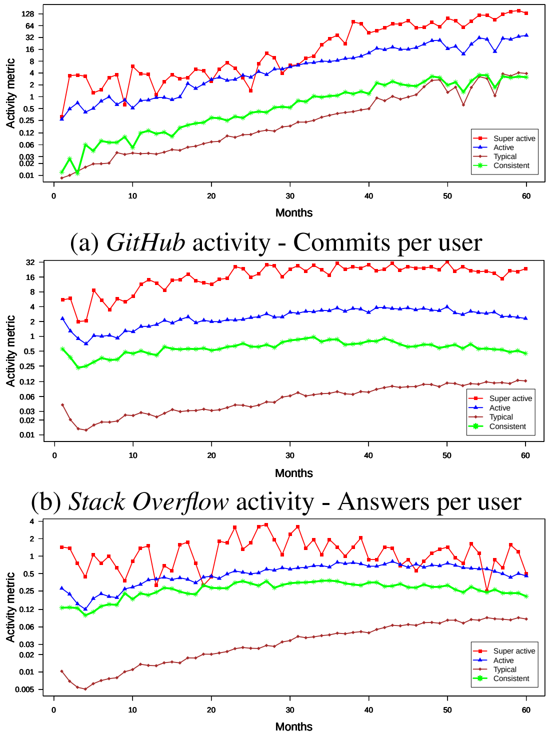
Finally, we checked the average values for the month of membership for the different user groups as well as all users. Except for the super-active users, the averages were for *Stack Overflow* and *GitHub* were in the same range (with 10% difference). The average difference between time of membership in *Stack Overflow* and *GitHub* was 11.8.

Combining the above-mentioned three arguments (moderate correlation between membership times in *Stack Overflow* and *GitHub* , similarity of membership patterns and the average comparison) that has been done for different user groups as well as all users, we conclude that users may have made the decision to join the two platforms at (around) the same time.

5.4.2 Activity Over Time

To compare the activity of the users in the two networks over time, we focused on the average work done by developers in each of the four groups of interest, concentrating on *Commits* in *GitHub* versus *Questions* and *Answers* in *Stack Overflow*, as the three more important indices of core development activity. The average activity of each user in each of the user groups over the 60 months lifetime is shown in Figure 4.

General Activity level: As it is shown in the Figure, for both *GitHub* and *Stack Overflow*, the level of activity from typical to active and super-active users increases by an order of magnitude (note that the y-axis is log-scaled). The activity levels of consistent users lie between those of typical and active users, with several times more answers than questions. Comparing the level of answers and questions in *Stack Overflow*, the difference between the level of questions of super-active, active and consistent users is much higher than the corresponding difference of their question levels. This means that more active users (that are actually more active committers, as indicated in the figure) are engaged more in answering than asking questions. This is mostly consistent (and in-part contradicting) with some of Vasilescu et al.'s findings [9]. They found that highly active committers provide more answers and ask fewer questions. Our results indicate that these users both ask and answer more questions than others. However, they provide answers several times –up to tens of times– more than they ask questions.



(a) *GitHub* activity - Commits per user

(b) *Stack Overflow* activity - Answers per user

(c) *Stack Overflow* activity - Questions per user

Figure 4: Time series analysis of commits, questions and answers of different groups during 60 months from September 2008 to August 2013.

The data depicted in Figure 4 reflects the activity of 235,823 users participating to *GitHub* and *Stack Overflow*, of which 16 are super-active, 852 are active and 234,955 are typical ones (the 1,167 consistent users are filtered out of the total shared data and can contain users from each of these groups). However, many of the typical users have only a very low level of activity: up to half of them are idle, *i.e.,* people with zero or only one commit, question, or answer. This is why we identify a set of "consistent" users, who have contributed at least one or more commits, questions and answers per month over 10 months, and exhibit a low variance of number of commits, questions and answers, *i.e.,* less than 25 over 60 months. The general timeline of the consistent users is somewhere in between the active and typical users. Also their pattern shows similarities to the other user groups. For the following analyses, we consider the behavior of consistent users as a sanity check for the other measurements.

Activity Growth Rate: According to Figure 4, the activity rate of users in *GitHub* grows dramatically over time, while in *Stack Overflow* the growth rate is almost linear. In fact, during the first months, the absolute levels of question asking/answering activity were more than that of commits, even for the typical users. However, after a couple of years or so (the time depends on the user type), the committing activity reaches and even exceeds the *Stack Overflow* asking and answering levels. For example, we inspect the consistent users as the most reliable users (with respect to steady activity and little change). Consider the range of activities for the first 10 months and compare it with the last 10 months (for *commits*, *questions* and *answers*):

- Commit: "0.01 to 0.1" → "1.5 to 4"
- Answer: "0.2 to 0.6" → "0.5 to 1"
- Question: "0.1 to 0.25" → "0.2 to 0.3"

At first (left side of the arrows), the level of activity in *Stack Overflow* is more than that of *GitHub*, but during the last months (values in the right side), the *GitHub* activity surpasses that of *Stack Overflow*. Note that the y-axis is log-scaled, so the linear patterns for *GitHub* are actually exponential. In the last months of the 60 month period, the *GitHub* activity level becomes 3 to 100 times higher than the question asking/answering levels of *Stack Overflow* –depending on the user type. Note that, in most cases, even the activity level in *Stack Overflow* decreases –e.g., answering for consistent users or asking for super-active and active users.

This rather dramatic difference is likely due to the rather unique status of *GitHub* in the field, which makes it compelling to developers. *Stack Overflow*, on the other hand, has many competitor platforms, such as for example, Yahoo! answers and Wikipedia. Also it may be because of the potential saturation of information that may be reached in *Stack Overflow*: once answered, a question can be revisited many times by developers working in different projects, but there is always new requirements during the development of new projects.

Fluctuations: The activity of super-active users exhibits more fluctuations (this is due to the limited number of users in this group). Excepting these users, the activity levels in *Stack Overflow* are rather consistent while *GitHub* activities exhibit fluctuations over time, likely due to the deadlines that projects face and to *"time of year"* phenomena. For example, there are some fluctuations (for all user groups) in the first year in *GitHub*. And a dramatic dip occurs in 52nd month (December 2012). Also a similar pattern can be observed exactly one year before (around month 40). The *GitHub* activity fluctuations may be due to holidays and the launching and closing of large-scale projects.

5.4.3 Overall Activity

Next, we calculated the percentage of a single user's activity (in the whole 60 months duration) over the total activity of all users in the platform. Figure 5 shows the results. Consistent with the findings of the Figure 4 in the previous Section, this Figure implies that active and super-active users answer more questions than they ask (approximately four and two times). The number of commits remain in the middle between the number of questions they ask and the number of answers they provide. Finally, all three activities decrease an order of magnitude as we move from super-active to active to typical users. Typical users only answer a few questions and ask more than twice as much. For consistent users, the rate is almost the same for questions and answers.

5.4.4 Activity Change Over Time

We also analyzed the variance of users' activity over time, calculating the average activity variance for the users of each user group over 60 months. Typical users show the least variance and the super-active ones the most. However, for most typical users, their variance is near (or equal) zero due to their low (or zero) level of activity. So this absolute

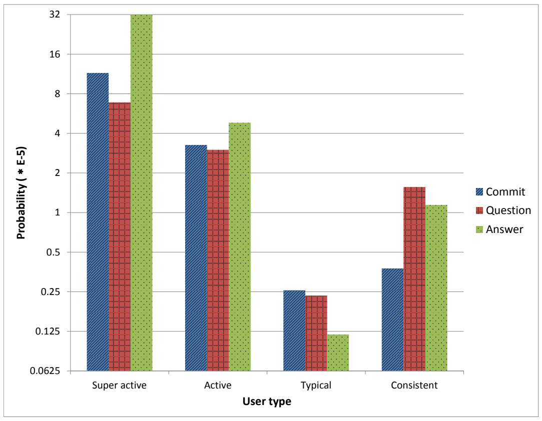


Figure 5: Density of simple activity metrics for users of four groups

Table 4: The percentage of activity change in *GitHub* and *Stack Overflow* with respect to the previous month

GitHub activity	*Stack Overflow* activity	%
increasing (↑)	increasing (↑)	4
decreasing (↓)	decreasing (↓)	3
increasing (↑)	decreasing (↓)	3
decreasing (↓)	increasing (↑)	3
no change (–)	increasing (↑)	19
no change (–)	decreasing (↓)	19
increasing (↑)	no change (–)	26
decreasing (↓)	no change (–)	23

variance value does not actually indicate "consistency" of this user group. However, with respect to the level of activity, the active users are the most consistent. Furthermore, a higher percentage of the active users are members of the consistent group (note that the three other user groups can overlap with consistent user group). Thus, we infer that active users are the most consistent ones. In other words, the active user group, exhibit a higher level of activity as well as consistent behavior.

Finally, we examined whether the levels of activity of a user in the two networks change (increase / decrease) together. The level of activity (commits in *GitHub* and question asking-and-answering in *Stack Overflow*) of each *user-month* was compared against their activity in the previous month. So for each user, we have 59 "activity change" levels that may be "increasing (↑) / decreasing (↓) / no change (–)", if the user's activity is "more than / less than / equal to" his/her activity in the previous month. Considering Table 4, only about 7% (first two rows) of common users exhibit the same change patterns in *GitHub* and *Stack Overflow* in either of 59 months (with respect to the previous month). Confirming our previous results, this implies that one cannot predict a user's activity in *GitHub* based on his/her activity on *Stack Overflow* and vice versa. Note that users whose activities have not changed in either of the networks are not considered here. These users had in almost all cases zero commits, questions and answers; hence, they are excluded from this analysis.

6 Discussion

We found that popular *GitHub* users, with large numbers of followers, are not mentioned frequently by name. This is similar to the findings reported by Cha et al. [1] who reported that popular users in Twitter, who have high numbers of followers, are not necessarily mentioned frequently by name, or retweeted. It would appear that there are two fundamentally different degrees of "recognition": the first represents the community's initial assessment of an individual as "interesting" while the second reflects the actual recognition of the individual by name. This intuition is also supported by our finding that all three higher-order metrics, i.e., (*development*, *management* and *popularity*), change together. So the users get popular while they write more code and monitor more projects. The lower levels of popularity can be gained with a little effort while achieving the higher levels needs much more effort, especially very high levels of development activity. In *Stack Overflow*, in terms of popularity and profile view, having good questions is more important than answers. While good –*e.g.,* straight or well-written– answers correlate with a users' popularity, good questions –*e.g.,* on-demand or well-written– play more important role in gaining popularity. Furthermore, casting up/down votes is important, but positive attitude is a more effective factor for popularity –because of the evident higher correlation of *popularity* with *UpVotes* rather than *DownVotes*.

Regarding the interdependencies between the users' activities across the two networks, we found both similarities and differences. The most important similarity is the adoption pattern: most users who participate in both platforms joined the two networks approximately at the same time. There is, however, a fundamental difference: with the weak correlation between the basic and higher-order activity metrics of the two networks, we can conclude that user activity in one network is not a very strong predictor of activity in the other network. This finding contradicts the results of Vasilescu et al. [9], who mentioned stronger relations between the two networks. There are several possible explanations for this difference. First, our data set is more than twice as big than the data set used in their study and covers the same 60-month period was considered for both networks, while their data set contains information for about 50 months of *Stack Overflow* activity and only 10 months of *GitHub* activity. Furthermore, our analyses consider the activities of users with multiple *GitHub* accounts; Vasilescu did not report merging the activities of such users as our study did. Finally, in order to validate these findings, we downloaded the Vasilescu data set and applied our analyses on it. Surprisingly, we obtained similar results as in our study (*i.e.,* all correlations were weak and no regression model fit). This implies that there is a fundamental difference in the nature of "dependencies" explored by their analyses and ours: they have used rank-based multiple-test procedures and targeted non-monotonic relations while we used correlation and regression analyses to discover monotonic/linear relations. Given the non-monotonic relations approved by Vasilescu's work and the weaker correlations discovered in our investigation, we conclude that there are user groups whose activities highly correlate. These, however, require more effort to identify and cannot be immediately predicted without access to rankings in both datasets. For example, the users may be clustered based on their behaviors to identify the sub-groups of users who perform similar activities in the two networks.

Reflecting on "threats to validity", we have to mention that, in the beginning, *GitHub* did not insist on validating email addresses, so it was impossible to identify some of the common users between the two networks. Many users use different email addresses to register in different social networks. As a result, several common users might be missed by the email matching algorithm. Advanced identity-merging algorithms can resolve this issue. Finally, another potential threat to validity may be our sample. While it is large enough (255,375 users), it covers around one tenth of the size of each of the two networks. So the question remains that whether it is really representative of the real population or not.

7 Conclusions and Future Work

In this study, we analyzed the activities of developers in *GitHub* and *Stack Overflow*. We defined three high-order metrics relevant to both networks (*i.e., development*, *management* and *popularity*) in terms of network-specific basic metrics. Relying on these common high-order metrics we conducted several interesting intra- and cross-network analyses.

Our findings indicate moderate and strong correlations between the derived metrics within each platform (*e.g., development*, *management* and *popularity*). Active developers contribute to the main development activities of the platform (i.e., committing in *GitHub* and answering in *Stack Overflow*), but they also engage in other managerial activities (like managing issues in *GitHub* and vote casting in *Stack Overflow*), and thus gain more popularity.

Further, our inter-network analyses reveal several interesting points of overlap between the two communities.

1. Early adopters of *GitHub* are also early adopters of *Stack Overflow*; this may indicate that adoption is motivated by a general tendency to belong to a community rather than by the tools themselves. Interestingly, more active users are earlier adopters: the majority of super-active users registered early in both networks, and were followed by the registrations of the most active users.

2. The activity of users in *GitHub* is related to their corresponding activity in *Stack Overflow*, but only weakly. In fact, only 1.6, 9.5 and 15.6% of super-active, active and consistent users in *GitHub* are also super-active, active and consistent in *Stack Overflow*. Examining the levels of user activity in the two platforms, we found that, in the beginning, users were more active in *Stack Overflow* but in the second half of the 60-month period, *GitHub* activity surpasses that of *Stack Overflow*.

3. User activity in *GitHub* exhibits higher fluc-

tuations than activity in *Stack Overflow*, not surprisingly since there is much more variety of activity types in *GitHub* than in *Stack Overflow*. Furthermore, active users are the most consistent users since they exhibit substantial activity levels steadily, unlike typical users whose activity levels are rather low, and unlike super-active users, whose activity levels vary substantially over time.

4. In each network, the active and super-active users exhibit much more intense activity, as compared to typical users. Particularly in *GitHub*, this coefficient increases up to tens of times in the last two years. For typical users the most frequent activity is committing, posting questions, and lastly answering. For active and super-active users they post answers the most, then commit and less frequently ask questions. Finally, more active *Stack Overflow* users are engaged in more answers than questions.

All in all, although some similarities exist between the two networks, the relation between activity in the two networks is not strong enough to predict the level of activity in the other network. This is likely due to the rather different objectives served by the two platforms: *GitHub* focuses on code generation and evolution and sharing repositories while *Stack Overflow* focuses mostly on questions answering and problem solving related to software development. The results provided by our paper should be of interest to developers in both networks, i.e., *Stack Overflow* and *GitHub*, as well as many other social platforms that leverage "software" as the core of their work. While there are connections between activity of the users in such networks, their relative isolation from each other makes it difficult to extract much useful information from the data. If the tools were more related, such as for example by supporting the submission of *GitHub* issues as *Stack Overflow* questions or the delivery of *GitHub* code as example for *Stack Overflow* answers, users' activities could be more easily correlated and deeper insight could be gained on how the two platforms support and/or antagonize each other.

The merged data set studied in this paper is accessible through the following address: `http://hypatia.cs.ualberta.ca/~alisajedi/Mining_GH_and_SO-MergedDataSet.zip`

As future work, we are considering pursuing more advanced approaches for identifying common users in the two datasets, in addition to e-mail matching). For example, identity-merging algorithms [3, 6] may be used to merge the activities of different accounts related to one user. Another interesting direction for future work involves the expansion of the studied platforms to include more social networks, such as Twitter, for example.

Acknowledgements

We would like to thank Ali Yadollahi for his comments and constructive feedback on this work. Dr. Hindle's work is supported by NSERC. Dr. Stroulia's work is supported by NSERC, AITF and IBM.

References

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

[2] S. Ghosh, N. Sharma, F. Benevenuto, N. Ganguly, and K. Gummadi. Cognos: crowdsourcing search for topic experts in microblogs. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 575–590. ACM, 2012.

[3] M. Goeminne and T. Mens. A comparison of identity merge algorithms for software repositories. *Science of Computer Programming*, 2011.

[4] G. Gousios. The ghtorrent dataset and tool suite. In *Proceedings of the 10th Working Conference on Mining Software Repositories*, MSR'13, pages 233–236, 2013.

[5] S. G. Jeffrey Dean. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[6] E. Kouters, B. Vasilescu, A. Serebrenik, and M. G. van den Brand. Who's who in gnome: Using lsa to merge software repository identities. In *Software Maintenance (ICSM), 2012 28th IEEE International Conference on*, pages 592–595. IEEE, 2012.

[7] M. J. Lee, B. Ferwerda, J. Choi, J. Hahn, J. Y. Moon, and J. Kim. Github developers use rockstars to overcome overflow of news. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pages 133–138. ACM, 2013.

[8] F. Thung, T. F. Bissyandé, D. Lo, and L. Jiang. Network structure of social coding in github. In *Software Maintenance and Reengineering (CSMR), 2013 17th European Conference on*, pages 323–326. IEEE, 2013.

[9] B. Vasilescu, V. Filkov, and A. Serebrenik. Stackoverflow and github: Associations between software development and crowdsourced knowledge. In *Proceedings of the 2013 ASE/IEEE International Conference on Social Computing. IEEE*, pages 188–195, 2013.

[10] R. Venkataramani, A. Gupta, A. Asadullah, B. Muddu, and V. Bhat. Discovery of technical expertise from open source code repositories. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 97–98. International World Wide Web Conferences Steering Committee, 2013.

[11] D. J. Watts and P. S. Dodds. Influentials, networks, and public opinion formation. *Journal of consumer research*, 34(4):441–458, 2007.