Linking Accounts across Social Networks: the Case of StackOverflow, Github and Twitter

Giuseppe Silvestri[1,2], Jie Yang[1], Alessandro Bozzon[1], and Andrea Tagarelli[2]

[1] Delft University of Technology, the Netherlands
giuseppe.silvestri.gios@gmail.com, {j.yang-3, a.bozzon}@tudelft.nl
[2] University of Calabria, Italy
andrea.tagarelli@unical.it

Abstract. Social Web accommodates a wide spectrum of user activities, including information sharing via social media networks (e.g., Twitter), question answering in collaborative Q&A systems (e.g., StackOverflow), and more profession-oriented activities such as social coding in code sharing systems (e.g., Github). Social Web enables the distinctive opportunity for understanding the interplay between multiple user activity types. To enable such studies, a basic requirement, and a big challenge, is the ability to link user profiles across multiple social networks.
By exploiting user attributes, platform-specific services, and different matching strategies, this paper contributes a methodology for linking user accounts across StackOverflow, Github and Twitter. We show how tens of thousands of accounts in StackOverflow, Github, and Twitter could be successfully linked. To showcase the type of research enabled by datasets built with our methodology, we conduct a comparative study of user interaction networks in the three platforms, and investigate correlations between users interactions across the different networks.

1 Introduction

Social Web comprises a diversity of social networking platforms, which cover a wide range of user activities. With the fact that a single user has multiple accounts across different social networks, it has now become increasingly important to link distributed user profiles belonging to the same user from multiple sources, which can benefit various applications. For instance, it has been shown that aggregating user profiles could improve personalized Web service such as recommendation systems by solving the cold-start problem [1].

Linking user profiles across multiple social networks also provides an opportunity for better understanding the interplay between different types of people's activities. Let us take as an instance the domain of software programmers: they share software related content in Twitter, seek or provide answers to software engineering related questions in StackOverflow, and collaboratively code in Github. These three different social networks (i.e., Twitter, StackOverflow and Github) are used by programmers differently, in terms of their purposes and correspondingly their activities. By aggregating the data sources from multiple networks, we might explore at large scale the complete spectrum of programmers' on-line professional activities.

Linking users' accounts across multiple social networks is considered a well-known problem, thus attracting multiple techniques and solutions [1, 6–8, 4, 5]. Previous studies addressed the online activities of professional users, but investigated a single type of activities in a single system [6, 7], or between two systems from a single perspective. For instance, [8] analyzes how participation in Q&A systems influences developers' productivity. [2] also considered the influence that each user has within and across two platforms, while exploiting features provided by StackOverflow (Up Votes and Questions) and Github (popular users are engaged more in commits, projects and issues). [3] focused on bridge users, in order to recognize how these users can favor information exchange across networks.

To drive a deeper investigation over users' professional activities, we are motivated to construct a cross-system users' accounts matching dataset from Twitter, StackOverflow, and Github, to enable future studies of professional activities from multiple perspectives. For instance, a dataset as such can help us understand how different types of users (e.g., users with different expertise) are engaged in different professional activities; it can also help in understanding how different types of social interactions among users can influence the evolution of communities of different professional activities. This paper contributes a methodology to link online users' accounts across Twitter, StackOverflow and Github, by exploiting different attributes of user profiles, platform specific API's and services, and a variety of accounts' matching strategies. As a first trail of valuing this dataset, we construct three social networks, including follower-followee networks of Twitter and Github, and helper-helpee networks of StackOverflow. By characterizing the networks features, we present our findings of how users interact with others in different activities, and how different activities of the same user correlate with each other.

The rest of the paper is organised as follows. Section 2 describes our methodology of matching users across StackOverflow, Github and Twitter, together with the corresponding results of user matching. Based on these matched users, Section 3 introduces our comparative study of user interactions between three user interaction networks in StackOverflow, Github ad Twitter, and Section 4 concludes our work.

2 Linking Accounts across Social Networks

This section describes our methodology of matching users across StackOverflow, Github and Twitter. We first discuss the general settings of data retrieval for the three social networking platforms, then present our user matching strategies and workflows.

2.1 Retrieving data from multiple platforms

StackOverflow. We downloaded the most recently released data dump from Internet archive[3]. Due to privacy concern, since the end of 2014[4] StackOverflow

[3] `https://archive.org/details/stackexchange`, accessed at April, 2015

[4] `http://meta.stackexchange.com/questions/221027/where-did-emailhash-go`

data dump no longer contains hashed user emails. While not crucial, hashed emails are a convenient and effective way to unambiguously match accounts. To overcome this limitation, we extended the data from the data dump released on September 2013 (which is the last released dump with hashed email addresses) with the latest data contained in the 2015 data dump.

Github. The GHTorrent project[5] has incrementally released Github data every two months since March 2012. We parsed its data from the first release containing user information (i.e., July, 2012) until the latest one on March 2015, and kept all versions of user information in our database for account matching.

Twitter. Given an existing user name, the related account information (e.g., profile picture, website) and related posts in Twitter can be retrieved via Twitter REST API[6]. The Twitter.com Search[7] functionality, on the other hand, allows for fuzzy retrieval of users accounts, returning a candidate set of accounts having screen names similar to the one provided as input. For our purposes, the latter proved more useful than the former for fuzzy matching.

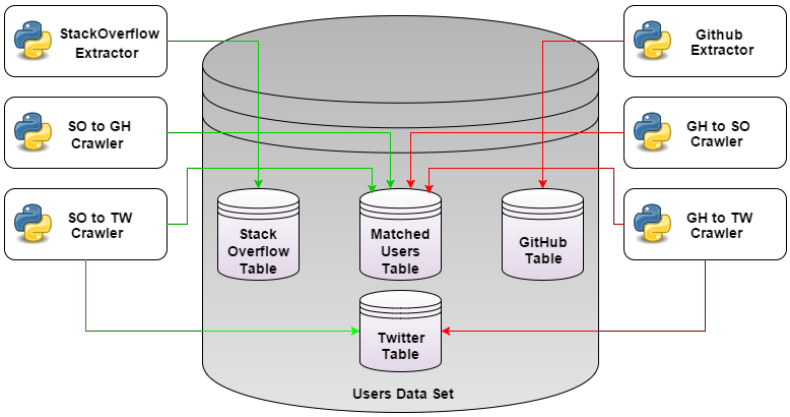


Fig. 1: Data collection workflow. SO, GH, TW are short for StackOverflow, Github, and Twitter, respectively.

The main workflow of accounts' linking across the three platforms is depicted in Figure 1. Accounts from StackOverflow and Github were dumped and processed first. We retrieved 4,132,407 and 4,288,132 accounts in StackOverflow and Github, respectively. These sets of accounts were then matched to each other and the resulting overlap further matched to a set of Twitter accounts. The latter was retrieved using a strategy that we will discuss later in this section.

[5] `http://ghtorrent.org`, accessed at April, 2015
[6] `https://dev.twitter.com/rest/public`
[7] `https://twitter.com/search-home`

2.2 Matching accounts across multiple platforms

We design three account matching strategies to find the same set of users in the platforms under study:

- **explicit matching**, which aims at identifying the links explicitly provided by users in one platform to their accounts in other platforms for user matching.
- **attribute-based matching**, which leverages unique attributes of users' accounts such as email to connect profiles across multiple platforms from the same user.
- **fuzzy matching**, which exploits less accurate user attributes such as login names and profile images to match user profiles.

Explicit matching is performed to link user accounts between StackOverflow and Github, and further link them to Twitter. Attribute-based matching is performed only between StackOverflow and Github, while fuzzy matching aims at linking matched users in StackOverflow and Github to Twitter. We introduce as follows the concrete steps we took for each of the matching strategies.

Explicit matching. Starting from our built dumps of StackOverflow and Github, we perform explicit matching by analyzing user-provided links from the user profiles in each of these platforms to the other platforms. We consider this a very reliable method for account linking: matching information are provided by users themselves, with strong incentives for truthful linking.

From StackOverflow to Github, Twitter. We analyze StackOverflow user profiles to find explicit links to GitHub and Twitter users. For StackOverflow users that provide links to their Github link, we parse the direct links, which are in the form of `https://github.com/GitHubLoginName` and obtain their Github login names, i.e., `GithubLoginName`. For StackOverflow users that provide direct links to Twitter, which is usually in the form of `http://www.twitter.com/TwitterScreenName`, we parse the Twitter screen name, i.e., `TwitterScreenName`. Both GitHub login name and Twitter screen name uniquely identifies one user in GitHub and Twitter, respectively.

From Github to StackOverflow, Twitter. We analyze Github user profiles similarly to match user profiles in StackOverflow and Twitter. For StackOverflow, we adopt an additional strategy to obtain a cross-reference to the same user: since some Github users provide their StackOverflow Careers profile[8], which is a CV-like page of senior StackOverflow users, we parse the HTML code of the corresponding pages in order to retrieve the direct link (in the form `http://stackoverflow.com/users/id`) to their real StackOverflow profile pages.

The result of explicit matching is reported in Table 1. As it can be observed, we are able to match thousands of users between the three platforms.

[8] `http://careers.stackoverflow.com/`, StackOverflow Careers

From	To	#Matched Users
StackOverflow	Github	4,536
	Twitter	10,068
Github	StackOverflow	433
	Twitter	7,012

Table 1: Explicit matching.

Attribute-based matching. StackOverflow and Github provide users with the option of registering their emails, which are encrypted into MD5 hashes in the data dumps. This technique is known from literature [8, 2] to be a reliable way to match users by their email reference.

There are in total 2,185,162 (≈52.9%) StackOverflow users and 510,523 (≈11.9%) Github users with email hash. Note that email hashes were previously considered for matching users between StackOverflow and Github in [8]. Besides using the email hashes explicitly provided by users, we exploit Gravatar[9] to increase the number of available hashes in both platforms. We find that many users use Gravatar to have a unique profile image across StackOverflow and Github. By making HTTP request for a Gravatar profile image, we obtain a user's MD5 email hash [10]. We identified 2,897,175 (≈67.6%) Github users, and 430,860 (≈10.4%) StackOverflow users with Gravatar email hash available.

$$\begin{aligned} query = &((StackOverflowUsers[emailhash] \cap GithubUsers[emailhash]) \\ &\cup (StackOverflowUsers[gravatarid] \cap GithubUsers[gravatarid]) \\ &\cup (StackOverflowUsers[emailhash] \cap GithubUsers[gravatarid]) \\ &\cup (StackOverflowUsers[gravatarid] \cap GithubUsers[emailhash])) \end{aligned} \tag{1}$$

Combing email hashes explicitly provided by users, and implicitly revealed from their Gravatar Id, we use Query 1 for StackOverflow-Github user matching, which encodes all meaningful joins between MD5 email hash and Gravatar Id attributes across the two platforms. The result of attribute-based matching is shown in Table 2. We finally obtained more than 600k exactly matched users between StackOverflow and Github.

Fuzzy matching. Matching accounts from StackOverflow and Github with Twitter accounts is intrinsically more difficult, since Twitter profiles need to be obtained via Twitter API services.

Lookup and search. Two types of query requests are here considered, namely Twitter REST API and Twitter.com Search, hereinafter referred to as *Lookup*

[9] `https://en.gravatar.com/` Gravatar, a globally recognized avatar.
[10] `https://en.gravatar.com/site/implement/images/` Gravatar: Image Request

Type	#Matched Users
SO emailhash - GH gravatarid	580,979
SO emailhash - GH emailhash	107,572
SO gravatarid - GH emailhash	1,224
SO gravatarid - GH gravatarid	4,752
Union all above types	604,083

Table 2: Attribute-based matching between StackOverflow and Github.

and *Search*, respectively. The former method returns the full profile information of the user corresponding to a given user screen name. Using Twitter REST API, each request can process up to 100 inputs. By contrast, Twitter.com Search permits to process only a single input for each request. While being less efficient, Twitter.com Search is however more flexible in terms of the input — it accepts any textual input.

We consider the following options of input for the *Search* method:

- login names, and names of users' StackOverflow and Github accounts;
- URLs of user's StackOverflow and Github accounts;
- users' website URLs identified from their StackOverflow and Github profiles.

To find the best input for the *Search* method, we analyzed how many accounts can be matched by using different user attributes. Matching is performed in two steps: (1) given a user attribute, retrieve candidate users via Twitter.com Search; (2) try matching the website URL of the Twitter candidates and the website URL of the user StackOverflow (Github) profile. Results have shown that using Github login name provides better matching of Twitter profiles than the URLs of their accounts in StackOverflow or Github, as well as their website URLs. We therefore chose to take Github login name as an input for *Search* to retrieve candidate Twitter profiles for matching.

Accuracy of *Lookup* and *Search* methods. To assess the performance of the *Lookup* and *Search* matching methods, we first categorized the Github login names into the following categories: 1) the login contains only lower-case characters, 2) it contains at least one upper-case character, 3) it contains numbers, and 4) it contains special characters. Figure 2a shows the distribution of Github login names according to the categorization above, from which we observe that the majority of them are in the "lower-case" category.

To understand how different categories differ in the probability that at least one candidate can be returned by *Lookup* and *Search*, in Figure 2b we analyzed the percentages of Github login names that have at least one candidate returned by *Lookup* and *Search*. High values indicate higher probability that the user can be matched. We observe that the *Search* method performs better than *Lookup* in all categories except in the "Number" category.

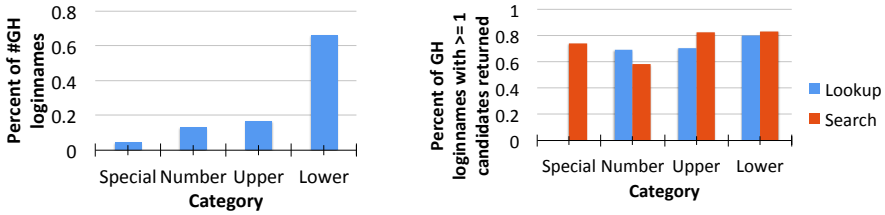


Fig. 2: Distributions of Github login names in selected categories (a) and number of candidates returned by *Lookup* and *Search* for each category (b).

Category	Lookup	Search	Gain
Lower	.29	.39	+.10
Upper	.28	.33	+.05
Number	.24	.27	+.03
Special	.00	.19	+.19

Table 3: Accuracy of Twitter user matching using *Lookup* and *Search* for different categories of Github login name.

For each category, we randomly selected 100 Github login names, took them as input for both *Lookup* and *Search* methods, then manually checked the matched accounts. A user is considered to be matched with a Twitter account if there is explicit Twitter information (e.g., personal website, profile description) that can identify the user with high confidence. Table 3 shows that *Search* performs better than *Lookup*, especially for Github login name that belong to the "lowercase" and "special characters" categories. The least gain of *Search* over *Lookup* corresponds to the category "Number" (less than 5%). Considering Figure 2b and the higher efficiency of *Lookup* method, we chose to use *Lookup* for Github login names in the "Number" category, and *Search* for the other categories.

Workflow of fuzzy matching. Figure 3 depicts the workflow of *Lookup* and *Search* methods. Given a user Github login name, it first determines whether to use *Lookup* or *Search*, then checks Twitter profiles for account matching. In the step of "Twitter Profile Check", a user is matched to a Twitter account if s/he satisfies the following criteria:

1. the website attribute of the user's Twitter profile is exactly the same as the website of his/her StackOverflow (Github) profile;
2. otherwise, the Twitter profile picture needs to be highly similar (e.g., $\geq 90\%$) to her/his profile picture in StackOverflow (Github).

In criterion 1 we ignored ambiguous websites such as `http://facebook.com`, which can bring to have False Positive for website matching, while for

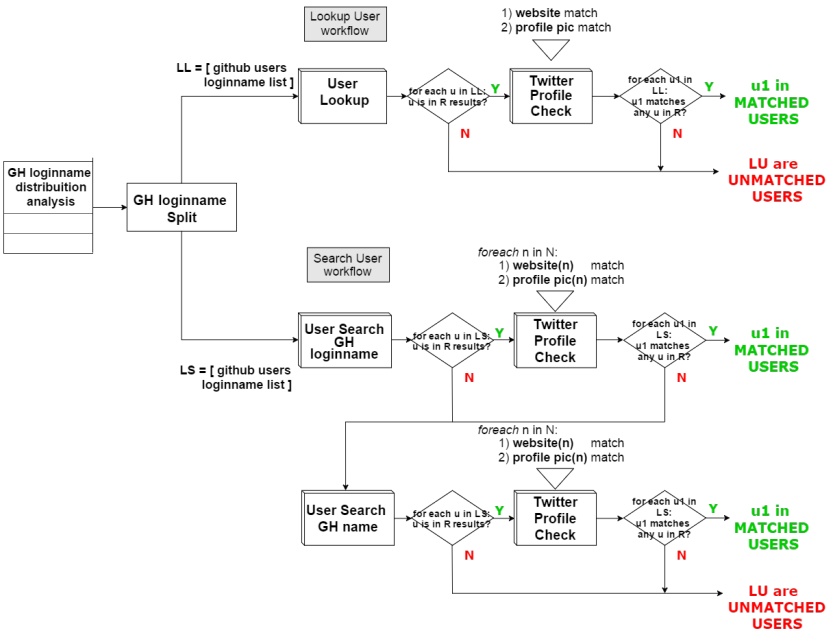


Fig. 3: Twitter *Lookup* and *Search* workflows.

criterion 2 we performed image similarity via *image hashing* [11]. We manually checked 100 users matched by website and profile picture, respectively. As before, user profiles were considered as matched if the provided information gives high confidence that they belong to the same user. The true positive rate of website and profile pictures are 100% and 98%, respectively, which indicate that users can be regarded as exactly matched.

Search method	#Users analyzed	#Users matched	Matching %
Lookup	176,508	9,316	5.28%
Search	240,000	37,449	18.43%
Total	**416,508**	**46,765**	**11.23%**

Table 4: Twitter user matching results.

To account for limitations with the Twitter APIs, at the time of this writing we were able to analysis a subset of linked accounts from StackOverflow and Github. We ordered accounts according to their popularity (measured by #fol-

[11] `http://hzqtc.github.io/2013/04/image-duplication-detection.html`, Image Duplication Detection

Graph	# Nodes	# Edges	Density
G_{SO}	6672	18995	4.267e-04
G_{GH}	13160	106792	6.167e-04
G_{TW}	16070	829846	2.213e-03

Table 5: Characteristics of the user networks in Twitter, StackOverflow, and Github.

lowees) in Github, and matched them to Twitter accordingly. Table 4 reports the user matching results. We analyzed 416k accounts, specifically 240k by using *Search* and 176k by using *Lookup*. The number of accounts matched are 37k and 9k, respectively, with a total of 46k accounts matched to Twitter.

3 User Interaction across Networks

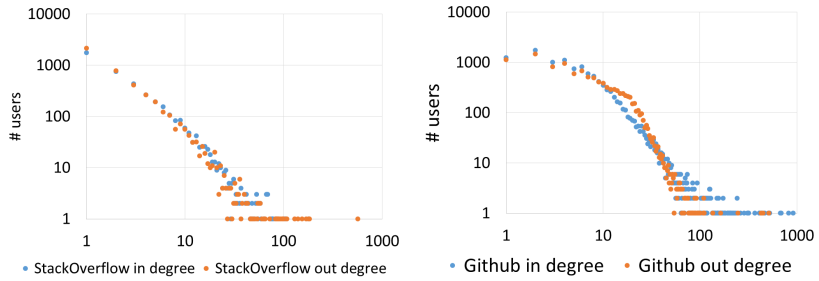
To showcase the type of research that is enabled by a dataset built with our methodology, we designed a study aimed at providing an answer to the following two research questions: *RQ1: how do users connect with each other in different social networks? RQ2: does the relative importance of users vary across social networks?* To this end, we first inferred the interaction networks over the same set of users in the three platforms, then analyzed network features and correlations of user centrality in the three networks.

Building user interaction networks. We constructed two directed graphs G_{TW}, G_{GH} that encode *following* relationships of users in Twitter and Github, respectively, i.e., a directed edge $e = u \rightarrow v$ indicates that user u follows user v. While being absent of explicit following-followee relationship, StackOverflow provides an implicit "help network" among users according to *who answers to whom*. Therefore, we built a directed graph G_{SO} such that an edge $e = u \rightarrow v$ indicates that user u is helped by v, i.e., at least one question of u is answered by v.

Due to the rate limit of Twitter REST API, we built the three user interaction network graphs for the 20k most popular users among the 46k matched users (Table 4). As before, popularity is defined according to #followees in Github.

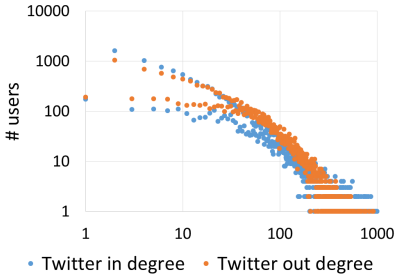
RQ1: How do users connect with each other in different social networks? Table 5 reports basic statistics of the users' networks in the considered social networks. By comparing the #nodes in the three networks, we observe that, in the same set of 20k users, more users are involved in both Github and Twitter interaction networks than those involved in StackOverflow interaction network. This indicates that users are more likely to be active in explicit interaction based on followship than in helping-based interaction.

Comparing the density of these networks, results show that users have similar connection intensity in StackOverflow and Github, both of which are however



(a) StackOverflow in- and out- degree.

(b) Github in- and out-degree.



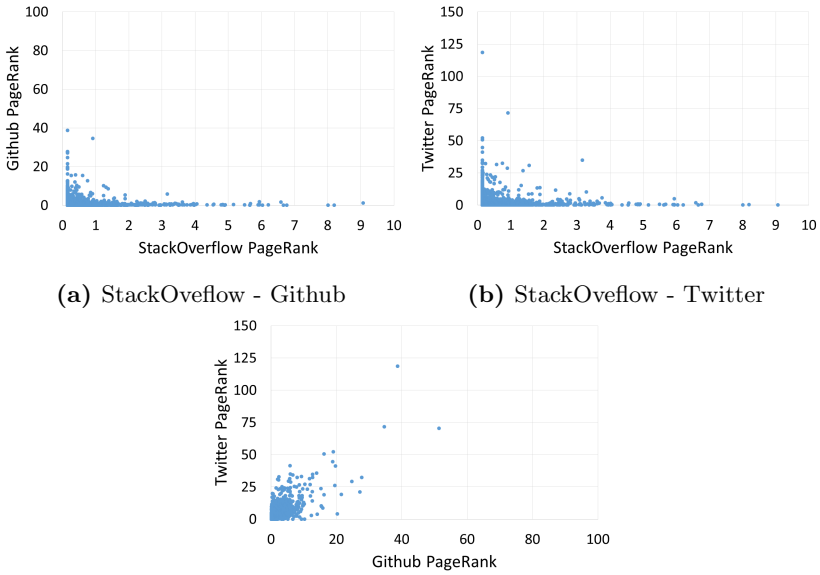
(c) Twitter in- and out-degree.

Fig. 4: Degree distribution for Twitter, StackOverflow and Github networks.

10 times lower than user interaction in Twitter. This would imply that users are more likely to connect with each other in general-purpose social networks like Twitter than in profession-oriented networks like StackOverflow and Github.

Figure 4 shows the in-degree and out-degree distributions over the three networks. In StackOverflow, both distributions conforms to power-law, indicating that most users follow (resp. are followed by) a small number of users, while there is a small number of users that follow (resp. are followed by) many users. In addition, in-degree distribution looks more skewed than out-degree distribution – in other words, users tend to follow the same set of users, who is followed by many users. Similarly in Github and Twitter, in-degree distribution is more skewed than out-degree distribution, indicating that a small number of users are highly popular in the network. Comparing the three networks, StackOverflow is the one that has most similar distributions of in-degree and out-degree. We consider the fact that the StackOverflow helping-helpee network is built implicitly from question-answering activity between users, while the following-followee relations in Github and Twitter are explicitly constructed by users. The result suggests that explicit connection mechanisms result in a more skewed popularity among the users of a platform.

RQ2: does the relative importance of users vary across social networks? To answer this question, we choose to correlate users' centrality scores



(a) StackOveflow - Github **(b)** StackOveflow - Twitter

(c) Github - Twitter

Fig. 5: Pair-wise network centrality correlations.

in the different networks. A high cross-network correlation of user centrality scores would indicate similar user importance in different settings; for instance, a high correlation of user centrality in StackOverflow and Github networks will suggest that a user who is helpful in answering to others' questions in StackOverflow will be followed by many users in Github (and vice versa); on the contrary, a low correlation would indicate that users' activity in one platform is not indicative of their activities in another platform, e.g., an influential user in Github may not likely to answer questions in StackOverflow.

To obtain users' centrality values, we used classic *PageRank* model. We then calculated Pearson correlation of the centrality values for the same set of users in every pair of graphs. Results are shown in Figure 5. For StackOveflow and Github networks, we have a Pearson coefficient of *-0.0185170* that reveals no linear correlation between PageRank values of users on both platforms; this means that, as shown in Figure 5a, most influential users on StackOverflow do not have the same importance on Github and vice versa. Similar remark can be made on StackOveflow versus Twitter, where Pearson correlation is *-0.0014857*. By contrast, in the Github - Twitter case, we observe a Pearson coefficient of *0.7554060*, which implies that user interactions of Github and Twitter networks are correlated.

4 Conclusions and Future Works

We addressed the problem of user matching across StackOverflow, Github and Twitter social networks. We proposed a methodology that combines different matching strategies and makes use of different user attributes and platform-specific services for linking user accounts. Many of the proposed linking strategies can be generalized to other social networking platforms. For instance, most social networking platforms provide REST API's and search, for which the linking techniques *Lookup* and *Search* can be applied. These methods together allow us to obtain much better results than in literature. Our study of interaction networks based on the matched users in the three platforms has provided interesting insights: 1) users in general-purpose social media networks like Twitter are more connected than in profession-oriented social networks like StackOverflow and Github; 2) social networking platforms that enable the functionality of explicit user connection (Github and Twitter) will result in more skewed distribution of user popularity, and more correlated user activities between them, than (with) the one that only provides implicit user connection mechanisms (StackOverflow). As part of future work, we plan to deepen our analysis of the user interaction networks properties such as the formation and evolution of communities, and the topics discussed by the users and communities across the three networks.

References

1. F. Abel, E. Herder, G.-J. Houben, N. Henze, and D. Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013.
2. A. S. Badashian, A. Esteki, A. Gholipour, A. Hindle, and E. Stroulia. Involvement, contribution and influence in github and stack overflow. In *CASCON '14 Proceedings of 24th Annual International Conference on Computer Science and Software Engineering*, pages 19–33. ACM, 2014.
3. F. Buccafurri, V. D. Foti, G. Lax, A. Nocera, and D. Ursino. Bridge analysis in a social internetworking scenario. *Inf. Sci.*, 224:1–18, 2013.
4. F. Buccafurri, G. Lax, A. Nocera, and D. Ursino. Discovering missing me edges across social networks. *Inf. Sci.*, 319:18–37, 2015.
5. P. Jain, P. Kumaraguru, and A. Joshi. @i seek 'fb.me': Identifying users across multiple online social networks. In *Proceedings of the 22Nd International Conference on World Wide Web Companion*, WWW '13 Companion, pages 1259–1268, Republic and Canton of Geneva, Switzerland, 2013. International World Wide Web Conferences Steering Committee.
6. C. Treude, O. Barzilay, and M.-A. Storey. How do programmers ask and answer questions on the web?: Nier track. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 804–807. IEEE, 2011.
7. J. Tsay, L. Dabbish, and J. Herbsleb. Influence of social and technical factors for evaluating contribution in github. In *Software Engineering (ICSE), 2014 36rd International Conference on*, pages 356–366. ACM, 2014.
8. B. Vasilescu, V. Filkov, and A. Serebrenik. Stackoverflow and github: associations between software development and crowdsourced knowledge. In *Social Computing (SocialCom), 2013 International Conference on*, pages 188–195. IEEE, 2013.