Mining Developer Behavior Across GitHub and StackOverflow

Yunxiang Xiong†, Zhangyuan Meng†, Beijun Shen†, Wei Yin‡
†School of Software, Shanghai Jiao Tong University, Shanghai, China
‡China Aeronautical Radio Electronics Research Institute, Shanghai, China
Email: {bruinx, 602389789, bjshen}@sjtu.edu.cn, yin_wei@careri.com

***Abstract*—Nowadays, software developers are increasingly involved in GitHub and StackOverflow, creating a lot of valuable data in the two communities. Researchers mine the information in these software communities to understand developer behaviors, while previous work mainly focuses on mining data within a single community. In this paper, we propose a novel approach to mining developer behaviors across GitHub and StackOverflow. This approach links the accounts from two communities using a CART decision tree, leveraging the features from usernames, user behaviors and writing styles. Then, it explores cross-site developer behaviors through T-graph analysis, LDA-based topics clustering and cross-site tagging. We conducted several experiments to evaluate this approach. The results show that the precision and F-Score of our identity linkage method are higher than previous methods in software communities. Especially, we discovered that (1) active issue committers are also active question askers; (2) for most developers, the topics of their contents in GitHub are similar to that of their questions and answers in StackOverflow; (3) developers' concerns in StackOverflow shift over the time of their current participating projects in GitHub; (4) developers' concerns in GitHub are more relevant to their answers than questions and comments in StackOverflow.**

***Index Terms*—Identity Linkage; Developer Behavior Mining; Machine Learning; GitHub; StackOverflow**

I. Introduction

In recent years, software developers are intensively involved in open source software development communities (e.g. GitHub) and knowledge sharing communities (e.g. StackOverflow). As developers continuously contribute to or exchange ideas through these communities, a lot of development data and knowledge are accumulated there. According to the data from ghtorrent[1] and archive.org[2], there are more than 20 million repositories and 5 million developers in GitHub, 30 million posts and 6 million users in StackOverflow as of August 2016. It is a great opportunity to understand characteristics and working habits of software developers through analyzing and mining these data.

Previous studies mainly focus on mining data within a single community [1] [2]. Meanwhile, it is more beneficial to conduct cross-community behavior mining, as some deep behaviors and developer relations can be discovered. Thus researches have been conducted on linking accounts across software communities and then making analysis. For example, Vasilescu [3] tried to find some associations between software development and crowdsourced knowledge, although the cross-site analysis is still quite simple.

Generally, we are facing three challenges when mining developer behaviors across software communities:

1) Identity linkage problem. The traditional social network relies on user names and email addresses for linking identities between communities. However, StackOverflow has no longer provided users' email addresses. Thus it lacks strong evidence for linking users with the same identities between GitHub and StackOverflow.
2) Data heterogeneous problem. Data across different software communities is heterogeneous. For example, labels of repositories in GitHub are programming languages, but those of Q&As in StackOverflow are technical terms.
3) Association mining. After identity linkage, it is also challenging to find the associations of developer behaviors across GitHub and StackOverflow, and recover the latent, valuable information of these data.

To address these challenges, this paper proposes a novel approach to mining developer behaviors across GitHub and StackOverflow, as shown in Fig. 1. It consists of two phases: identity linkage and behavior mining. At the identity linkage phase, we extract the features from the developer profile and behavior data, including the similarity between usernames, user behaviors, and user writing styles. And then classification and regression tree (CART) algorithms are applied to link the accounts of developers between GitHub and StackOverflow. At the behavior mining phase, we raise three research questions for exploring the patterns on developer behaviors across these two software communities. Statistics, Natural Language Processing (NLP) and machine learning technologies are adapted to analyze and mine the merged developer behavior data.

Our main contributions are summarized as follows:

1) We propose an approach to mining cross-site developer behaviors. It links identities between GitHub and StackOverflow by leveraging features from usernames, user behaviors, and writing styles, using CART decision tree. And then it mines the merged developer behavior data to find some valuable observations.
2) We conducted several experiments to evaluate the mining approach. The results show that the precision and F-Score of our identity linkage method are higher than

DOI reference number: 10.18293/SEKE2017-062
[1]http://www.ghtorrent.org
[2]https://archive.org/download/stackexchange

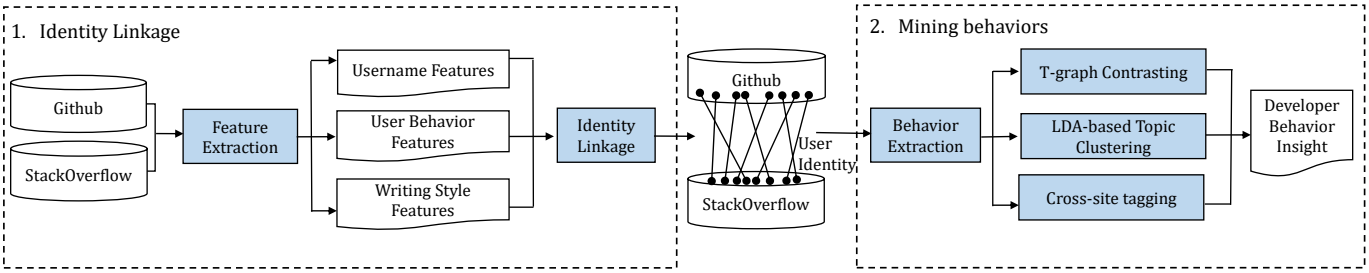

Fig. 1: Approach Overview

previous methods in software communities. Especially, we discovered that (1) active issue committers are also active question askers; (2) for most developers, the topics of their contents in GitHub are similar to that of their questions and answers in StackOverflow; (3) developers' concerns in StackOverflow shift over the time of their current participating projects in GitHub; (4) developers' concerns in GitHub are more relevant to their answers than questions and comments in StackOverflow.

II. Related Work

A. Identity Linkage in Software Communities

Several methods have been proposed to solve the identity linkage problem in software communities. A simple algorithm [4] was proposed by Goeminne using several simple rules to judge if two user pairs are the same person. Bird et al. [5] proposed a more advanced algorithm that some text similarity metrics are used, such as Levenshtein distance. Furthermore, a semantic-based method LSA was proposed to link users by Erik [6]. However, these methods didn't use username as the most important one of features to improve prediction accuracy, nor do they take full advantage of the textual information left by developers in software development.

B. Mining Developer Behaviors on Software Communities

There are many researches focusing on mining the data from a single community. For example, William [1] showed some developer behaviors and sentiments from open source repository mining; Christoph [2] argued that how do programmers ask ad answer questions on StackOverflow. On the other hand, Few people studied the association between these two communities. Bogdan [3] investigated the interplay between StackOverflow activities and the development process in GitHub. They show that the QA activity rate correlates with the code changing activity. So, across software communities, there are still a lot of patterns and insights to explore.

III. Identity Linkage across Software Communities

Accounts from GitHub and StackOverflow are less connected. Linking these accounts, which is called identity linkage, is a prerequisite of behavior mining across these two communities. In this section, we define the identity linkage problem, and then give our method, and its experimental results.

A. Problem Definition

Let P denote the collection of all natural persons, U_g denote the collection of users in GitHub and U_s denote the collection of users in StackOverflow. Let T(u) = p be a mapping function to map a user in GitHub or StackOverflow to a nature people in the real world. Now our goal is to find such a function f to solve the identity problem. For a user pair(c_1,c_2), where $c_1 \in U_g$ and $c_2 \in U_s$, $f(c_1, c_2) \rightarrow \{0, 1\}$. If T($c_1$) = T($c_2$), which means c_1 and c_2 refer to the same nature people, $f(c_1, c_2)$ equals 1, else $f(c_1, c_2)$ equals 0.

B. Feature Extraction

To solve the identity linkage problem defined above, we extract three kinds of features to calculate the similarities between two users firstly. Then we use a ground-truth data collection based on all these features as the input data and apply CART Decision Tree, a machine learning algorithm, to train an identity linkage model. Finally, each user pair is assigned a probability by the model. Here, these features are presented in detail, which are username features, user behavior features, and writing style features.

1) Username Features: Since people usually use similar usernames in different communities, it is very significant to extract useful features from usernames. Four string matching algorithms are chosen to measure the username similarity features: (1) levenstein distance, which is the number of transfers needed to transfer one string to another one; (2) jaro winkler distance, which is commonly used for measuring the similarity of short strings especially for usernames; (3) longest common substring, which is the longest string which is a substring of two strings; (4) longest common subsequence, which is the longest subsequence common to two strings.

The value of the jaro winkler distance is in the range [0,1], and we compute the ratio and transfer the value of the other three methods in the same range. In the experiments, we will set all these metrics as features to predict the identity linkage using the decision tree model, and two of them with the best performance will be chosen.

2) User Behavior Features : Existing empirical studies about social behaviors (e.g., [7]) show that, a user's social behavior exhibits a surprisingly high level of consistency across different communities over a sufficiently long period of time. It is rational to hypothesize that two users in GitHub and StackOverflow correspond to the same nature people in real world if they have a high level of synchrony.

For each repository in GitHub or each question in StackOverflow, there is a tag labeled to describe the programming language or relational technologies. Therefore, we can obtain topics in user behaviors with these tags, and measure the similarity of two user behaviors by the distribution similarity of the topics in their behaviors.

However, the tagging systems in these communities are very different. For example, the number of tags in GitHub and StackOverflow are 57 and 21300 respectively. Tags in GitHub are marked by the system automatically but by users themselves in StackOverflow. To solve this problem, we converse those synonymous tags into the same tags based on a synonyms relation[3], and a common set of all tags both in GitHub and StackOverflow are extracted. Then these tags are used to build the distribution of topics in user behaviors.

Another problem we encountered is, people are not always using different communities at the same time so that amount of information could be missing in such a process. Therefore we propose a user behavior matching method inspired by bio-stimulation [8] to reduce the impact of that problem. The main idea of the bio-stimulation is that the maximum stimulation from a pooled signal set plays a significant role for perception. So we segment the user behaviors by different time period. For example, we divide developers' behaviors into four quarters for each year and three months for each quarter.

Suppose each user pair (u^1,u^2) where $u^1 \in$ GitHub and $u^2 \in$ StackOverflow. For each month, we obtain all the language tags of the projects u^1 in GitHub and get a tag distribution for the tags belong to the common set. At the same time, we also catch all the questions u^2 asked on the StackOverflow in this month and get a tag distribution. Then, a cosine similarity is used to measure the user behaviors similarity for the month, denoted as $s_{mr}(i)$. Following formula is defined to calculate the similarity of user behaviors for a quarter and for a year, where N is the number of months. In our experiment, we measure an average similarity of user behaviors per year, as the final behavior similarity between two users.

$$S_{mr} = \frac{1}{N}(\sum_{i=1}^{N}(s_{mr}(i))^q)^{\frac{1}{q}}, q \geq 1 \quad (1)$$

3) Writing Style Features: Most of user generated data in the GitHub and StackOverflow are textual. Some studies [9] [10] [11] have shown that the user's writing style can help to achieve reliable results in user recognition situations. So we apply the method in [11] to extract user writing style features, listed in table I. Then, the writing style similarity between two users is measured by KL-divergence, using those features.

C. Identity Linkage

With all the features above, we train a identity linkage model on a ground-truth data set, using classification and regression tree (CART). By this model, we can obtain the probability of whether the user pair refer to the same user. And then, the identity linkage problem is converted to a matching problem

[3]http://stackoverflow.com/tags/synonyms

TABLE I: WRITING STYLE FEATURES

Feature	Definition
Length	number of different words
Vocabulary richness	frequency of hapax legomena, ddis legomena
Word shape	frequency of words with different combinations of upper and lower case letters
Word length	frequency of words that have 1-20 characters
Letters	frequency of a to z, ignoring case
Digits	frequency of 0 to 9
Punctuation	frequency of . ? ! , ; " ( )
Function words	frequency of words like 'the', 'of', and 'then'

in bigraph. Suppose user u^1 in GitHub and each candidate user u^2 in StackOverflow, a conditional Heuristic Greedy Matching (HGM) is used to avoid false positive matching and finally generate a candidate user pair as (u^1,u^2).

For example, in Fig. 2, each user pair has a probability which is assigned by the CART decision tree. In HGM, as shown in Fig. 2(a), the information generation of each user is calculated firstly. Suppose user X is the owner of the most information and his best candidate is X'. User pair (X, X') will be the first linkage selected by HGM because (X, X') shares the highest probability. After selecting, X and X' will be deleted in the candidate list before our next matching. Finally, three user pairs (X, X'), (Y, Y') and (Z, Z') are matched shown in Fig. 2(b).

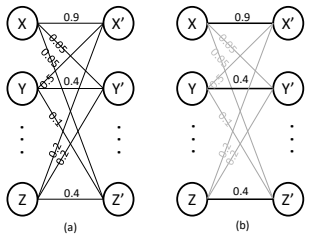

Fig. 2: An Example of Heuristic Greedy Matching Algorithm

D. Experiments

The experiment data is from ghtorrent[1] and archive.org[2] before May, 2016. To construct the ground-truth data, users' profile urls are used as the evidence to verify who are the same users. As the result, 16,000 users are linked corresponding to 8,000 people and we divide them into 5 groups and adopt 5-folder cross validation to evaluate the performance of our method.

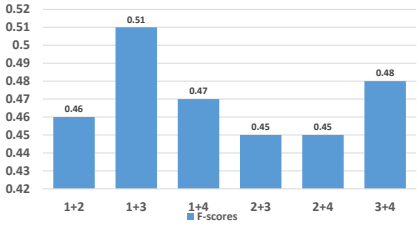

Fig. 3: Results for Combination of Different Username Metrics

1) Selection of Username Metrics: With the decision tree model, all potential combinations from 4 metrics are set as features to do prediction, and the best combination is selected. The result is illustrated in Fig. 3, where levenstein distance,

longest common substring, longest common subsequence and jaro winkler distance are abbreviated by 1,2,3 and 4 respectively. The experiment shows that the combination of levenstein distance and longest common subsequence has the best performance.

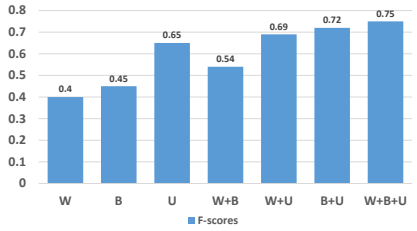

Fig. 4: Contribution for Different Features

2) Feature Contribution: In this experiment, all kinds of features above are used to train the decision tree model. They are username similarity (abbreviated by U), user behavior similarity (abbreviated by B), and user writing style similarity (abbreviated by W). Fig. 4 illustrates how the model is affected by each feature and the combination of features. It is demonstrated that the model trained with all features has the best performance.

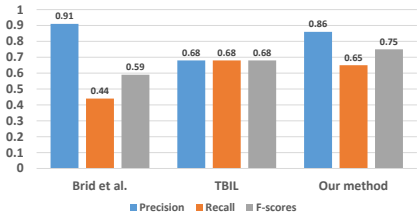

Fig. 5: Comparison with Other Methods

3) Comparison with Other Methods: We compare our method with the TBIL [12] method and the Bird's method [5]. Bird focuses on the username and email address of users. The TBIL is a state-of-the-art method, which uses usernames, user topics and user skills as features. Fig. 5 illustrates the result of the comparison. Bird's method achieves a high precision (around 0.91), but low recall and F-Score. That is because not all user use similar usernames among different communities and some users don't offer their email addresses. Since the TBIL uses a full matching strategy, it achieves a same precision, recall and F-Score (0.718). Our method has a good precision and recall, and get the best F-Score (around 0.75). Compared to those methods, our method adopts a bio-stimulation method with more features, which can analyze the user's topics and actions meticulously, and thus has a better performance. Furthermore, we use a conditional greedy-based matching to avoid false positive matching, since not all users have two different accounts in GitHub and StackOverflow.

IV. Mining Developer Behavior

After identity linkage, we will explore the developer behavior patterns across GitHub and StackOverflow in this section.

A. Research Questions

Three main research questions are designed as follows:

RQ1: Do one developer's activities in GitHub reflect his activities in StackOverflow? If yes, how?

We wonder if the developer's behaviors in GitHub and StackOverflow are relevant, and if the productivity of GitHub developer is relevant to his participation in StackOverflow. For example, are active issue committers in GitHub also active question askers in StackOverflow?

RQ2: How is the relevance of developers' concerned topics between GitHub and StackOverflow?

By our experience in software development, if a developer who participates in the development of a java project will encounter a lot of java errors or problems, then he is very likely to concern on java-related contents in knowledge-sharing community. Therefore, we attempt to know how is the relevance of developers' concerned topics between GitHub and StackOverflow and if the developers' concerns in StackOverflow will shift over the time of their current participating projects in GitHub. By collecting the developers' textual contents in GitHub and StackOverflow, we will measure the similarity between their topics.

RQ3: Which kind of activities in StackOverflow is more relevant to the developer's concerns in the software development process?

We will also explore which one will be closer to developers' concerned topics in GitHub among their questions, answers and comments in StackOverflow and which activity in StackOverflow is more representative of what developers concern in the software development process.

B. Behavior Mining

Guided by the above research questions, we collect and extract following developer behavior data from GitHub and StackOverflow: the number of developers' repositories in GitHub; the number of developers' questions, answers, comments and age in StackOverflow; descriptions (or readme file) of repositories, and issue data in GitHub; and textual contents of questions, answers and comments in StackOverflow. On this data, we conduct following analysis and mining:

1) T-graph Analysis: We summarize the results of T [13], a multiple contrast test procedure using 5% family-wise error rate, by means of T-graphs [14] showed in Fig. 7. The edges correspond to the results of the pairwise comparisons and nodes to the different groups being compared in such a directed acyclic graph. For example, if A have a higher value than B for a given metric, there is an edge from A to B (A→B).

2) LDA-based Topic Clustering: We merge all one's documents before applying LDA, because a user always generate more than one document. Firstly, Each document must be

converted to a bag of-words vector because the LDA method is a bag-of-words model. For each document, Stopwords[4] such as 'is' and 'the' need to be removed. Then, we use stemming to convert words into their root form[5] and remove some low frequency (no more than 20 times in all documents) words. After that, we set these text vectors of all documents as input, and apply the LDA method to get each user's topic distribution. We use the approach designed by [15] to decide the number of topics K by experiments. Finally, we measure the similarity of topic distributions by the symmetrical KL-divergence.

3) Cross-site Tagging: The tagging systems in StackOverflow and GitHub are very different. According to the principles set forth in [12], we can use a method to mark GitHub with tags in StackOverflow based on cross-site tagging which consists of two steps: (1) Unnecessary tags removal in StackOverflow. Interestingly, we found that 20% of tags could cover all questions in StackOverflow by experiments. Therefore, we remove some low frequency tags and rewrite the remaining 80% tags by some rules [12]. (2) Tag transfer from StackOverflow to GitHub. Since all questions and answers in StackOverflow (every answer is linked to a tagged question) are marked, these tagged data are used to train a naive Bayesian model for text classification. Then each repository in GitHub is labelled and gets a tag distribution TD using this model, based on the text contents in the readme file or project instruction.

After cross-site tagging, we calculate the co-occurrence of every two labels and apply the spreading activation to get more related labels. In one iteration (Fig. 6), the spreading activation method propagates corresponding similarity to other tags with its weight. And then, we measure the similarity of the tag distribution by symmetric KL-divergence.

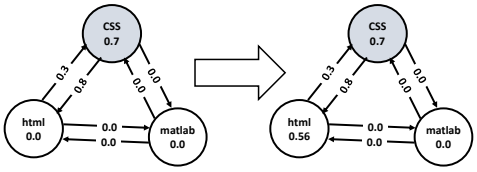

Fig. 6: An Example of Spreading Activation in One Iteration

C. Experiments and Findings

We conduct several experiments by the above technologies to answer three research questions.

For RQ1, we observe the distributions of the number of issues and questions. For each ordered pair of issues and questions, with the data sorted along one dimension, we split the other dimension into many groups and compare the distributions.

We perform experiments with our groups being quartiles. Fig. 8 shows that the most and second active 25% of the issue committers (Q1&Q2) ask more questions in StackOverflow than other quartiles (Q3&Q4), but Q1 and Q2 can not be

[4]http://www.textfixer.com/resources/commonenglish-words.txt

[5]The stemming package is in http://www.nltk.org/api/nltk.stem.html

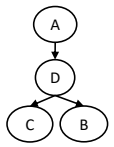

Fig. 7: T-graph

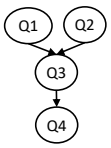

Fig. 8: Q1&Q2 Ask More Questions on SO than Others

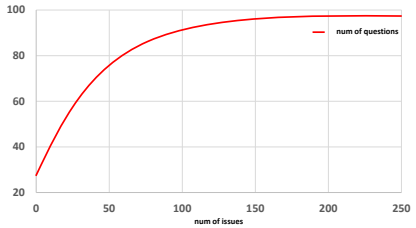

Fig. 9: Relationship Between the Num of Questions and the Num of Issues.

distinguished. This phenomenon is consistent with the result using polynomial fitting as shown in Fig .9 that active issue committers are also active question askers.

> Active issue committers are also active question askers.

For RQ2, we measure everyone's similarity of his textual contents between GitHub and StackOverflow by LDA and the tagging system. The similarities of developers are sorted from low to high, and illustrated by a fitting curve in Fig. 10. It's easy to observe in the figure that a small amount of developers' value is very low (only 0 to 0.28), but large number of developers have the similarity value from 0.3 to 0.5. The similarity between the contents of developers concerns in GitHub and StackOverflow is about 0.45. In addition, the experimental result shows that the cross-site tagging system outperforms the LDA method, because those tags maintained by StackOverflow already have a high degree of summary of the textual content.

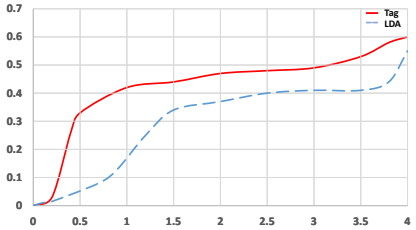

Fig. 10: The Similarity Measured by Cross-site Tagging System and LDA

> For most developers, the topics of their contents in GitHub are similar to that of their questions and answers in StackOverflow.

And then, we conduct in-depth research on RQ2. Suppose a developer participates in the project R, the readme file or description text of R is set to D_g, and the developer's direct participation in the project R starts at time T_s. Set the time interval Δt months, we obtain all questions, answers and comments of this developer in StackOverflow in $T_s \pm \Delta t$

and set to D_s. Then, the tagging system helps us to get the tag distribution for all D_s and D_g, and the KL-divergence is used to calculate the similarity between D_s and D_g. In this experiment, we set Δt to 4,6,8,10,12 and 14. And for each Δt, we simply filter out the upper and lower 10% extremum of the calculated similarities. The median value of the remaining similarity data is taken as the ordinate, and the abscissa is Δt as shown in Fig. 11. When Δt is 8 or 10, the contents that developers concerned about in GitHub and StackOverflow have the highest correlation, and the similarity is low when Δt is less than 8 or greater than 10. So, we speculate that developers will pay more attention to some of the project-related areas in a period of time.

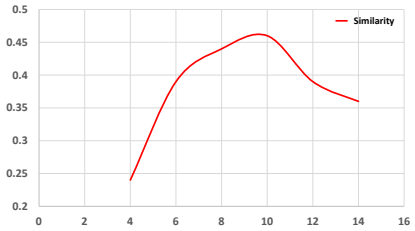

Fig. 11: The Similarity Changed With the Time of Projects

Developers' concerns in StackOverflow shift over the time of their current participating projects in GitHub.

For RQ3, we respectively compare the questions, answers, and comments of each developer in StackOverflow to the textual contents they left in GitHub. Fig. 12 shows that the textual contents that developers left in GitHub are more relevant to their answers than questions and comments in StackOverflow. In another word, the developer's answers are more representative of what developers concern in the software development process.

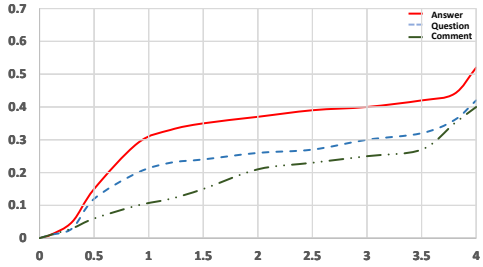

Fig. 12: Comparing Different Activities in SO to Contents in GH

The contents of developers' concerns in GitHub are more relevant to their answers than questions or comments in StackOverflow.

V. Conclusion and Future Work

In this paper, we propose a novel approach to developer behavior mining across GitHub and StackOverflow. It links the accounts from GitHub and StackOverflow, by leveraging the features from usernames, user behaviors and writing styles. Then, it mines developer behavior data across these two communities, and gains some valuable findings.

In the future, we plan to research the identity linkage problem among more than two software communities and continue to explore how StackOverflow influences GitHub, for example, what kind of issues or problem always be post to StackOverflow and what are the performance of software developers with different programming abilities in these two communities.

Acknowledgement

Beijun Shen is the corresponding author. This research is supported by National Natural Science Foundation of China (Grant No. 61472242) and 973 Program in China (Grant No. 2015CB352203).

References

[1] W. N. Robinson, T. Deng, and Z. Qi, "Developer behavior and sentiment from data mining open source repositories," in *49th Hawaii International Conference on System Sciences (HICSS)*, pp. 3729–3738, IEEE, 2016.

[2] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web?: Nier track," in *33rd International Conference on Software Engineering (ICSE)*, pp. 804–807, IEEE, 2011.

[3] B. Vasilescu, V. Filkov, and A. Serebrenik, "Stackoverflow and github: Associations between software development and crowdsourced knowledge," in *International Conference on Social computing (SocialCom)*, pp. 188–195, IEEE, 2013.

[4] M. Goeminne and T. Mens, "A comparison of identity merge algorithms for software repositories," *Science of Computer Programming*, vol. 78, no. 8, pp. 971–986, 2013.

[5] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan, "Mining email social networks," in *International Workshop on Mining Software Repositories (MSR)*, pp. 137–143, ACM, 2006.

[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 *28th IEEE International Conference on Software Maintenance (ICSM)*, pp. 592–595, IEEE, 2012.

[7] G. Pickard, W. Pan, I. Rahwan, M. Cebrian, R. Crane, A. Madan, and A. Pentland, "Time-critical social mobilization," *Science*, vol. 334, no. 6055, pp. 509–512, 2011.

[8] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "Hydra: Large-scale social identity linkage via heterogeneous behavior modeling," in *ACM SIGMOD international conference on Management of data*, pp. 51–62, ACM, 2014.

[9] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *Journal of the American society for information science and technology*, vol. 57, no. 3, pp. 378–393, 2006.

[10] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 2, p. 7, 2008.

[11] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *IEEE Symposium on Security and Privacy (SP)*, pp. 300–314, IEEE, 2012.

[12] W. Mo, B. Shen, Y. Chen, and J. Zhu, "Tbil: A tagging-based approach to identity linkage across software communities," in *Asia-Pacific Software Engineering Conference (APSEC)*, pp. 56–63, IEEE, 2015.

[13] F. Konietschke, L. A. Hothorn, E. Brunner, *et al.*, "Rank-based multiple test procedures and simultaneous confidence intervals," *Electronic Journal of Statistics*, vol. 6, pp. 738–759, 2012.

[14] B. Vasilescu, A. Serebrenik, M. Goeminne, and T. Mens, "On the variation and specialisation of workloadła case study of the gnome ecosystem community," *Empirical Software Engineering*, vol. 19, no. 4, pp. 955–1008, 2014.

[15] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.