

Exploring Identical Users on GitHub and Stack Overflow

Takahiro Komamizu, Yasuhiro Hayase, Toshiyuki Amagasa, Hiroyuki Kitagawa
University of Tsukuba, Japan

taka-coma@acm.org, {hayase, amagasa, kitagawa}@cs.tsukuba.ac.jp

Abstract

Analyzing behaviours of developers in different platforms (in particular, GitHub and Stack Overflow in this paper) can reveal interesting facts related to development activities. There are only few datasets for analysing cross-platform user behaviours, especially across GitHub and Stack Overflow. Users on GitHub and Stack Overflow are identifiable by equivalences of email addresses. In order to increase the number of identifiable users on these datasets, this paper retrieves potentially identifiable users between GitHub and Stack Overflow not relying only on email addresses. This paper employs a classification-based link prediction, which design the user identification problem as a link prediction problem on the bipartite graph consisting of users of GitHub and those of Stack Overflow. With the identification method, this paper generates a probabilistic dataset containing pairs of users with probabilities (or confidences). This paper, as well, publishes the identification tool in order to enable further data generation on appearing datasets of GitHub, Stack Overflow and others. The generated dataset and tool are highly helpful to accelerate researches on mining software repositories.

1. Introduction

Mining software repositories (or MSR) has become a largest and important research area not only in software engineering but also other data scientific research areas such as data mining and social network analysis. In particular, cross-platform analysis is a promising analytical task, however, few researches have studied is on software-related platforms. Software-related platforms including public software repository sites like GitHub¹ and Q&A sites like Stack Overflow² contain lots of knowledge related to software and programming. Many people are related to software repositories, for instance, software developers and software users, and they ask several questions on Q&A sites about software developments, software how-to, parameter

¹<https://github.com/>

²<http://stackoverflow.com/>

settings of software, etc. Therefore, analysing not only software themselves but also activities of related people in other platforms may reveal highly important and useful facts for software engineering, for instance, users' activity relationship analysis on two or more platforms, and recommending related repositories and question-answers for improving developments in projects.

However, even though cross-platform analysis is emerging topic, available datasets related to software repositories are limited due to privacy issues and other concerns. In fact, there are lots of software repositories potentially on the Web, only a few of them however are open in public in the form of datasets (e.g., database snapshots of GitHub [1] and Stack Overflow [2]). Additionally, even if such databases are available, connecting two or more software repositories for cross-platform analysis is still problematic, especially, connecting two users on the different platforms is a hard task. This is mainly because of the lack of common identities of users. Names, affiliations, occupations, etc. do not solely work well to identify users on different repositories.

This paper attempts to connect users on GitHub [1] and Stack Overflow [2] whose datasets are published in the previous MSR conferences. A straightforward and sure matching approach to connect users on GitHub and Stack Overflow is that checking equivalences of MD5-hashed user emails of GitHub and pre-hashed user emails of Stack Overflow (as discussed in [3]). The number of users matched in the above matching approach is 53760, while that of GitHub users is 499,485 and that of Stack Overflow users is 1,295,620. This indicates that only 10.763% of GitHub users and 4.149% of Stack Overflow users are matched in the above matching approach. It is highly possible that more users can be matched, of whom cannot be matched via emails. Thus, this paper attempts to enhance the matching results with matchings which are not available only by email address information.

This paper reports a development of user identifications among repositories, namely, GitHub and Stack Overflow. This research models the user identification problem as a link prediction problem [4], which can also be modeled as a classification problem determining whether a pair of users is

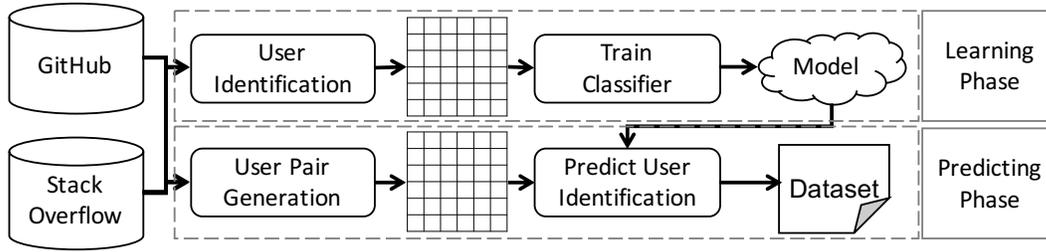


Figure 1: Overview of the proposed link prediction-based user identification. In the learning phase, user identification module processes users in GitHub and Stack Overflow datasets to identify users and generate a (user-pair \times attribute-combination) matrix for learning which consists of similarities between users in terms of attributes. The predicting phase firstly generates a (user-pair \times attribute-combination) matrix in the way analogous to the learning phase, and then trained classifiers in the learning phase predicts whether each pair of users is identical.

identical. The classification problem handles training data from [1] and [2], because users are identified by the same email addresses (this paper assumes that same users have same email addresses, and this assumption is same as [3]). The training data have only positive examples, so as to classify whether users are identical, the classification problem needs negative examples which users are surely not identical. Since there are far large number of negative examples comparing with that of positive examples, *data skewness problem* must be arisen. Data skewness problem is solved by down-sampling technique [5]. With the selected training data, this paper tries representative classification methods including linear regression [6], logistic regression [6], k-nearest neighbors [7], decision tree [8], and random forest [9], to observe which methods are feasible for user identification task. Experiments show that logistic regression, decision tree, and random forest are fairly better than other methods, and, interestingly, different weights of attributes are derives in the three methods (cf. Figure 2(a) and Figure 2(b)).

This paper mainly contributes to share the extracted datasets as well as a tool which identify users on GitHub and Stack Overflow using aforementioned identification methods. The datasets are probabilistic because of the result of statistical classification methods, however, the datasets are still useful for cross-platform analysis with handling probabilities. The user identification tool provides more possibilities to extend the datasets for published datasets of GitHub, Stack Overflow, and other software repositories. The classification methods can be replaced by appearing more sophisticated methods. The datasets and the tool are expected to accelerate research and developments related to software repository mining and other close fields.

Contributions of this paper are summarized as follows:

- **User identification mechanism** is proposed in this paper. The mechanism is inspired from classification-based link prediction methods. To realize successful user identification, this paper discusses the construction of training

data by tackling with the data skewness problem.

- **Various classification methods are examined** in order to test which classification methods are appropriate to this task. The real dataset-based experimentation realizes that best classifiers are possible to identify users with 10% cross-validation errors.
- **Datasets and a tool** for user identification are made public and customizable. Therefore, the tools can be improved by public developers. For instance, classification modules can be replaced by more sophisticated modules, and extending identification methods by adding other datasets related to software engineering and others.

2. Link Prediction-based User Identification

This paper extends the datasets of GitHub [1] and Stack Overflow [2] by a supervised learning technique (i.e., a classification-based link prediction) to probabilistically identify users in each of them. This paper models user identification problem as a link prediction problem [4] which identifies presences of links between nodes in a graph. The user identification problem on users of GitHub and those of Stack Overflow is modeled as a link prediction problem on a bipartite graph. This work realizes the link prediction based on similarities between users.

Figure 1 overviews the proposed framework for link prediction-based user identification. The framework consists of two phases, learning phase and predicting phase. The learning phase trains classifiers using obviously identifiable users who are identified by email addresses. The predicting phase identifies users using the trained models in the learning phase. The following sections explain the detail of the framework including attribute selection, similarity computations via the attributes, and a prediction methodology using existing standard classification methods.

Table 1: Selected attributes on datasets. For GitHub dataset, attributes on `users` table and descriptions of `projects` table characterize GitHub users. For Stack Overflow dataset, attributes of `users` table, question contents in `posts` table, and replies for questions in `comments` table.

Dataset	Table	Attribute	Type
GitHub	users	name	text
		location	text
		created_at	date & time
	projects	description	text
Stack Overflow	users	display_name	text
		location	text
		creation_date	date
		about_me	text
	posts	body	text
		tags	text
		title	text
	comments	comments	text

2.1. Attribute Selection on Each Dataset

The aim of this paper is to identify users on the datasets, thus user-related attributes are only necessary in the original complicated data. In GitHub data, this paper mainly uses `users` table which includes personal information registered on GitHub, and also includes project information because repository information indicate users’ interests on developments. Similarly, in Stack Overflow data, this paper uses `users` table as well as questions and answers which also indicate users’ interests on developments. Consequently, Table 1 for classification depicts selected attributes and their content types for each dataset. There are three types on the selected attributes, namely, `text`, `date & time`, and `date`.

In order to compute similarities between users, combinations of attributes are determined in a heuristic manner, and similarities for the combinations are calculated via well-studies metrics. This paper decides the combinations as shown in Table 2. Basic idea of the combinations are to combine semantically similar attributes. For instance, “users.name” in GitHub and “users.display_name” in Stack Overflow are combined because they both represent names of users, and “projects.description” in GitHub and “Stack Overflow.users.about_me” are describing interests of users, indeed, descriptions of projects related to users can represent users’ interests.

2.2. Similarity Measures

Measuring similarities for the combinations denoted above are classified into (1) similarities between textual attributes, and (2) similarities between date & time and date. For textual similarities, various similarity functions are available (e.g., edit distance and cosine similarity be-

Table 2: Combinations of attributes (b) for (user-pair × attribute-combination) matrix constructions. The combinations of attributes are selected based on similar contexts like name vs. display_name, location vs. location, description of projects vs. body of questions, etc.

Attributes on GitHub	Attributes on Stack Overflow
users.name	users.display_name
users.location	users.location
users.created_at	users.creation_date
projects.description	users.about_me
projects.description	posts.body
projects.description	posts.tags
projects.description	posts.title
projects.description	comments.comments

tween bags of words), and set-based similarity on trigram-based bag of words has achieved good matching performance. In the combinations (Table 2), this paper further divides the combinations of textual attributes into two, namely, “users.name” in GitHub and “users.display_name” in Stack Overflow, and other combinations. The similarity function for the former is cosine similarity (Equation 1) with trigram-based bag of words vectors, because names are relatively short and are same or similar in different services. That for the latter is cosine similarity (Equation 1) with TFIDF-based vectors.

$$\text{Cosine}(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|} \quad (1)$$

For the similarity between the date & time (i.e., “users.created_at” in GitHub) and date (i.e., “users.creation_date” in Stack Overflow), this paper defines a similarity function between dates by converting the date & time into date. The similarity function takes the inverse of the duration between dates, formally:

$$\text{DateSim}(\text{date}_1, \text{date}_2) = \frac{1}{|\text{date}_1 - \text{date}_2|} \quad (2)$$

2.3. Link Prediction

Given two users in GitHub and Stack Overflow, a classification-based link prediction classifies whether they will be connected in the future. There are a tremendous number of classification methods, and this paper selects fundamental classification methods (i.e., linear regression [6], logistic regression [6], k-nearest neighbors [7], decision tree [8], and random forest [9]), because they are well-known methods as well as they are available on popular machine learning libraries including scikit-learn³,

³<http://scikit-learn.org/>

Spark⁴, WEKA⁵, and Mahout⁶.

Classification is applied to a matrix which rows correspond with pairs of GitHub users and Stack Overflow users and columns correspond with combinations of attributes shown in Table 2. Elements in the matrix hold similarities of corresponding pairs of users on a combination of attributes.

For learning classifiers, training data are prepared from known identification method which is based on email addresses and their hashed values. As discussed in [3], users on GitHub and Stack Overflow are identified by the equivalences of MD5-hashed user emails of GitHub and pre-hashed user emails of Stack Overflow. The identified user pairs are used as positive examples. The negative examples are extracted based on identified users (denoted as I) by making pairs of identified users with other users. Formally, given a user $gu \in GitHub.users \cap I$ (or $su \in StackOverflow.users \cap I$), a negative example pair is (gu, nsu) where $nsu \in StackOverflow.users \setminus I$ (or (ngu, su) where $ngu \in GitHub.users \setminus I$).

2.4. Skewness Problem

Classification-based link prediction on the aforementioned training data easily fails, because of the data skewness problem. The data skewness problem is a famous problem on classification tasks, that is a classifier is trained to classify all labels as positive (or negative) when learning examples are skewed to positive (or negative) examples. This is because, in the training step of a classifier, it gets lower error rates if it classifies all examples into a major label.

In order to solve the skewness problem, this paper employs down-sampling technique [5]. Down-sampling are typical options to avoid the skewness problem, which equalizes the number of positive examples and that of negative examples. Suppose that the number of positive examples in the aforementioned dataset is M and that of negative examples is N where $M \ll N$ (in the fact of the dataset, $M = 53760$ and $N = 96.5$ billion), down-sampling performs random sampling on the negative examples in order to make $M \approx N$.

3. User Identification Quality Analysis

This section inspects the qualities of user identifications over the classification methods (i.e., linear regression, logistic regression, k-nearest neighbors, decision tree, and random forest) and discusses the balance of importance of the combinations of attributes to contribute to accurate classifications. The inspection is done with datasets of GitHub [1] and Stack Overflow [2]. The number of users in the former dataset is 499,485 and that of the latter dataset is 1,295,620.

⁴<http://spark.apache.org/>

⁵<http://www.cs.waikato.ac.nz/ml/weka/>

⁶<https://mahout.apache.org/>

In the learning phase, 53760 users are used for positive examples and 107,520 users are used for negative examples (extracted as discussed in Section 2.4).

For determining best classification methods for this task, this paper examines the five classification methods by 10-fold cross-validation of the training examples. Figure 2(a) displays average cross-validation errors of classification methods. The figure indicates that random forest (RF), logistic regression (LG), and gradient boosting decision tree (GBDT) are better accuracy than linear regression (LR) and k-nearest neighbors (kNN), thus this paper employs RF, LG, and GBDT classifiers for the tool (Section 4)⁷.

Interestingly, there are differences of learned weights for the combinations of attributes. Even though the selected classifiers achieve similar accuracy of classifications as shown in Figure 2(a), they have different preferences on the combinations. Figure 2(b) displays normalized weights (in order to make them comparable) indicating the importance of the combinations of attributes for accurate classifications. For instance, RF gives a higher weight on the combination of user names, which indicate that user names are the most important factor to identify users. For another example, GBDT indicates the combination of dates is the second most important factor. As the different weights are learning by these classifiers, they might provide different probabilistic datasets.

4. Dataset and Tool

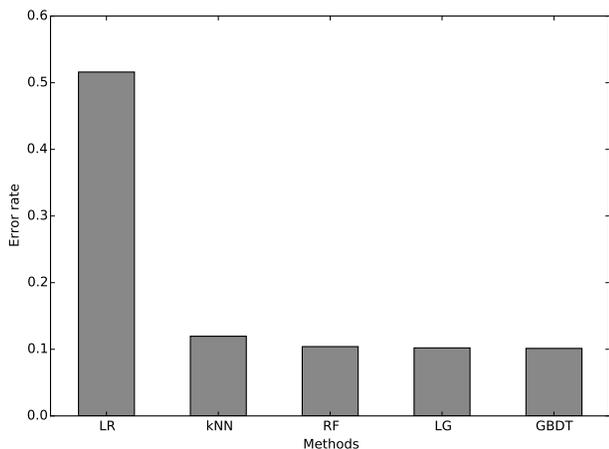
The generated dataset and generating tool are available on project page PJD_GHSO⁸ on GitHub. The dataset consists of a simple schema (namely, $\langle g_user, s_user, prob \rangle$), where g_user and s_user represent user IDs of GitHub and Stack Overflow, respectively, and $prob$ expresses the probability when these users are identical. For the first draft of published datasets, it contains 50k pairs of users classified by GBDT classifier.

The tool contains classifier learning modules and prediction modules. For the learning modules, they include similarity computation modules and learning modules. The former loads raw data to provide similarity measures for each combination of attributes on the pairs of training examples. The latter learns classifiers based on the similarities on the combinations of attributes and true labels of the pairs of users. The learned models are available in the tool and they are updatable for leaving space for improving classification accuracy. With the learned modules, prediction modules load them and predict labels for given pairs of users in GitHub and Stack Overflow. The prediction results are written in the aforementioned format.

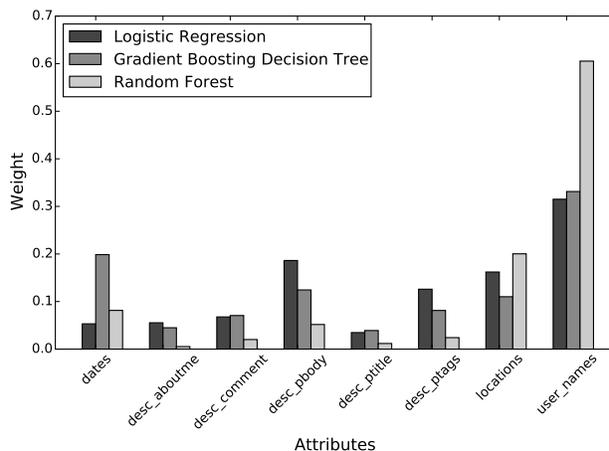
⁷The published tool includes all of the classifier models.

⁸https://github.com/Taka-Coma/PJD_GHSO

Note that the project page is yet to be designed for public but for reviewing this paper. When this paper accepted, the page includes title of this paper, conference name, and other related information.



(a) Average cross-validation errors of classification methods. Comparing five classification methods: linear regression (LR), k-nearest neighbors (kNN), random forest (RF), logistic regression (LG), and gradient boosting decision tree (GBDT). LR is significantly worse than others and kNN are the second worst, and others are close and not significantly different. Consequently, RF, LG, and GBDT are better classifiers for the task.



(b) Normalized learned weights of attributes. The better classifiers in (a) are selected. The weights indicate the importance of each combination of attributes. Interestingly, the weights for each classification (LG, GBDT, and RF) are different. For example, RF considers much higher importance on similarities between user names for accurate classification, while GBDT considers that similarities between dates are important.

Figure 2: Experimental evaluations: (a) performance comparison of classification methods, and (b) learned weights for combinations of attributes for selected classification methods.

5. Discussion – Research & Application

The proposed tool and dataset increase research opportunities related to MSR, knowledge discovery, data mining, data management, etc. Developer behaviour analysis [3] is one of the fundamental research in order to reveal behavioural facts of developers on multiple platforms (i.e., GitHub and Stack Overflow). Vasilescu et al. [3] have analyzed associations of users’ activity between GitHub and Stack Overflow. However, their work is limited to users identified by email address-based matching, thus the number of identified users are limited comparing with the proposed dataset in this paper. Therefore, they might miss some facts behind. The proposed dataset increases the opportunities of user behavioural analyses on GitHub and Stack Overflow in a probabilistic manner. Indeed, the dataset is not perfectly certain, but stochastic analyses can be used instead.

The dataset can be considered as an auxiliary dataset for applications on GitHub and Stack Overflow. For instance, repository recommendation on GitHub [10, 11, 12] has been studied to recommend relevant repositories to users. There are several ways of evaluating relevance of repositories (e.g., network analysis-based relevance, vector space similarities). For another example, user recommendation for review process on GitHub [13, 14] has been studied to smoothly managing developments. In general case of rec-

ommendation, lack of users’ activities is critical for recommendation accuracy including the cold-start problem. In such situation, joining auxiliary datasets helps enhance users’ activities on the dataset to the prior dataset, therefore, the joined dataset realizes better recommendation as well as solving the cold-start problem. Consequently, the proposed dataset acts as an auxiliary dataset to both GitHub and Stack Overflow datasets, and is expected to improve accuracy of applications.

6. Related Work

This paper aims at finding identical users on different platform, and this paper is, in the best of our knowledge, the first work of exploring identical users on GitHub and Stack Overflow and publishing datasets and identification tools. Wang et al. [15] have studied the user identification problem on different sites. Their approach only relies on user names by taking variations (like abbreviation) of user names into account for similarity computations between users on different platforms. They show that even only user names, moderate user identification accuracies can be achieved. While, this paper takes possible attributes into account to improve user identification accuracy, and the learned weights in Figure 2(b) indicate other factors are also important to identify users in different platforms. Motoyama et al. [16] gather attributes as sets of words and cal-

culate similarities among users based of the sets. They consider all attributes equally, however, this paper distinguishes attributes and classifiers take different importances for different combinations of attributes. Zhou et al. [17] have also studied user identification basically on social media network platforms like Twitter⁹, Sina Microblog¹⁰, Facebook¹¹, and RenRen¹². They rely on topologies of social media networks, so their approach is not much applicable to datasets on GitHub and Stack Overflow. This is because users on GitHub (resp. Stack Overflow) are connected lesser than those on microblog-based social media networks. Zheng et al. [18] and Kong et al. [19] have been proposed content-based user identification. Zheng et al. [18] have identified users by their writing styles of messages on social media networks. The writing styles for individuals on GitHub and Stack Overflow are not surely evidential for identifying users between them, due to the differences of what to write as contents. While, Kong et al. [19] have attempted to identify users with spatio, temporal and textual similarity measurements with assumption of location-based social media networks. However, GitHub and Stack Overflow are not location-based services, therefore, their approach is not applicable.

7. Conclusion

This paper reports a user identification tool between GitHub users and Stack Overflow users by applying classification-based link prediction methods, and publishes a dataset of identified users with probabilities. This dataset expands the originally identified users on datasets of GitHub and Stack Overflow with email address-based identification. The proposed tool classifies with 10% errors in learning process.

This paper suggests two future works for improving the proposed tool in terms of quality and performance. There are several ideas for improving the quality: (1) ontological similarities on geographical attributes (i.e., locations) can provide more appropriate similarities on geographical attributes, and (2) more sophisticated classification and regression methods can improve the classification accuracy. The performance can be improved by two folds, scalability and storage. The learning module and predicting module should become faster by parallel and distributed computing (e.g., Hadoop, Spark or GPGPU). Because of large number of possible user combinations, large storage is necessary even for intermediate results.

⁹<http://www.twitter.com>

¹⁰<http://www.weibo.com>

¹¹<https://www.facebook.com/>

¹²<http://www.renren.com>

Acknowledgement

This research was partly supported by the program *Research and Development on Real World Big Data Integration and Analysis* of RIKEN, Japan.

References

- [1] G. Gousios, "The GHTorrent Dataset and Tool Suite," in *MSR 2013*, 2013, pp. 233–236.
- [2] A. Bacchelli, "Mining Challenge 2013: Stack Overflow," in *MSR 2013*, 2013.
- [3] B. Vasilescu, V. Filkov, and A. Serebrenik, "StackOverflow and GitHub: Associations between Software Development and Crowd-sourced Knowledge," in *SocialCom 2013*, 2013, pp. 188–195.
- [4] D. Liben-Nowell and J. Kleinberg, "The Link-Prediction Problem for Social Networks," *Journal of the Association for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
- [5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *JAIR*, vol. 16, pp. 321–357, 2002.
- [6] D. A. Freedman, *Statistical Models: Theory and Practice*. Cambridge University Press, 2009.
- [7] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Trans. Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [8] J. H. Friedman, "Stochastic Gradient Boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.
- [9] T. K. Ho, "Random Decision Forests," in *ICDAR 1995*, 1995, pp. 278–282.
- [10] M. Guendouz, A. Amine, and R. M. Hamou, "Recommending Relevant Open Source Projects on GitHub using a Collaborative-Filtering Technique," *IJOSSP*, vol. 6, no. 1, pp. 1–16, 2015.
- [11] L. Zhang, Y. Zou, B. Xie, and Z. Zhu, "Recommending Relevant Projects via User Behaviour: An Exploratory Study on Github," in *CrowdSoft 2014*, 2014, pp. 25–30.
- [12] T. Matek and S. T. Zebec, "GitHub open source project recommendation system," *CoRR*, vol. abs/1602.02594, 2016.
- [13] M. M. Rahman, C. K. Roy, and J. A. Collins, "CORRECT: Code Review Recommendation in GitHub Based on Cross-Project and Technology Experience," in *ICSE 2016*, 2016, pp. 222–231.
- [14] Y. Yu, H. Wang, G. Yin, and T. Wang, "Reviewer recommendation for pull-requests in GitHub: What can we learn from code review and bug assignment?" *Information & Software Technology*, vol. 74, pp. 204–218, 2016.
- [15] Y. Wang, T. Liu, Q. Tan, J. Shi, and L. Guo, "Identifying Users across Different Sites using Usernames," in *ICCS 2016*, 2016, pp. 376–385.
- [16] M. A. Motoyama and G. Varghese, "I Seek You: Searching and Matching Individuals In Social Networks," in *WIDM 2009*, 2009, pp. 67–75.
- [17] X. Zhou, X. Liang, H. Zhang, and Y. Ma, "Cross-Platform Identification of Anonymous Identical Users in Multiple Social Media Networks," *IEEE TKDE*, vol. 28, no. 2, pp. 411–424, 2016.
- [18] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: Writing-style features and classification techniques," *JASIST*, vol. 57, no. 3, pp. 378–393, 2006.
- [19] X. Kong, J. Zhang, and P. S. Yu, "Inferring Anchor Links across Multiple Heterogeneous Social Networks," in *CIKM 2013*, 2013, pp. 179–188.