Procedia Computer Science

Volume 80, 2016, Pages 376–385

ICCS 2016. The International Conference on Computational Science

Identifying Users across Different Sites using Usernames

Yubin Wang[1,2], Tingwen Liu[1,2,*], Qingfeng Tan[1,2], Jinqiao Shi[1,2], and Li Guo[1,2]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] National Engineering Laboratory for Information Security Technologies, Beijing, China
{wangyubin, liutingwen, tanqingfeng, shijinqiao, guoli}@iie.ac.cn

Abstract

Identifying users across different sites is to find the accounts that belong to the same individual. The problem is fundamental and important, and its results can benefit many applications such as social recommendation. Observing that 1) usernames are essential elements for all sites; 2) most users have limited number of usernames on the Internet; 3) usernames carries information that reflect an individual's characteristics and habits *etc.*, this paper tries to identify users based on username similarity. Specifically, we introduce the self-information vector model to integrate our proposed content and pattern features extracted from usernames into vectors. In this paper, we define two usernames' similarity as the cosine similarity between their self-information vectors. We further propose an abbreviation detection method to discover the initialism phenomenon in usernames, which can improve our user identification results. Experimental results on real-world username sets show that we can achieve 86.19% precision rate, 68.53% recall rate and 76.21% F1-measure in average, which is better than the state-of-the-art work.

Keywords: user identification, username similarity, self-information model, abbreviation detection

1 Introduction

Identifying users across different sites, which tries to find the accounts that belong to the same individual, is a fundamental and important problem. This work can be applied in many applications, such as user profiling and personalized recommendation. Given a targeted user, ArnetMiner [15] enriches the user's profile by integrating the information extracted from the corresponding accounts elsewhere. In [11], users' auxiliary information on Twitter are exploited to address the typical problems in single network-based recommendation solutions to recommend YouTube video.

In this paper, we focus on addressing the important problem using usernames, owing to following three reasons. First, usernames are essential elements for all sites, while user attributes and social behaviors do not exist in some sites or hard to collect for researchers. In this case, prior identification approaches [2, 10, 5, 15] designed for social networks do not work well. Even

*Tingwen Liu is the corresponding author of this paper.

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.336

if all user attributes and social behaviors needed are available, our work is still valuable as it can be used to improve prior social graph based approaches. Second, most users have limited number of usernames on the Internet, and these usernames usually have the same or similar naming rules. Because it is hard for users to memory too many different and casual usernames. Third, usernames may also reflect the characteristics and habits of an individual. For example, username `shmilyszw` in CSDN consists of `shmily` (an abbreviation of "See How Much I Love You") and `szw` (probably an abbreviation of someone's name).

For two given usernames, this paper tries to determine whether they belong to the same individual based on username similarity and username abbreviation. Username similarity is intended to define how much similar the two usernames are, and username abbreviation is to check if one username (or its substrings) is an initialism of the other username (or its substrings). We assume that two usernames with high similarity and initialism phenomenon are very likely to belong to the same individual.

Distance metrics, such as Levenshtein distance, are intuitive and easy-to-implement tools to quantify username similarity. However, they are not the best choice. Because a username usually consists of multiple relatively independent parts, while these distance metrics do not consider the permutation of the username parts. This paper introduces the self-information model to quantify the similarity between usernames. We extract 1296 content features and 77 pattern features for each username, which are integrated as a vector by the self-information model with the self-information of each feature as its weight. Then we quantify the similarity of any two given usernames as the cosine similarity between their self-information vectors.

We reduce the problem of detecting the initialism phenomenon into the problem of getting the minimum number of meaningless characters for each username. A meaningless character is the one that is not a member of any word in a given username after splitting the usernames to get some non-overlapped words. Note that there may be multiple different ways to split a username. The problem is NP-hard and addressed in this paper based on the dynamic programming strategy.

We make three key contributions in this paper. First, we quantify username similarity based on the self-information vector model and our proposed content features and pattern features. Second, we propose a dynamic programming algorithm to detect the initialism phenomenon between usernames. Third, we conduct experiments on real-world username sets and validate the effectiveness of our work.

2 Related Work

Prior work on identifying users across different sites can be divided into three categories: user attribute based approaches, social graph based approaches and hybrid approaches.

User attribute based approaches [12, 8, 13, 4, 14] are designed for these sites where social network structures are unavailable. As a result, we could only obtain and leverage the attributes of users, especially usernames, to identify users across different sites. Perito *et al.* [8] used a 5-gram Markov Chain model to compute the username observation likelihood as the estimation the uniqueness of the username. Their work is limited to only using this single feature to link different usernames. Zafarnai *et al.* [13] extended this work and conducted a more in-depth analysis of the features of the usernames. They proposed the methodology MOBIUS to model the features of usernames according to the users' behavioral patterns when creating usernames and employed machine learning for effective identification. Then Zafarnai *et al.* [14] generalized their work in [12, 13], and give a further detailed discussion on the problem of user identification across social media. Our work is a user attribute based approach, that identifies user across

different sites using only usernames. Thus, our work has the widest applicability, and it can be used to improve the results of prior identification work.

We sometimes have to identify users only based on social network structures for some types of sites and applications, such as on anonymous communication systems. And social graph based approaches [1, 6, 7, 10] are well designed for these sites. Narayanan and Shmatikov [7] presented a framework for analyzing privacy and anonymity in social networks and developed a new re-identification algorithm targeting anonymized social network graphs. They successfully de-anonymized several thousand users in the anonymous graph of Twitter using another social network Flickr as the source of auxiliary information. Tan *et al.* [10] reviewed the user mapping task in [7] as a potential manifold alignment problem across social structures. They built a hypergraph to model relations and proposed a manifold alignment algorithm to rank all users in the other network by their possibilities of being the corresponding user.

There are also some hybrid approaches[2, 5, 15] on identifying users across different sites, which address the problem by considering both user attributes and social network structures. Cui *et al.* [2] studied the problem of finding email correspondents in social networks. Their approach integrated similarity between profiles and communication networks. Liu *et al.* [5] proposed a multi-objective learning framework HYDRA which incorporated user attributes, user generated content and social social network structure to link user accounts in different social networks. They handled the missing information among social data associated with a user that most methods did not consider. The above methods considered only local consistency of account pairs between two sites, and Zhang *et al.* [15] argued that the global consistency among multiple network is also important. Thus, they considered both of the local and global consistency, and built an energy-based model to connect heterogeneous social networks.

3 Problem Statement

When creating accounts, people need to choose one username as the unique identification to sign in a site in the future. Usernames are restricted to only consisting of alphanumeric characters and some special characters, such as dot (.), hyphen (-) and underscore (_). The lengths of usernames are usually in the range of 4 and 20.

As for identifying users across different sites, there are two general problems needed to be addressed, namely decision problem and searching problem.

Decision Problem: given two usernames u and v on two different sites, determine whether the two usernames belong to the same individual.

$$D(u,v) = \begin{cases} 1 & \text{if } u \text{ and } v \text{ belong to the same individual} \\ 0 & \text{otherwise} \end{cases}$$

Searching Problem: given a username u and a set of usernames V on another site, find all usernames in V that belong to the same individual with u.

$$S_u(V) = \{v \mid v \in V,\ D(u,v) = 1\}$$

This paper aims at addressing the first problem, namely the decision problem. The searching problem can be reduced to the decision problem by checking every member in V linearly. We will propose some methods to accelerate the searching process in our future work.

	moon	mood	ben
Feature 1: starting with "mo" ?	*Yes*	*Yes*	*No*
Feature 2: ending with "n" ?	*Yes*	*No*	*Yes*
Feature 2: containing "oo" ?	*Yes*	*Yes*	*No*
Feature 4: less than 4 characters ?	*No*	*No*	*Yes*

moon → ⟨*Yes, Yes, Yes, No*⟩ → ⟨1, 1, 1, 0⟩
mood → ⟨*Yes, No, Yes, No*⟩ → ⟨1, 0, 1, 0⟩
ben → ⟨*No, Yes, No, Yes*⟩ → ⟨0, 1, 0, 1⟩

Figure 1: An example of representing usernames as binary vectors.

4 Our User Identification Approach

As mentioned above, two usernames with high similarity and initialism phenomenon are very likely to belong to the same individual. In this paper, we simply determine that two usernames belong to the same individual if the similarity between them is higher than a predefined threshold, or one username is an abbreviation of the other, shown as follows:

$$D(u,v) = \begin{cases} 1 & \text{sim}(u,v) \geq \tau \text{ or } \text{abbr}(u,v) = 1 \\ 0 & \text{otherwise} \end{cases}$$

where $\text{sim}(u,v)$ indicates the similarity between u and v, τ is the predefined threshold, and $\text{abbr}(u,v) = 1$ if there is a initialism phenomenon between u and v.

Next we will give a detailed description of the assignment processes of $\text{sim}(u,v)$ and $\text{abbr}(u,v)$ respectively. We first introduce the self-information vector model of integrating features to quantify the similarity between usernames. Then we describe our content features and pattern features that are extracted from usernames and used in the self-information vector model. At last we give our dynamic algorithm to discover the initialism phenomenon between usernames.

4.1 Self-information Vector Model

The self-information vector model is used to integrate multiple features extracted from each username into a vector, then the problem of quantifying the similarity between two usernames is translated into the calculation of similarity between their self-information vectors. Here we do not use the Levenshtein distance to quantify the similarity between usernames, because a username usually consists of multiple relatively independent parts, while Levenshtein distance does not consider the permutation of the username parts.

Intuitively, two usernames with more common features are more similar. As shown in Figure 1, `moon` is more similar with `mood`, comparing with `ben`, as `moon` and `mood` have two out of four common features, while `ben` and `mood` have no common feature.

Before giving our self-information vector model, we first introduce a feature indicator function used in the model.

Feature Indicator Function: given a feature λ and a username u, feature indicator function $\mathcal{I}_\lambda(u)$ is defined to indicate that whether u satisfies feature λ:

$$\mathcal{I}_\lambda(u) = \begin{cases} 1 & \text{if } u \text{ satisfies the feature } \lambda \\ 0 & \text{otherwise} \end{cases}$$

Then if we extract m features $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ from username u, then we can construct a binary vector $\mathcal{B}_u$ of m members for u:

$$\mathcal{B}_u = \langle \mathcal{I}_{\lambda_1}(u), \mathcal{I}_{\lambda_2}(u), \ldots, \mathcal{I}_{\lambda_m}(u) \rangle$$

As shown in Figure 1, usernames `moon`, `mood` and `ben` can be represented as $\langle 1,1,1,0 \rangle$, $\langle 1,0,1,0 \rangle$ and $\langle 0,1,0,1 \rangle$. Then we can get the number of common features between two usernames by counting the number of positions in two binary vectors both with value 1.

Note that for binary vectors, all features have the same importance. However, it is not invariably suitable. Because giving the same importance to each feature does not distinguish each feature's ability of describing usernames, and does not consider the interaction between different features. For example, for two given features "starting with `abc`" (denoted as λ_1) and "starting with `abcdef`" (denoted as λ_2), obviously feature λ_2 should be given more importance than λ_1. Because feature λ_2 carries more information, and any username that satisfies feature λ_2 must also satisfies λ_1.

How to assign weights for different features become an open problem. In this paper, we use each feature's self-information as its weight. Self-information is derived by Shannon [9] to measure the quantities of information. The self-information of a specific message m is defined as $I(m) = -\log \Pr(m)$. Then we can represent each username as a self-information vector.

Self-information Vector: given m features $\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$, we can represent username u as a self-information vector $\mathcal{V}_u$:

$$\mathcal{V}_u = \langle \mathcal{I}_{\lambda_1}(u) \cdot W(\lambda_1), \mathcal{I}_{\lambda_2}(u) \cdot W(\lambda_2), \ldots, \mathcal{I}_{\lambda_m}(u) \cdot W(\lambda_m) \rangle$$

where

$$W(\lambda) = I(\mathcal{I}_\lambda(\cdot) = 1) = -\log \Pr(\mathcal{I}_\lambda(\cdot) = 1)$$

We also note that it is impossible to get the true value of $\Pr(\mathcal{I}_\lambda(\cdot) = 1)$. However, we can get an estimated value of $\Pr(\mathcal{I}_\lambda(\cdot) = 1)$ on username set U using the following way:

$$\hat{\Pr}(\mathcal{I}_\lambda(\cdot) = 1) = \frac{|\{u \in U \mid \mathcal{I}_\lambda(u) = 1\}|}{|U|}$$

As each username is represented as a weighted vector, we can use the cosine similarity to quantify the similarity between usernames. Cosine similarity which is widely used in information retrieval to measure the similarity between vectors. Then the similarity between two given usernames u and v can be calculated as:

$$\mathrm{sim}(u, v) = \cos(\mathcal{V}_u, \mathcal{V}_v) = \frac{\mathcal{V}_u \cdot \mathcal{V}_v}{\|\mathcal{V}_u\| \cdot \|\mathcal{V}_v\|}$$

4.2 Username Feature Used in Self-information Vector Model

Before extracting features from usernames, we conduct two necessary steps on usernames, namely converting usernames in lowercase and removing special characters. This is because usernames are not case sensitive on most sites, and special characters usually act as delimiters for the username. Taking special characters into account will increase the complexity of our model and disregarding them would give the same results. This paper aims at extracting features from usernames by two aspects, namely the content of usernames and the patterns when creating usernames. Table 1 shows an overview of the features used in this paper.

Table 1: Information of features used in this paper

Category	Sub-Category	# of Features
Content Features	n-gram	1296
Pattern Features	LD-permutation Pattern Feature	6
	LD-gram Pattern Feature	64
	Date Pattern Feature	4
	Keystroke Pattern Feature	3

Content Feature: we construct some features in the form of "containing a specific n-gram" to represent the content of a username. In this way, we can reduce the impact of different permutations of the relatively independent parts in usernames. Note that the value of n should be neither too big or too small. Value n should not be too big, as the lengths of usernames are limited, and big n will generate too many feautres. Finally the self-information vectors are very sparse. Value n should also not be too small, as grams with small lengths will appear frequently in many usernames. Experimentally, we set $n = 2$ to achieve better results. Therefore, we extract $(26 + 10)^2 = 1296$ content features, where 26 is the number of letters, 10 is the number of digits, and 2 is the value of n.

Pattern Feature: all the 76 pattern features can be further divided into four categories, namely LD-permutation pattern feature, LD-gram pattern feature, date pattern feature and keystroke pattern feature.

LD-permutation pattern feature: Observing that each username consists of continuous letters and digits, there are six L(etter)D(igit)-permutation pattern features if limiting the number of continuous letters and digits is not more than 3: "only letters", "only digits", "letters + digits", "digits + letters", "letters + digits + letters" and "digits + letters + digits". There are very few users with more than 3 continuous letters and digits.

LD-gram pattern feature: this type of pattern features aims at making a quantitative analysis of the conversion between letters and numbers. we transform each username to a string with only L and D. For example, username `niudan1986` can be transformed to the string `LLLLLLDDDD`. Then we construct features in the form of "containing a specific m-gram" to represent the conversion between letters and numbers. We experimentally set $m = 6$ to get better performance. Finally, we can get $2^6 = 64$ LD-gram pattern features.

Date pattern feature: many usernames include a variety of dates, such as date of birth. Considering different date formats appear in a username or not, we can get some date pattern features. This paper only focuses on the widely-used four date formats, namely "year + month + day", "month + day + year", "day + month + year" and "month + day". Then we can get 4 date pattern features.

Keystroke pattern feature: different people may have different keystroke ways when creating a username. Keystroke patterns may be useful to identify users. As observed in our collected data, usernames `adsaasdasd` and `fdfdfffd` belong to the same individual, because they are created by repeatedly clicking some adjacent

Algorithm 1: Optimal Username Split Algorithm

Input: A username u and a set of words $\mathcal{W}$.
Output: The optimal split of the username u.

$n := Length(u)$;
$Next := [1...n]$;
for $i := n;\ i \geq 1;\ i--$ **do**
  $Next[i] := i + 1$;
  foreach $w \in \mathcal{W}$ *which appears in* u *where starting from the* i^{th} *position* **do**
    $m := Length(w)$;
    if *cutting off* $u_i u_{i+1} \cdots u_{i+m-1}$ *is local optimal* **then**
      $Next[i] := i + m$;
$\mathcal{S}_u^* = \{\}$;
for $i := 1;\ i \leq n;\ i := Next[i]$ **do**
  $p := Next[i]$;
  $\mathcal{S}_u^* := \mathcal{S}_u^* \cup u_i u_{i+1} \cdots u_{p-1}$;
return $\mathcal{S}_u^*$;

keys on a QWERTY keyboard. We assign a coordinate to each character on keyboards and extract three keystroke pattern features for each username: 1) whether all characters are in the same row on keyboard, *e.g.* `asdfhj`; 2) whether each two consecutive characters in the username are adjacent on keyboards, but not in the same row, *e.g.* `qawsxd`; 3) whether each character in the username is the same with or adjacent to the next one on keyboard, *e.g.* `asefcc`. The three keystroke patterns are inspired by the work of a large-scale empirical analysis of web passwords [3].

4.3 Abbreviation Detection

To detect the initialism phenomenon, we need to split usernames first. In this paper, a split of username u is a ordered list of non-empty strings $\mathcal{S}_u = \{s_1, s_2, \ldots, s_n\}$ such that $u = s_1 s_2 \cdots s_n$. And a character in u is regarded as meaningless with respect to the split $\mathcal{S}_u$ if the character is one member of $\mathcal{S}_u$. Specific to the detection of the initialism phenomenon, we further restrict that split $\mathcal{S}_u$ must only consist of words and meaningless characters. We define that $\text{abbr}(u, v) = 1$ if there are at least q consecutive single characters in the split of one username that are the initials of the q consecutive words in the split of the other username.

How to split usernames is an open question. In this paper, our goal is to split a username with the minimum number of meaningless characters. In this way, we can get some semantic information carries in usernames as much as possible. We design a dynamic programming algorithm as shown in Algorithm 1 to split a username.

We experimentally set $q = 2$ to get better performance. For example, for usernames `zgxxidian123` and `zhangguoxin012` in our collected data, word set $\mathcal{W}$ as a set of Pinyin words, the split of username `zgxxidian123` is {`z`, `g`, `x`, `xi`, `dian`, `1`, `2`, `3`}, and the split of `zhangguoxin012` is {`zhang`, `guo`, `xin`, `0`, `1`, `2`}. We can find that `zgx` in `zgxxidian123` is a string consisting of the initials of three consecutive words `zhang`, `guo` and `xing` in the split of username `zhangguoxin012`. So `zgx` is an abbreviation of `zhangguoxin`, and there is an initialism phenomenon between `zgxxidian123` and `zhangguoxin012`, namely

abbr(zgxxidian123, zhangguoxin012) = 1. Obviously these two usernames are very likely to belong to the same individual.

5 Experimental Results

5.1 Experimental Setup

We collected three account sets, namely CSDN, 17173 and 178, which are leaked in Dec, 2011[1]. The CSDN website is one of the biggest communities of software developers in China, which also provides online forums, blog hosting and IT news services. The CSDN set consists of 6,302,988 records containing usernames and email addresses. The 17173.com site is a leading online media site and value-added information service provider for Chinese game players. It also provides online player communities, game downloading and online game video services. The 17173 set consists of 2,500,264 records, which also contain usernames and email addresses. The 178.com is one of most popular game portals in China. Similar to the 17173.com site, the 178.com site also provides game information, game downloading, and webpage game services. The 178 set consists of 3,827,603 records, which also contain usernames and email addresses.

For two records in different sets of CSDN, 17173 and 178, if they have the same email address, we think that the two records are created and used by the same individual. Thus, the corresponding two usernames constitute a positive instance used to validate the effectiveness of our work on identifying users using only usernames. Based on this assumption, we find that there are 155,878 positive instances between CSDN set and 17173 set, 112,603 positive instances between CSDN set and 178 set, and 145,849 positive instances between 17173 set and 178 set. We also construct the same number of negative instances as the positive instances of collected sets. Each negative instance is extracted from two randomly selected records with different email addresses. Finally, we get three experimental sets: CSDN+17173, CSDN+178, and 17173+178.

5.2 Effects of Our Work

We evaluate the effectiveness of our work on identifying users with three metrics: precision rate, recall rate and F_1-measure. When our work correctly predicts a positive instance, it is a true positive (TP). When our work correctly predicts a negative instance, it is a true negative (TN). When our work predicts a negative instance as a positive instance, it is a false positive (FP). When our work predicts a positive instance as a negative instance, it is a false negative (FN). The precision rate is defined as the proportion of predicted positive instances that are really positive, which is $PR = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$. The recall rate is the proportion of real positive instances that are correctly predicted, which is $RR = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$. The F_1-measure is the harmonic mean of the precision rate and the recall rate, and defined as $\frac{2 \cdot PR \cdot RR}{PR+RR}$.

Figure 2 shows the change of the precision, recall and F_1-measure of our work for different threshold τ. As shown in Figure 2, we can get high F_1-measure when $0.1 \leq \tau \leq 0.2$. Thus, we use 0.15 as the optimal value of τ to achieve good identification results in this paper.

In this paper, we compare our work with prior methods on identifying users using only usernames. To the best of our knowledge, MOBIUS [13, 14] is the state-of-the-art method which tackles the same problem. MOBIUS models the users' behavior patterns when creating usernames, such as username length likelihood and unique username creation likelihood. It employs machine learning to learn a binary classifier for identification. We newly extract the

[1] http://en.people.cn/90778/7688084.html

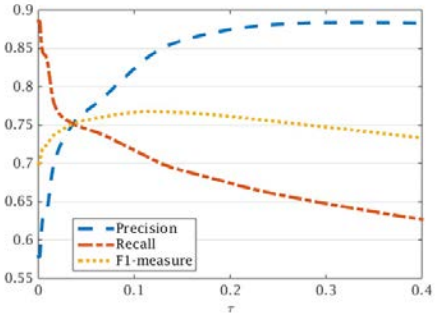

Figure 2: Change of precision, recall and F_1-measure for different threshold τ.

Table 2: Information of features used in this paper

Experimental sets	Name-Match			MOBIUS			Our Work		
	prec	rec	F_1	prec	rec	F_1	prec	rec	F_1
CSDN+17173	82.93	47.43	60.35	86.26	73.02	79.09	85.80	76.63	**80.96**
CSDN+178	83.56	32.92	47.23	84.61	64.73	73.35	87.23	64.68	**74.28**
17173+178	82.08	33.11	47.19	84.67	62.86	72.15	85.54	64.27	**73.39**
Average	82.86	37.82	51.59	85.18	66.87	74.86	86.19	68.53	**76.21**

LD pattern, date pattern and keystroke pattern of usernames which are not considered in MOBIUS. Our approach quantifies the similarity between usernames and it can be more easily extended into other methods compared with learning a binary classifier. We also devise a baseline method denoted as Name-Match for comparison, which is based on the assumption that two usernames belong to the same individual if and only if they are exactly the same.

We conduct ten-fold cross validation on each experimental set. Table 2 shows the precision, recall and F_1-measure of Name-Match, MOBIUS and our work on identifying users. As shown in Table 2, we can find that our work achieves 86.19% precision rate, 68.53% recall rate and 76.21% F_1-measure in average, which is better than the state-of-the-art work. Specifically, in terms of F_1-measure, our work achieves about $(76.21 - 51.59)/51.59 \approx 48\%$ improvement over the baseline method Name-Match and achieves about $(76.21 - 74.86)/74.86 \approx 1.8\%$ improvements over MOBIUS. We also note that the results of MOBIUS on our datasets are not as good as the results in [13, 14]. It is probably because the features they used may not apply to our datasets.

6 Conclusion and Future Work

In this paper, we studied the problem of identifying users across different sites using only usernames. Based on the assumption that usernames with high similarity and initialism phenomenon belong to the same individual, we proposed a self-information vector model to quantify the similarity and also proposed a dynamic programming algorithm to detect the initialism phenomenon between usernames. Experimental results on real-world username sets showed that we can achieve 86.19% precision rate, 68.53% recall rate and 76.21% F1-measure in average, which is better than the state-of-the-art work.

Extensions to our work can be expanded from the following three aspects. First, we will design more probable and appropriate features from usernames to further improve the results of our work in future. Second, we plan to design an efficient method to address the searching problem, namely speed up the process of searching similar usernames on large-scale username sets. Third, we will extend our work to apply in prior approaches, such as user attribute based approaches and hybrid approaches, to observe its effectiveness on improving results.

Acknowledgement

This work was supported in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA06030200.

References

[1] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore Art Thou R3579X?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In *Proc. of WWW*, pages 181–190, 2007.

[2] Yi Cui, Jian Pei, Guanting Tang, Wo-Shun Luk, Daxin Jiang, and Ming Hua. Finding Email Correspondents in Online Social Networks. *World Wide Web*, 16(2):195–218, 2013.

[3] Zhigong Li, Weili Han, and Wenyuan Xu. A Large-scale Empirical Analysis of Chinese Web Passwords. In *Proc. of USENIX Security*, pages 559–574, 2014.

[4] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a Name?: An Unsupervised Approach to Link Users across Communities. In *Proc. of ACM WSDM*, pages 495–504, 2013.

[5] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. Hydra: Large-scale Social Identity Linkage via Heterogeneous Behavior Modeling. In *Proc. of ACM SIGMOD*, pages 51–62, 2014.

[6] Arvind Narayanan and Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *Proc. of IEEE S&P*, pages 111–125, 2008.

[7] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing Social Networks. In *Proc. of IEEE S&P*, pages 173–187, 2009.

[8] Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, and Pere Manils. How Unique and Traceable Are Usernames? In *Proc. of PETS*, pages 1–17, 2011.

[9] Claude Elwood Shannon. A Mathematical Theory of Communication. *Bell System Technical*, 27:379–423, 1948.

[10] Shulong Tan, Ziyu Guan, Deng Cai, Xuzhen Qin, Jiajun Bu, and Chun Chen. Mapping Users across Networks by Manifold Alignment on Hypergraph. In *Proc. of AAAI*, pages 159–165, 2014.

[11] Ming Yan, Jitao Sang, and Changsheng Xu. Unified Youtube Video Recommendation via Cross-network Collaboration. In *Proc. of ACM ICMR*, pages 19–26, 2015.

[12] Reza Zafarani and Huan Liu. Connecting Corresponding Identities across Communities. In *Proc. of ICWSM*, pages 354–357, 2009.

[13] Reza Zafarani and Huan Liu. Connecting Users across Social Media Sites: A Behavioral-modeling Approach. In *Proc. of ACM SIGKDD*, pages 41–49, 2013.

[14] Reza Zafarani, Lei Tang, and Huan Liu. User Identification across Social Media. *ACM Transactions on Knowledge Discovery from Data*, 10(2):16, 2015.

[15] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, and Philip S Yu. COSNET: Connecting Heterogeneous Social Networks with Local and Global Consistency. In *Proc. of ACM SIGKDD*, pages 1485–1494, 2015.