Other Times, Other Values:
Leveraging Attribute History to Link User Profiles across Online Social Networks

Paridhi Jain[†], Ponnurangam Kumaraguru[†], Anupam Joshi*
[†]Indraprastha Institute of Information Technology (IIIT-Delhi), India
*University of Maryland, Baltimore County (UMBC), USA
{paridhij, pk}@iiitd.ac.in, joshi@cs.umbc.edu

ABSTRACT

Profile linking is the ability to connect profiles of a user on different social networks. Linked profiles can help companies like Disney to build psychographics of potential customers and segment them for targeted marketing in a cost-effective way. Existing methods link profiles by observing high similarity between most recent (current) values of the attributes like name and username. However, for a section of users observed to evolve their attributes over time and choose dissimilar values across their profiles, these current values have low similarity. Existing methods then falsely conclude that profiles refer to different users. To reduce such false conclusions, we suggest to gather rich history of values assigned to an attribute over time and compare attribute histories to link user profiles across networks. We believe that attribute history highlights user preferences for creating attribute values on a social network. Co-existence of these preferences across profiles on different social networks result in alike attribute histories that suggests profiles potentially refer to a single user. Through a focused study on *username*, we quantify the importance of username history for profile linking on a dataset of real-world users with profiles on Twitter, Facebook, Instagram and Tumblr. We show that username history correctly links 44% more profile pairs with non-matching current values that are incorrectly unlinked by existing methods. We further explore if factors such as longevity and availability of username history on either profiles affect linking performance. To the best of our knowledge, this is the first study that explores viability of using an attribute history to link profiles on social networks.

1. INTRODUCTION

Today, Online Social Networks (OSNs) offer innovative services that ease the access to news, campaigns, art, talent, business opportunities and personal connections. For instance, Twitter's retweet feature enables quick access to news, campaigns, and crisis information while pin boards of Pinterest facilitate reach to the work of artists, photographers, and fashion designers. In order to enjoy these services simultaneously, a user innocuously registers herself on multiple OSNs. During registration, she shares personal infor-

mation, lists her friends and later creates content to share with her friends. The quality, quantity and veracity of the information created and shared by her vary with the OSN, thereby resulting in dissimilar profiles of the same user, scattered on the world wide web, with no explicit links directing to one another. These disparate profiles liberate her from any privacy concerns that could emerge if the profiles were implicitly collated. However, linking these disparate unlinked profiles can benefit various stakeholders.

Companies like Disney and PepsiCo carry out psychographic segmentation based upon customers' activities, interests, opinions and lifestyles to adapt marketing strategies on their needs [2]. It is the most effective segmentation citing a rise of 24% in business performance [27]; however includes high cost in both time and money [8]. The cost of constructing psychographics of each customer can be brought down with the use of her linked social profiles [4]. Social media marketers often run a campaign on multiple online social platforms like blogs and OSNs and offline platforms like TV to maximize reach to their customers. Often, they need to estimate audience size to measure the success of the campaign. An arithmetic sum of users from each platform is inflated as users can engage in the same campaign via their multiple OSN profiles. Therefore, it is important to de-duplicate users across platforms by linking their profiles in order to estimate campaign ratings and invest accordingly [1, 5].
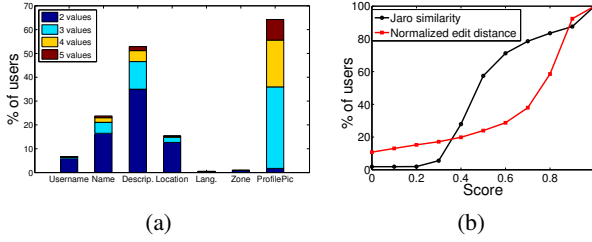
Profile linking methods compare attributes like username and name to find connection between a pair of profiles. However, challenges like dissonant social platforms with partially overlapping list of supported attributes and heterogenous attributes holding veracious values impede effective profile linking. Literature suggests various methodologies equipped with tools that compare overlapping attributes of examined user profiles and match attributes on different metrics. Similarity between text attributes like name is estimated using Jaro similarity while media attributes like profile-picture are compared using face detection algorithms and histogram matching [13, 14, 17, 20, 26]. These methodologies consider most recent (current) values of the attributes and assume high similarity to infer a link between respective profiles. However, the low similarity between these values need *not* imply different users for reasons such as attribute evolution over time [16, 18]. Consider the following scenario – A user registers on Twitter and Facebook with the same username value, she favors Twitter and updates her Twitter profile more frequently than her Facebook profile. After a few weeks, she chooses a new username on Twitter but makes no such changes on Facebook. Due to evolution of username over time on a favored social network, she now owns dissimilar usernames on her profiles. On observing dissimilarity, existing methods that match

Figure 1: Attribute evolution on Twitter. (a) Around 73.21% users tend to change their attributes on Twitter. (b) Users who evolve their username have low similarity between usernames across their profiles. For these users, attribute history can be leveraged for profile linking.

only username falsely conclude that Twitter and Facebook profiles refer to different users.

To examine if a significant section of users change attributes, we deploy an automated system to track 8.7 million Twitter users every fortnight and record changes to their attributes. Figure 1(a) shows the distribution of users that evolve over time and hold distinct values for their attributes. On a two month period, we observe that 73.21% users change their attributes and assign distinct values. Thereby, we gather that attribute evolution is an evident phenomenon. Further, we test if evolution causes dissimilar current values across profiles of users and hence, filter users who evolve their usernames. We compute Jaro similarity and Edit distance between usernames across their profiles and plot the user distribution (see Figure 1(b)). Observe that 78% users have usernames with Jaro similarity < 0.7 and 62% users with Edit distance > 0.7 implying dissimilar values for a majority section of users. Thus, due to username evolution, current usernames have low similarity which is falsely manipulated by existing methods as different users.

For the section of users who evolve and select dissimilar attribute values across profiles, we propose to collect rich information created due to their tendency towards evolution i.e. past values. These past values created by a user, termed as *attribute history*, reveal her preferences and behavior while creating them like the choice of length or characters of username. Co-existence of user preferences across her profiles on different OSNs for a prolonged period of time create alike attribute histories, thus suggesting a potential link to a single user. We, therefore, present the first study that quantifies importance of attribute history to link profiles that are missed by existing methods on observing dissimilar current values. For the aforementioned scenario, due to user's initial choice to select same username across her profiles, past username on Twitter is an exact match to current username on Facebook, and hence an inspection of username history correctly links the profiles. We also cater to the likelihood of different user profiles with similar attribute values.

Scope: A user profile is composed of multiple attributes; each signifies a unique characteristic of the user. Among the attributes, literature suggests username to be an important and discriminating attribute for profile linking [19, 22, 28]. Though a small section of users change username on Twitter (<10%), it is the most common publicly available attribute across OSNs that can uniquely identify users within an OSN. In addition to availability and uniqueness, usernames can only contain alphanumeric and special characters irrespective of the preferred language of the user profile, thereby allowing clean string comparisons. We, therefore, choose to track changes to *username*, collect a set of values, and use the value set for profile linking. History of other attributes like name, description and profile-picture can further help in identifying user profiles of the same user; however lack of their universal support and

availability across social platforms direct us to limit our scope to only username. For this study, we ask following research question: *Given two user profiles and respective username histories on a pair of OSNs, can we predict that profiles belong to the same user?*

Contributions: On a labelled dataset of 128,251 pairs of username histories accessed from users with profiles on four popular social networks – Twitter, Instagram, Facebook and Tumblr, we examine the viability of using username history for profile linking and impact of various factors that govern its effectiveness. We show that:

- Out of 89% profile pairs that current values fail to link, a comparison of username histories correctly link 44% profile pairs while keeping a false positive rate of 1.65%. Therefore, attribute history helps profile linking.

- Out of 45% profile pairs that username history fails to link, 90.88% results from Twitter-Tumblr profile comparisons, while 5.50% results from Twitter-Instagram profile comparisons. Therefore, effectiveness of attribute history varies with the choice of OSNs user profiles belong to.

- Availability of username history only on one profile increases false linkings by 12% as compared to its availability on both.

- Importance of username history for profile linking is directly proportional to its longevity.

To the best of our knowledge, this is the first study that provides insights about estimating the use of attribute history for profile linking. We believe that attribute history can also help other applications that are build on derived behavioral characteristics of users.

2. PROBLEM STATEMENT

We now formally define the research question using following definitions and notations. User profiles under examination belong to a pair of social networks, SN_A and SN_B, termed as *source profile S* and *candidate profile C*, respectively. An evolved *username* set U is a set of pairs, where each pair contains new value and time of evolution of the attribute, ordered on time of evolution i.e. $U = \{(u_1, t_1), (u_2, t_2), \cdots, (u_L, t_L)\}$, where $t_i < t_{i+1}$. Here, L denotes the length of the username set, t_1 denotes the time when first username change is recorded and t_L denotes the time last username change is recorded; u_L represents the most recent (current) value. Username sets on source and candidate profiles are denoted by U_S and U_C, respectively. If past usernames of the candidate profile are *not* available, set U_C is replaced by the current username u_c. We define our problem as –

Given a source profile S on SN_A, a candidate profile C on SN_B and their respective username sets U_S and U_C, each composed of pairs of usernames and their receptive evolution timestamps, find if U_S and U_C refer to the same user $\mathcal{I}$.

A collection of methods can solve the problem. Heuristic approaches like rule based methods, collaborative approaches like crowd sourcing and manual tagging, and algorithmic approaches like machine learning can look for similarities between username sets and infer the potential link between them.

We model profile linking as a classification problem with three phases – feature extraction, labelled dataset collection and supervised machine learning algorithm usage for correct profile identification (see Figure 2). Features extract similarities between usernames across username sets by capturing unique behavioral charac-
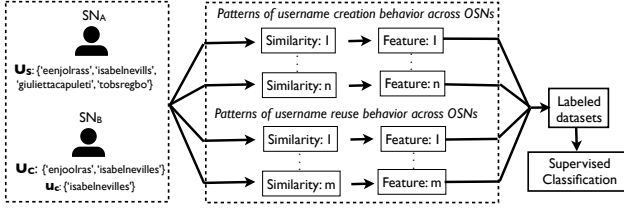
Figure 2: Proposed methodology to compare username sets and capture similarities based on unique behavioral patterns while creating and reusing usernames over time.

teristics and consistent preferences that a user exhibits while choosing usernames across her profiles over time (Section 3); Labelled datasets collect users who evolve their usernames on popular social networks (Section 5) followed by supervised classification by a single and an ensemble of classifiers organized in a framework (Section 6). Section 7 describes relevance of our study to related literature. Finally, Section 8 presents discussion and Section 9 concludes this research with directions for future work.

3. FEATURES

Individuals often exhibit unique and consistent behavior while creating their attribute values across her profiles on different social networks. Cross-OSN analysis of users on OSNs shows that 85% users have more then 50% matching attribute values across different OSNs [7]. These attributes, however, evolve over time, leading to matching histories than current values. Further, a recent study shows that users exhibit similar choices while selecting usernames across OSNs [28]. We believe that such choices may co-exist and repeat over time.

User preferences while creating usernames on her profiles are affected by self-bias and limited memory. Biased choices push users to create similar usernames across her profiles resulting in username creation behavioral patterns. Limited memory constrains users to reuse a username either used in the past within an OSN or across OSN profiles, thus producing username reuse behavioral patterns. We discuss these patterns and methods to quantify them into features.

3.1 Username creation

On a social network, users converse by tagging another user's username with '@' tag. Tagged user specifies username properties that aid these interactions. For instance, a user chooses short usernames on OSNs that restrict message length to help her friends post more content when tagging her [3]. Choice of username properties that remain static over time as well as repeat across OSNs constitute *static behavioral patterns*. For instance, a static choice of a user to create short username across OSNs is a static behavioral pattern.

With time, changing expectations from an OSN can push her to change a few username properties but reflect the same changes across her other OSN profiles. Similar transitions in username properties constitute *temporal behavioral patterns*. For instance, the user later realizes that short usernames confuse her followers, therefore now creates long personalized usernames. Such a transition in user behavior, if evident across all her OSN profiles, results in a temporal behavioral pattern. We now explore methods to quantify these patterns.

3.1.1 Static behavioral patterns

Username creation behavior can be captured by observing three string properties of usernames – length, choice of characters, and the arrangement of characters. It is likely for a user to create user-

names of similar length with a limited set of characters compiled in similar fashion. For both source and candidate username sets, we calculate these properties and compare using different methods.

Length of a username l_{u_i} is calculated by counting alphanumeric characters in the username. Length distribution of usernames in source $\mathcal{L}_S$ is compared with that of usernames in candidate username set $\mathcal{L}_C$ using JS divergence. The low divergence hints use of similar username lengths across OSNs.

To compare choice of characters, we compare character distribution of usernames in source $\mathcal{C}_S$ with that of usernames in candidate username set $\mathcal{C}_C$ using Jaccard similarity index J and cosine similarity cos. The best value at '1' for both metrics implies same choice of characters on username sets, made by the same user.

To compare the arrangement of characters, we compute string similarity between usernames of different sets. We calculate normalized Longest Common Subsequence (LCS) similarity score between u_i and u_j such that u_i, u_j belong to different sets and estimate mean, median and standard deviation of score distribution $\mathcal{A}$. The low standard deviation of the distribution hints similar arrangement of characters likely to be made by the same user while high mean and median values denote the high similarity among usernames in the two sets. In a nutshell, static features are:

$$F_{static} : (JS(\mathcal{L}_S || \mathcal{L}_C), J(\mathcal{C}_S, \mathcal{C}_C), cos(\mathcal{C}_S, \mathcal{C}_C), \mathbb{E}(\mathcal{A}),$$
$$med(\mathcal{A}), \ \sigma(\mathcal{A}))$$

3.1.2 Temporal behavioral patterns

With changing requirements on an OSN over time like privacy concerns, a user can consider to change a few properties of new usernames she creates within an OSN. For instance, user can start using initials over full name in her username, thereby anonymizing and shortening its length. It is likely that her new preferences influence usernames created on other OSNs as well. Similar transitions in selected properties of username with time across OSNs result in similar evolutionary patterns of properties. We capture such patterns by comparing evolution sequence of the username properties computed for each username set.

Consecutive usernames of each username set are compared on length, character distribution and arrangement of characters, resulting in three comparison vectors for each set – length, character, and arrangement vector. Length vector $\mathbb{L}$ is a sequence of lengths l_{u_i}, character vector $\mathbb{C}$ is a sequence of Jaccard index and cosine similarity scores between character distribution while arrangement vector $\mathbb{A}$ is a sequence of string similarity scores between consecutive usernames of a username set. For the arrangement vector, we use four string similarity metrics – Edit distance, Jaro similarity, LCS similarity and Longest Common Substring similarity (LCSub). Multiple similarity metrics ensures different penalties for character insertion, deletion, replacement, and re-arrangement. Normalized versions of string similarity scores are used in the arrangement vectors.

Length, character and arrangement vectors for two username sets are compared to find any correlation between the two sets. We use normalized cross-correlation (NCC) to compute the correlation, whose values ranges from -1 to 1. This metric is used to find correlation between two time series data lists as a function of lag τ at which the time series best align each other, also used for temporal analysis on Twitter in [24]. A positive correlation implies similar pattern of evolution of the username property on both username sets, from which we may link username sets to the same user. In a nutshell,

$$F_{temporal} : (NCC(\mathbb{L}_S, \mathbb{L}_C), NCC(\mathbb{C}_S, \mathbb{C}_C), NCC(\mathbb{A}_S, \mathbb{A}_C))$$

3.2 Username reuse

With an increasing number of OSNs and evolving preferences, a user struggles to remember her latest usernames on all OSNs in order to sign in or use the usernames for interactions. However, a naive reuse of a username borrowed from her other OSN profiles can ease her cognitive load [28]. Reused username can either be a latest username or an old username from any of her OSN profiles. Frequent tendency to reuse a username from other profiles results in a set of common usernames appearing in the same order at the same time across user profiles indicating user synchronous behavior across her profiles.

3.2.1 Occasional reuse

User's choice of reusing a username from her other profiles at least once results in observing that username on different profiles at different times. To find the common username, we intersect username lists extracted from each username set. If the intersection results in an empty set, there is a possibility that the username she wants to use is already taken by a different user within the OSN. In that case, user can make minor modifications to the selected username to create an available version and use the available version on the OSN. With minor modifications, selected username and its available version have a high string similarity score. We, therefore, perform pairwise comparisons between usernames from different sets to find best matching username pair,

$$\max_{(u_i, t_i) \in U_S, (u_j, t_j) \in U_C} Sim(u_i, u_j)$$

We compute the similarity based on four string based metrics – edit distance, jaro similarity, LCSub similarity and LCS similarity. We acknowledge that the existence of a common username or a pair of similar usernames between two username sets can be co-incidental. It is likely that different users pick the same username at some point in their past. This can happen to usernames derived from celebrity, brand or popular names. Therefore, we calculate second best similarity score between usernames from different sets. A low second best similarity indicates that the best similarity can be an outlier, implying that username sets refer to different persons.

3.2.2 Frequent reuse

Repeated use of borrowed usernames results in a set of common usernames between profiles of a user. We examine if there exists a set of common usernames and compute a boolean feature. We estimate the ratio of common usernames to the size of smaller username set which denotes if all (or few) usernames are copied from other OSN profiles. A sequential and simultaneous use of common usernames across OSNs lends support to the belief that username sets refer to the same user. It is highly unlikely for different users to choose same usernames in the same order at the same time across multiple OSNs. Further, similar sequential ordering of common usernames in both sets is an indicator of a single user consistently choosing same usernames over time across her profiles. Earlier research suggests Smith-Waterman algorithm as an effective algorithm to measure sequential ordering [13], originally proposed to perform sequence alignment in protein sequences [25]. We use Smith-Waterman similarity to estimate sequential ordering between common usernames in the username sets. To capture temporal synchrony, we use timestamps of evolution to find if same usernames are used on both sets at the same time.

As described earlier, users may make minor modifications to a selected username, in order to create an available version to use on the OSN. We incorporate such minor modifications while calculating set of common usernames. We consider two usernames as
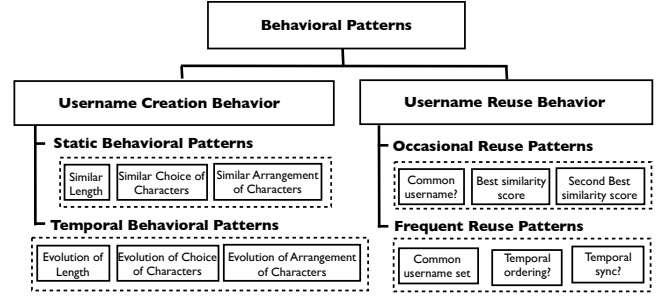

Figure 3: Various username creation and reuse behavioral patterns of users.

variations of the same username, if LCS string similarity is above a threshold. We adjust the threshold from 0.8 to 1 and compute set of common usernames and other features accordingly.

To summarize, we list possible similarities between username sets resulting from synchronous user behavior when selecting usernames within and across OSNs over time. Discussed methods quantify these similarities into a set of twenty-six normalized features. We extract these features from a labeled dataset of username sets, learn a supervised classifier and use it to predict connection between test username sets. In scenarios where past usernames are accessible only on one user profile, we compute static behavioral patterns of username creation and occasional reuse between the source username set on source profile and candidate current username.

4. FRAMEWORK

We experiment with two plausible supervised frameworks – Independent framework and Cascaded framework.
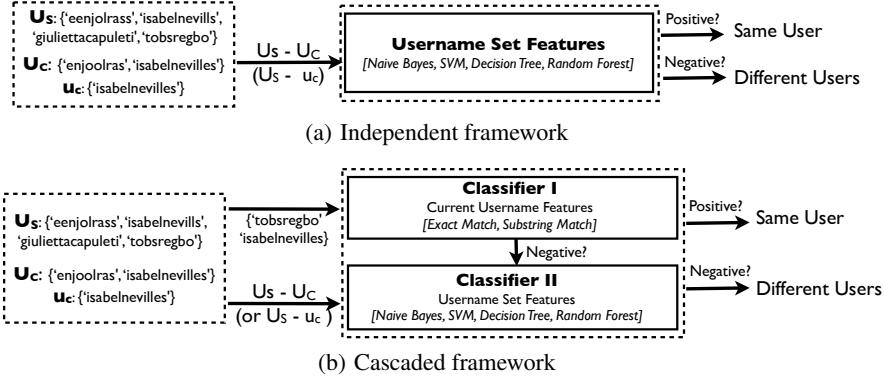
4.1 Independent framework

Most profile linking approaches use a feature set, labelled dataset and a single classifier to predict link between test profiles [17, 19, 22, 28]. Classifier decision is not revised further either manually or computationally. We experiment with such a framework by learning a supervised classifier on proposed features extracted from username sets in the labelled datasets (see Figure 4(a)). However, we suspect the dominance of a subset of features that extract similarities between histories than current values. Hence, trained classifier can be biased towards finding similar histories and can falsely label username sets with dissimilar past but similar current values as negative. To avoid this, we suggest cascaded framework.

4.2 Cascaded framework

Cascaded framework is an ensemble of two classifiers trained on different features to uncover link between two profiles and is extensively used in machine learning domain [12]. **Classifier I** extracts current username features and uses an existing method to classify username sets while **Classifier II** extracts proposed username set features and uses a supervised classifier to re-classify username sets labelled as negative by **Classifier I** (see Figure 4(b)). We further experiment with two existing profile linking methods as **Classifier I** and different supervised classification techniques as **Classifier II** of the framework. These existing methods act as baselines, also used in [19, 28] to evaluate performance of the suggested features:

- **Exact matching (b1):** Two username sets are linked if current usernames are an exact match.

- **Substring matching (b2):** Two username sets are linked if substring similarity score between respective current usernames is beyond a threshold. We use Jaro similarity score to

(a) Independent framework

(b) Cascaded framework

Figure 4: Independent v/s cascaded framework. Independent framework uses proposed features to classify username sets while cascaded framework uses proposed features to re-classify username sets predicted as negative by another classifier.

compute substring similarity, and vary the threshold to report best achieved accuracy.

5. DATA COLLECTION

For a positive dataset, we need to know accounts of a user on multiple OSNs. We start with Twitter, choose a random set of users and find their profiles on three popular social networks that contain quality information about a user[1] – Facebook, Instagram, and Tumblr. All networks, except Facebook, allow multiple changes to username. Facebook allows username change only once.

Ground Truth: One way to identify other OSN profiles of selected Twitter users is by manually search for the profiles, which is cumbersome and time-consuming. Another way is to exploit the tendency of users to broadcast hyperlinks to other OSN profiles via URL attribute of their Twitter profiles [15]. Such users *self-identify* themselves on other OSNs. For instance, a user posts *www.facebook.com/username* on her URL attribute, thereby informing other Twitter users about her Facebook profile. Similar methods are used in literature to create positive datasets either from social aggregation sites, forums or social networks where users *self-list* their OSN accounts [28].

Username History: Once user profiles are identified across OSNs, we collect past usernames owned by the user profiles. We build an independent tracking system for Twitter to monitor any changes to 8.7 million randomly chosen Twitter profiles as on October 2013. Tracking system repeatedly query Twitter Search API with *user_id* of the user profile after every fortnight and store responses mentioning username, name, URL and similar details, user owns at the time of the query. The system then compare consecutive API-responses to take a note of any changes to usernames, names, URLs, etc. Unique usernames chosen by the queried user profile over the tracking period of October 1, 2013 to November 26, 2013, constitute past username set on Twitter. Note that, the system collects only publicly available data on social networks and does not engage in any user authorization asking for private data.

To gather past usernames used on other OSN profiles of the user, one can deploy a similar independent tracking system to track each OSN profile. However, configuring and deploying a tracking system for each OSN requires extensive infrastructure.[2] To reduce infrastructure costs, we use an alternate way to record user-

name changes on other OSNs. We record URL attribute of the Twitter user profile to mark any changes to her username on other OSN. For instance, a Twitter user puts her URL attribute from *www.instagram.com/happygu!* to *www.instagram.com/gulben!* to notify Twitter followers (or others) about her new username on Instagram. We exploit this method to record username changes on users' Facebook, Instagram or Tumblr profiles. We also time username changes on the social networks. Other methods to collect past usernames are discussed in Section 8.

Pre-processing: Recorded usernames on Twitter, Facebook, Instagram and Tumblr profiles are processed prior comparison. Usernames on most social networks are case-insensitive, therefore, usernames are converted to lower case. Further, different OSNs allow a different set of special characters in the usernames. Twitter allows underscore '_', Tumblr allows the hyphen '-', Instagram and Facebook allow dot '.'. A user's wish to reuse a past username on other OSN in its exact form can be restricted by the use of special characters. She needs to replace the special characters with those allowed on the other OSN. To avoid low similarities or miss exact username matches between two username sets, we remove special characters from the usernames. Since no feature captures choice of special characters, their removal will not affect our results.

Dataset: Without the loss of generalizability, we use Twitter profile as a source profile and the corresponding username set as a source username set U_S for our experiments. We use other OSN profile (Tumblr, Facebook or Instagram) as a candidate profile and the respective username set as a candidate username set U_C. If candidate usernames set is not accessible, current username of the candidate profile is used as u_c. Post processing, we collect 18,959 $U_S - U_C$ username set pairs and 109,292 $U_S - u_c$ pairs, totaling 128,251 instances whose username sets are known to belong to a single user and hence are positive instances (see Table 1). We create an equal number of negative instances, by randomly pairing a username set of a positive instance with a username (set) of a different positive instance, which are known to belong to different users. We extract features from positive and negative instances and use features in an engineered framework that effectively classifies username sets as same or different users.

Table 1: Datasets capture username changes of 128,251 users within two months on source and candidate networks.

	Tumblr	Facebook	Instagram	Total
$U_S - U_C$	14,301	1,166	3,492	18,959
$U_S - u_c$	58,285	31,076	19,931	109,292

---

[1] http://mashable.com/2013/04/12/social-media-demographic-breakdown/

[2] Tumblr API does not share a unique *user_id* of a user to keep track of changes to her Tumblr profile, hence development of an automated tracking system is challenging.

Table 2: Accuracy, FNR and FPR of supervised frameworks, baselines learned using current username features and their integration with another classifier learned on proposed feature set using different supervised classification techniques.

Framework Config.	$U_S - U_C$			$U_S - u_c$		
	Accuracy	**FNR**	**FPR**	**Accuracy**	**FNR**	**FPR**
Independent [Naive Bayes]	72.19%	55.86%	0.13%	73.58%	48.15%	4.7%
Cascaded [b1→Naive Bayes]	72.48 %	55.27%	0.14%	73.59%	47.12%	5.7%
Exact Match (b1)	55.38%	89.34%	0.00%	59.26%	84.61%	0.00%
Substring Match (b2)	60.99%	78.46%	0.00%	58.25%	83.53%	0.00%
b1 → Naive Bayes	72.48%	55.27%	0.14%	73.59%	47.12%	5.7%
b1 → SVM [Linear]	**76.74%**	**45.16%**	1.65%	74.75%	47.9%	2.6%
b1 → SVM [RBF]	76.02%	47.33%	0.94%	74.52%	49.18%	1.8%
b1 → Decision Tree	67.95%	30.31%	33.76%	66.92%	31.79%	34.35%
b1 → Random Forest	72.09%	32.47%	23.4%	70.13%	33.4%	26.33%
b2 → Naive Bayes	72.51%	54.97%	0.17%	73.44%	46.2%	6.77%
b2 → SVM [Linear]	**76.84%**	**45.16%**	1.25%	74.88%	47.89%	2.21%
b2 → SVM [RBF]	75.92%	47.33%	0.94%	74.42%	49.18%	1.84%
b2 → Decision Tree	67.74%	30.89%	33.4%	66.86%	31.87%	34.19%
b2 → Random Forest	72.46%	32.06%	22.86%	70.17%	33.24%	26.22%
b1 w/o Tumblr	66.17%	67.51%	0.00%	57.56%	85.35%	0.00%
b2 w/o Tumblr	70.95%	57.9%	0.00%	61.8%	76.69%	0.00%
(b1 → SVM [Linear]) w/o Tumblr	**91.20%**	**16.6%**	0.96%	83.7%	30.56%	2.1%
(b2 → SVM [Linear]) w/o Tumblr	91.04%	16.6%	0.96%	83.56%	30.56%	2.18%

6. EVALUATION

We evaluate listed frameworks on two genre of instances: $U_S - U_C$ instances (18,959 positive; 18,959 negative) and $U_S - u_c$ instances (109,292 positive; 109,292 negative) and on three metrics – *accuracy*, *false negative rate* (FNR) and *false positive rate* (FPR). Accuracy shows number of username sets correctly classified. False negative rate shows number of username sets falsely classified as unlinked while false positive rate shows the number of username sets falsely classified as linked. We use Naive Bayes classifier for both independent or cascaded framework with **b1** as the **Classifier I** (see Table 2). We observe that cascaded framework gives a slight better accuracy and false negative rate than independent framework while maintaining a low false positive rate.

Performance of Cascaded Framework: We now experiment with different baselines used as **Classifier I** and different supervised machine learning algorithms as **Classifier II** in the cascaded framework. Table 2 details 10-fold cross validated accuracy, FNR and FPR of the framework. Classifying $U_S - U_C$ instances with only **b1** results in false negative rate of 89.34% and an accuracy of 55.38%. The high false negative rate alerts that most users have non-matching current usernames across their OSN profiles. **Classifier II** learned using Naive Bayes technique exploits username set features of **b1** negative predictions and reclassifies them. Reclassification reduces false negative rate to 55.27% thereby boosting accuracy to 72.48%. A significant reduction in false negative rate by 34% is due to the comparison of username history. We experiment with other supervised methods to learn the classifier, and achieve best accuracy with SVM (reduction by 44.18%) while maintaining a low FPR. With baseline **b2** as **Classifier I**, the framework achieves best accuracy of 76.84% and reduction in false negative rate by 33.30% with SVM classifier learned on username set features as **Classifier II**. ROC curves in Figure 5 shows that in order to gain higher TPR with **Classifier II**, which directly contributes to the reduction in FNR of the framework, we need to compromise on FPR of the framework.

Significant reduction in FNR of the framework implies that the username history helps in linking user profiles and is an important feature for profile linking methods. An example where baselines fail to link with current usernames but cascaded framework compares the username sets and finds the link is two chronologically ordered sets – { U_S: ['*eenjolrass', 'isabelnevills', 'giuliettacapuleti',*
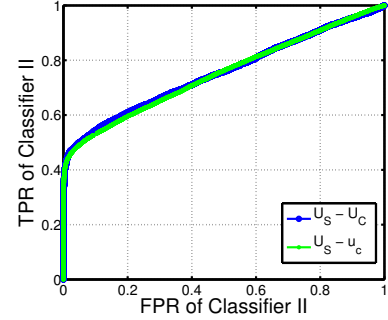

Figure 5: ROC curve when SVM with RBF kernel is used as **Classifier II**. 40% TPR at low FPR implies that FNR of the framework reduces by 40%.

'*tobsregbo*'], U_C: ['*enjoolras', 'isabelnevilles*'] }. We see that current usernames do not match, however two of the past usernames are similar.

Classification of $U_S - u_c$ instances shows similar trends. On comparing classification accuracies of $U_S - U_C$ and $U_S - u_c$ instances, we observe that without access to candidate's past usernames, framework achieves a little less but similar accuracies. Lower linking accuracies for $U_S - u_c$ can be attributed to a slight increase in FPR. We, therefore, investigate if history availability on both profiles is beneficial for profile linking. Using $U_S - U_C$ instances, we create another dataset where we access only the current username of the candidate profile. With **b1** and SVM classifier (linear), we achieve an accuracy of 70.43% (FNR: 45.25%, FPR: 13.77%). Observe that due to increased FPR, profile linking accuracy fall from 76.74%, when username history on both profiles is available, to 70.43%, when username history is available only on source profile. Therefore, a comparison of a single username with a set may lead to higher FPR than a comparison of two username sets.

Impact of choice of OSNs: Though cascaded framework significantly reduces false negative rates, we are curious why false negative rates are still high ($\sim$45%). To answer the question, we plot a distribution of false negative instances among the three candidate social networks (see Figure 6(a) and 6(b)). We find that an enormous 90.88% of false negatives results from Twitter-Tumblr username set comparisons. A high false negative rate on Tumblr can be attributed to the lowest Jaro similarity between most sim-

(a) $U_S - U_C$ instances   (b) $U_S - u_c$ instances   (c) Jaro similarity distribution
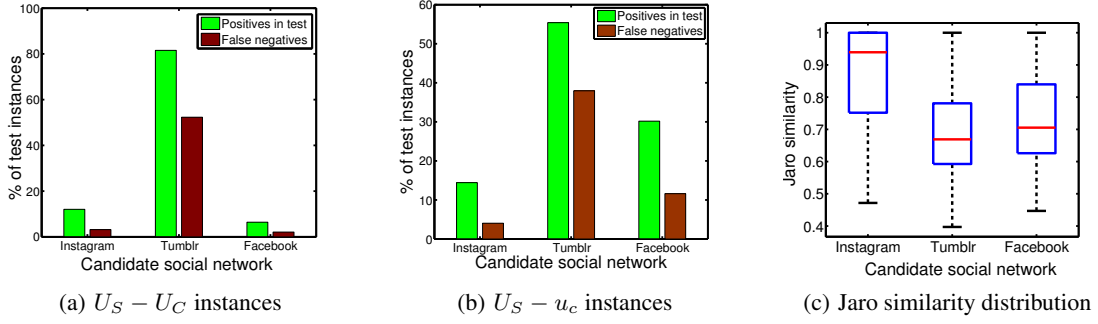
Figure 6: False negatives distribution among three candidate networks; Tumblr results in most false negatives. On further analysis, we observe that among the three candidate networks, Tumblr usernames have least Jaro similarity with corresponding Twitter usernames.

ilar usernames from Tumblr and Twitter username sets (see Figure 6(c)). For instance, a user's usernames on Twitter – ['*articulatedan*', '*radicaliguori*', '*satanichowell*'] do not hold any similarity with her usernames on Tumblr – ['*ptvkitty*', '*piercethecait*', '*ptvcait*']. Best Jaro similarity score for the username sets is 0.56. For instances like this, we need support of other attributes like name, location to find link between the two profiles. We thus evaluate cascaded framework only on instances with candidate profile on either Facebook or Instagram. We achieve an accuracy of 91.20% on 4,658 $U_S - U_C$ instances (FNR: 16.6%, FPR: 0.96%) and 83.70% on 51,007 $U_S - u_c$ instances (FNR: 30.56%, FPR: 2.1%). On removal of candidate network Tumblr, a significant improvement in the accuracy shows that proposed cascaded framework can accurately find links between two user profiles given the username sets resemble and are created with similar behavioral characteristics.

To summarize, the key observations are – i) Cascaded framework performs better than an independent framework, ii) A comparison of username history reduces false predictions by 44% which are caused by the only comparison of current usernames, iii) Cascaded framework returns lower FPR with availability of past usernames on both profiles than on only one, iv) Success of the framework relies on the platforms to which examined profiles belong to; Twitter-Tumblr username set comparisons lead to 90.88% false negatives while Twitter-Instagram leads to 5.50% for U_S - U_C instances (FNR on U_S - u_c for Twitter-Tumblr: 70.81%; Twitter-Instagram: 7.82%). Our experiments on fairly large datasets give a detailed proof of concept on the importance of using username history for profile linking. Applicability and dependencies of the framework are discussed in Section 8.

6.1 Feature importance

We now detail features that help the most during classification of usernames sets. We examine feature weights to estimate their importances for the most accurate framework configuration – Exact matching (**b1**) followed by temporal matching using SVM and compute them by squaring coefficients of features returned by **Classifier II** as suggested in [11]. Top-10 features, calculated between source and candidate username sets, are –

1. Maximum normalized LCSub similarity.

2. Second best normalized LCS similarity.

3. Minimum normalized edit distance.

4. Maximum normalized jaro similarity.

5. Median of LCS similarity between source and candidate username pairs.

6. Standard deviation of LCS similarity between source and candidate username pairs.

7. Mean Jaccard similarity between alphabet distribution of source and candidate username pairs.

8. Second best normalized edit distance.

9. Maximum normalized LCS similarity.

10. Second best normalized LCSub similarity.

Note that, top-10 features capture username creation behavior of a user. Username creation behavior play an important role for classification, but username evolutionary features and reuse behavior have relatively weaker roles. We analyze if evolutionary and frequent reuse patterns can contribute better given a longer history to find connections between the user profiles in Section 8.

7. RELATED WORK

Profile linking is a well studied problem in literature. Existing literature addresses the problem of connecting user profiles across social networks by comparing current values of the attributes of user profiles. The suggested methods explore combination of attributes to compare and techniques to measure the similarity. Profile attributes such as username and name are compared using string similarity measures [13, 17, 22, 28], content attributes such as posts and message length are compared using language models [10, 13, 17], and network attributes such as number of friends and nature of ties with friends are compared using graph algorithms [6, 17, 21]. Few studies suggest crowd-sourced mechanisms to match user profiles across OSNs [23]. Others have examined the effectiveness of using only usernames to connect user profiles across OSNs [22, 28]. The state-of-the-art method MOBIUS compares a candidate username with a set of usernames owned by a user profile on other OSNs. MOBIUS assumes that user's unique behavior often leads to redundancies / similarities among the usernames across OSNs, which can be captured into features. Supervised classification techniques then predict if a candidate username and usernames on other OSNs are linked [28].

Most profile linking methods compare user profiles based on current values of the attributes observed at the time of executing the method. Existing methods are successful when users do not evolve their attribute values over time. However, recent studies show that users frequently change their attributes to suit their changing preferences on different OSNs [18, 29], similar to our observations in the study. In these scenarios, current values of the attributes on multiple profiles of a user may not match, thereby leading existing methods to falsely infer that user profiles as different users.

To address the limitations of existing methods and complement MOBIUS, we suggest considering attribute history to find links between user profiles. We propose to compare a candidate username with a set of past usernames of multiple user profiles across networks, not just with current values. We re-implement MOBIUS and build a framework with **Classifier I** extracting top-10 features by comparing candidate username with a set of current usernames on other OSNs, as proposed by the authors and **Classifier II** extracting username set features by comparing candidate username history with other profiles' username histories as proposed in this work. On a dataset of 8,997 users who have profiles on more than two social networks as well past history on all the social networks, 42.67% instances are false negatives i.e. **Classifier I** miss the link among profiles. **Classifier II** identifies links among 30.72% more instances, reducing false negatives to 11.95%. Therefore, we see that attribute history complements state-of-the-art method and extends support to existing profile linking methods.

8. DISCUSSION

On a dataset of real-world users, we show that username history holds its significance by extending performance to existing methods for profile linking. However, its effectiveness varies with the choice of OSNs. We observe that majority users create different usernames on Tumblr as compared to their profiles on Twitter, Facebook or Instagram. Differences between the username sets hints disparate user needs and choices across OSNs. We think that profile linking strategies need to tune according to the nature and genre of the OSN with a prior knowledge of popular user behavior on that OSN. Now, we discuss applicability of attribute history along with other dependencies of the framework that uses attribute history for linking.

8.1 Applicability

Apart from observing users over time on OSNs, one can get user history archived by external services like DataSift[3] or Gnip[4]. We further suggest other two methods to collect past usernames – via timeline and via public datasets.

8.1.1 Via timeline

On social networks like Twitter and Instagram, users converse by tagging another user's username with '@' tag. When a user changes her username, old tweets and replies where others tagged her with her old username stay on her timeline. By listing old posts with replies and extracting mentions from the tweets, one may list her past usernames. We believe that a recent history of past usernames can be captured by this method.

8.1.2 Via public datasets

Multiple researchers collect private and public posts related to a topic, event or a campaign ranging over a period of time. They often store information about authors who created these posts. One may query these databases with the *user_id* of a user and find posts created by her at different times. If the author details are recorded with each post, one may list unique usernames used by the user in the past. With this methodology, we find past usernames of 4% of 128,251 Twitter users, via datasets shared by an event monitoring tool, MultiOSN [9].

With the listed methods, applicability of the proposed profile linking framework can be extended to random users who are not tracked continuously over time.

8.2 Dependency

We test the proposed framework for dependency on the grounds of understanding how much history is required for efficient profile linking. In other words, does a longer history on source username set impact framework accuracy? To answer the question, we create a dataset of $U_S - u_c$ username sets with 502 users from the dataset of 109,292 users who had changed their Twitter username maximum number of times (5 times) within tracking period of two months. We further partitioned 502 $U_S - u_c$ sets into 4 datasets $(d_i)_{i=2}^5$, where dataset d_i contains instances with first i past usernames from their respective U_C sets. For instance, d_2 contains 502 $U_S - u_c$ instances, where each U_C contains only first two usernames of the five usernames in the username set. FNR by cascaded framework with respect to the baseline **b1** on the derived datasets with varying set sizes is shown in Figure 7.

Observe that as past username set size increases, difference between FNR of the framework and FNR of the baseline increases, thereby indicating that longer the username history of a Twitter user, better the matching with a candidate username or set.
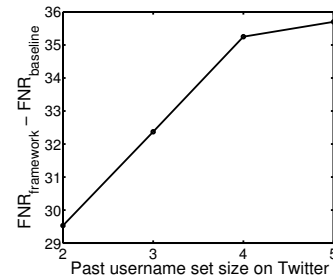

Figure 7: Higher FNR reduction with increasing source username set size.

8.3 Importance of temporal creation and reuse behavior

Section 6.1 suggests that username creation behavior helps better than other behavioral patterns to suggest if username sets refer to a single user. We suspect that a user's evolutionary behavior or her tendency to reuse usernames across social networks over time are of little help to the classification process due to fewer instances with these features. We, therefore, repeat feature importance analysis for another dataset with longer username history. We randomly sample a set of 10,000 users from 128,251 users on Twitter and record their attributes every fifteen minutes for 12 months (November 26, 2013 - November 28, 2014). Out of 10,000 users, 47% users change their username at least once during the tracking period. To create ground truth dataset, we filter users who self-identify themselves on at least one of the candidate social networks – Instagram, Tumblr or Facebook. For 682 users, we retrieve their current username on either of the candidate networks while for 155 users we retrieve their past usernames on both Twitter and one of the candidate networks. SVM classifier with linear kernel, used as **Classifier II**, ranked 'username reuse' features above 'username creation' features deemed important earlier – ratio of common usernames to candidate set size, and number of common usernames found between source and candidate username sets are ranked above than mean Jaccard similarity between two sets. Therefore, we gather that relative importance of behavioral patterns to reveal a potential link between two user profiles varies with the longevity of the attribute history on either profiles. Username reuse behavior can be only observed over a prolonged track of history, however has proven useful feature for profile linking.

9. CONCLUSION AND FUTURE WORK

In this work, we emphasize that attribute values evolve over time on one or multiple profiles of a user. This causes non-matching current attribute values on the profiles. Existing methods do not consider attribute evolution and falsely predict non-matching user profiles as different users. To avoid such mistakes, we propose to compare attribute's past values, not just current values, to revise the prediction. Focusing on username, we find similarities between username sets, comprised of past and current username of a user profile. We assume that user behavior of username creation and its reuse across her profiles is unique; these user behavioral patterns can remain static or vary over time and are captured into a set of features. Novel cascaded framework of classifiers uses proposed features to rectify erroneous classifications based only on current values. We show that our framework outperforms existing profile linking methods by reducing false negative errors by 44%. In conclusion, comparing username history along with current values help in effective profile linking.

In future, we plan to extend our work on two fronts. First, further reduce the false negative rate with the help of other features such as name, description and profile picture, and second, evaluate the proposed framework on a random set of users rather than users who self-identify themselves. We believe that an accumulation and knowledge of a user's past profiles can also help in other research domains other than profile linking. Our work can be extended to profile search frameworks, which is an important part of profile resolution process [15], however is not the focus of this study.

10. REFERENCES

[1] The Five Biggest Mistakes in Measuring Social Media. www.clickz.com/clickz/column/1716119/the-five-biggest-mistakes-measuring\ \social-media, 2009.

[2] Segmenting Target Markets. thewaltdisneyco.blogspot.in/2011/11/chapter\-8\ -segmenting\-targeting\-markets.html, 2011.

[3] 4 Reasons Why You Need to Change Your Username on Twitter. www.adweek.com/socialtimes/twitter-username-tips/453851?red=at, 2012.

[4] How Social Media Influences Market Segmentation. http://www.marketingtechnews.net/news/2012/mar/16/how-social-media-influencing-\ \marketing-segmentation/, 2012.

[5] Deduplicate Audiences With Cross-Media Planning. http://www.beet.tv/2015/01/br15nielsenhohnan.html, 2015.

[6] Sergey Bartunov, Anton Korshunov, S Park, Wonho Ryu, and Hyungdong Lee. Joint Link-Attribute User Identity Resolution in Online Social Networks. In *SNAKDD*, 2012.

[7] Terence Chen, Mohamed Ali Kaafar, Arik Friedman, and Roksana Boreli. Is More always Merrier?: A Deep Dive into Online Social Footprints. In *WOSN*, 2012.

[8] G. Cockerell. *Making Marketing Meaningful*. Kendall Hunt Publishing Company, 2010.

[9] Prateek Dewan, Mayank Gupta, Kanika Goyal, and Ponnurangam Kumaraguru. Multiosn: Realtime monitoring of real world events on multiple online social media. In *I-CARE*, 2013.

[10] Oana Goga, Howard Lei, Sree Hari Krishnan Parthasarathi, Gerald Friedland, Robin Sommer, and Renata Teixeira. Exploiting Innocuous Activity for Correlating Users across Sites. In *WWW*, 2013.

[11] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene Selection for Cancer Classification using Support Vector Machines. *Machine learning*, 2002.

[12] Geremy Heitz, Stephen Gould, Ashutosh Saxena, and Daphne Koller. Cascaded Classification Models: Combining Models for Holistic Scene Understanding. In *NIPS*, 2009.

[13] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying Users Across Social Tagging Systems. In *ICWSM*, 2011.

[14] Danesh Irani, Steve Webb, Kang Li, and Calton Pu. Large Online Social Footprints–An Emerging Threat. In *CSE*, 2009.

[15] Paridhi Jain, Ponnurangam Kumaraguru, and Anupam Joshi. @ I seek 'fb. me': Identifying Users across Multiple Online Social Networks. In *WWW Companion*, 2013.

[16] Pei Li, Xin Luna Dong, Andrea Maurino, and Divesh Srivastava. Linking Temporal Records. *VLDB*, 2011.

[17] Siyuan Liu, Shuhui Wang, Feida Zhu, Jinbo Zhang, and Ramayya Krishnan. HYDRA: Large-scale Social Identity Linkage via Heterogeneous Behavior Modeling. In *SIGMOD*, 2014.

[18] Yabing Liu, Chloe Kliman-Silver, and Alan Mislove. The Tweets They are a-Changin': Evolution of Twitter Users and Behavior. In *ICWSM*, 2014.

[19] Anshu Malhotra, Luam Totti, Wagner Meira, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying User Footprints in Different Online Social Networks. In *ASONAM*, 2012.

[20] Marti Motoyama and George Varghese. I seek You: Searching and Matching Individuals in Social Networks. In *WIDM*, 2009.

[21] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing Social Networks. In *SP*, 2009.

[22] Daniele Perito, Claude Castelluccia, Mohamed Ali Kâafar, and Pere Manils. How Unique and Traceable Are Usernames? In *PETS*, 2011.

[23] Mohamed Shehab, Moo Nam Ko, and Hakim Touati. Social Networks Profile Mapping using Games. In *USENIX*, 2012.

[24] Xiaolin Shi, Ramesh Nallapati, Jure Leskovec, Dan McFarland, and Dan Jurafsky. Who Leads Whom: Topical Lead-lag Analysis across Corpora. In *NIPS Workshop*, 2010.

[25] Temple F Smith and Michael S Waterman. Identification of Common Molecular Subsequences. *Journal of molecular biology*, 1981.

[26] Martin Szomszor, Ivan Cantador, Escuela Politecnica Superior, and Harith Alani. Correlating User Profiles from Multiple Folksonomies. In *HT*, 2008.

[27] A. Weinstein. *Handbook of Market Segmentation: Strategic Targeting for Business and Technology firms*. Haworth Press, 2004.

[28] Reza Zafarani and Huan Liu. Connecting Users across Social Media Sites: A Behavioral-modeling Approach. In *KDD*, 2013.

[29] Jun Zhang, Chaokun Wang, and Jianmin Wang. Learning Temporal Dynamics of Behavior Propagation in Social Networks. In *ICWSM*, 2014.