Connecting Users across Social Media Sites: A Behavioral-Modeling Approach

Reza Zafarani and Huan Liu
Computer Science and Engineering
Arizona State University
{Reza, Huan.Liu}@asu.edu

ABSTRACT

People use various social media for different purposes. The information on an individual site is often incomplete. When sources of complementary information are integrated, a better profile of a user can be built to improve online services such as verifying online information. To integrate these sources of information, it is necessary to identify individuals across social media sites. This paper aims to address the cross-media user identification problem. We introduce a methodology (MOBIUS) for finding a mapping among identities of individuals across social media sites. It consists of three key components: the first component identifies users' unique behavioral patterns that lead to information redundancies across sites; the second component constructs features that exploit information redundancies due to these behavioral patterns; and the third component employs machine learning for effective user identification. We formally define the cross-media user identification problem and show that MOBIUS is effective in identifying users across social media sites. This study paves the way for analysis and mining across social media sites, and facilitates the creation of novel online services across sites.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*

Keywords

User Identification; Cross-Media Analysis; MOBIUS

1. INTRODUCTION

Verifying ages online is important as it attempts to determine whether someone is "an 11-year-old girl or a 45-year-old man". Its significance is convincingly pointed out by The New York Times [16], which reported "Skout, a mobile social networking app, discovered that, within two weeks, three adults had masqueraded as 13- to 17-year olds. In three separate incidents, they contacted children and, the police say, sexually assaulted them." Age verification is also an elusive problem to solve. In 2008, a serious effort was made to evaluate age verification technologies when the Internet Safety Technical Task Force was convened with experts from academia and Web companies. The same report mentions that "four years later, members of that task force sound, at best, deflated" and that "an informal survey of major figures in the Artificial Intelligence industry revealed that little, if any, research is being done on age verification". The New York Times report suggests that age verification is even more difficult for social media, where people can expect a degree of anonymity.

This paper proposes an alternative solution addressing the age verification problem by exploiting the nature of social media and its networks. Social media can provide rich, diverse, sometimes spurious, information otherwise not available. It is an easy and conducive platform for people of all walks of life participating, sharing, and interacting with a large number of users anytime and anywhere. Many users likely have multiple accounts at different social media sites to serve their disparate needs. When false information (e.g., incorrect age) is provided, information inconsistencies likely arise across sites, as well depicted in the saying, "a thousand lies are needed to hide one lie". Detecting these inconsistencies can help provide a first line of security toward solving the age verification problem. One way to detect these inconsistencies is to start connecting the different identities of a user across social media sites. For example, if a user has multiple user accounts that are associated with inconsistent profile information, a further investigation should be warranted to verify the individual's claimed age.

Connecting user identities across social media sites is not a straightforward task. The primary obstacle is that connectivity among user identities across different sites is often unavailable. This disconnection happens since most sites maintain the anonymity of users by allowing them to freely select usernames instead of their real identities, and also because different websites employ different user-naming and authentication systems. Moreover, websites rarely link their user accounts with other sites or adopt Single-Sign-On technologies such as openID, where users can logon to different sites using a single username (e.g., users can login to Google+ and YouTube with their GMail accounts). Regardless, there exists a mapping between usernames across different sites that connects the real identities behind them. *Can we find this mapping?*

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.
Copyright 20XX ACM 978-1-4503-2174-7/13/08 ...$15.00.

In this paper, we introduce a methodology (MOBIUS) for finding the mapping among identities across social media sites. Our methodology is based on behavioral patterns that users exhibit in social media, and has roots in behavioral theories in sociology and psychology. Unique behaviors due to environment, personality, or even human limitations can create redundant information across social media sites. Our methodology exploits such redundancies to identify users across social media sites. We use the minimum amount of information available across sites.

Section 2 formally presents the user identification problem across social media sites. Section 3 describes behavioral patterns that users exhibit in social media that can be harnessed to develop a user identification technique. Section 4 details experiments for identifying corresponding identities, followed by related work in Section 5. Section 6 concludes this research with directions for future work.

2. PROBLEM STATEMENT

Information shared by users on social media sites provides a *social fingerprint* of them and can help identify users across different sites. We start with the *minimum* amount of information that is available on *all* sites. In terms of information availability, *usernames* seem to be the minimum common factor available on all social media sites. Usernames are often alphanumeric strings or email addresses, without which users are incapable of joining sites. Usernames are unique on each site and can help identify individuals, whereas most personal information, even "first name + last name" combinations, are non-unique. We formalize our problem by using usernames as the atomic entities available across all sites. Other profile attributes, such as gender, location, interests, profile pictures, language, etc., when added to usernames, should help better identify individuals; however, the lack of consistency in the available information across all social media, directs us toward formulating with usernames. When considering usernames, two general problems need to be solved for user identification:

I. Given two usernames u_1 and u_2, can we determine if they belong to the same individual?

II. Given a single username u from individual $\mathcal{I}$, can we find other usernames of $\mathcal{I}$?

Question **I** can be answered in two steps: 1) we find the set of all usernames C that are likely to belong to individual $\mathcal{I}$. We denote set C as *candidate usernames* and, 2) for all candidate usernames $c \in C$, we check if c and u belong to the same individual. Hence, if candidate usernames C are known, question **II** reduces to question **I**. Since finding candidate usernames has been discussed in detail in [19], from now on, we focus on question **I**. One can answer question **I** by learning an *identification function* $f(u, c)$,

$$f(u,c) = \begin{cases} 1 & \text{If } c \text{ and } u \text{ belong to same } \mathcal{I} \text{ ;} \\ 0 & \text{Otherwise.} \end{cases} \quad (1)$$

Without loss of generality, we can assume that username u is known to be owned by some individual $\mathcal{I}$ and c is the candidate username whose ownership by $\mathcal{I}$ we would like to verify. In other words, u is the prior information (history) provided for $\mathcal{I}$. Our function can be generalized by assuming that our prior is a *set* [1] of usernames $U = \{u_1, u_2, \ldots, u_n\}$ (hereafter referred to as "prior usernames"). Informally, the usernames of an individual on some sites are given and we have a candidate username on another site whose ownership we need to verify; e.g., usernames u_t and u_f of someone are given on Twitter and Facebook, respectively; can we verify if c is her username on Flickr?

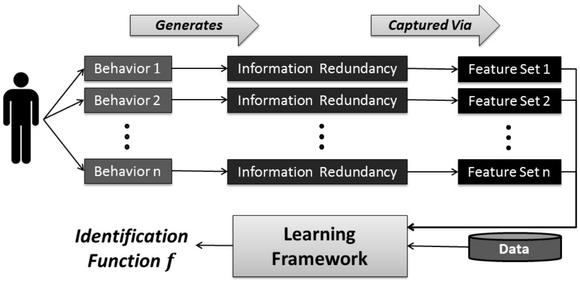

Figure 1: MOBIUS: Modeling Behavior for Identifying Users across Sites

Definition. ***User Identification across Social Media Sites.*** *Given a set of n usernames (prior usernames) $U = \{u_1, u_2, \ldots, u_n\}$, owned by individual $\mathcal{I}$ and a candidate username c, a user identification procedure attempts to learn an identification function $f(.,.)$ such that*

$$f(U,c) = \begin{cases} 1 & \textit{If } c \textit{ and set } U \textit{ belong to } \mathcal{I} \text{ ;} \\ 0 & \textit{Otherwise.} \end{cases} \quad (2)$$

Our methodology for ***MO**deling **B**ehavior for **I**dentifying **U**sers across **S**ites* (**MOBIUS**) [2] is outlined in Figure 1. When individuals select usernames, they exhibit certain behavioral patterns. This often leads to *information redundancy*, helping learn the identification function. In MOBIUS, these redundancies can be captured in terms of data features. Following the tradition in machine learning and data mining research, we can learn an identification function by employing a supervised learning framework that utilizes these features and prior information (*labeled data*). In our case, the labeled data is sets of usernames with known owners. Supervised learning in MOBIUS can be performed via either classification or regression. Depending on the learning framework, one can even learn the probability that an individual owns the candidate username, generalizing our binary f function to a probabilistic model ($f(U, c) = p$). This probability can help select the most likely individual who owns the candidate username. The learning component of MOBIUS is the most straightforward. Therefore, we next elaborate how to analyze behavioral patterns related to user identification and how features can be constructed to capture information redundancies due to these patterns. To summarize, MOBIUS contains 1) *behavioral patterns*, 2) *features* constructed to capture information redundancies due to these patterns, and 3) a *learning* framework. Due to the interdependent nature of behaviors and feature construction, we discuss them together next.

[1] Mathematically, a set can only contain distinct values; however, here a user may use the same username on more than one site. In our definition of username set, it is implied that usernames are distinct when used on different sites, even though they can consist of the same character sequence.

[2] The resemblance to the Möbius strip comes from its *single-boundary* (representing a single individual) and its *connectedness* (representing connected identities of the individual across social media).

3. MOBIUS: BEHAVIORAL PATTERNS AND FEATURE CONSTRUCTION

Individuals often exhibit consistent behavioral patterns while selecting their usernames. These patterns result in information redundancies that help identify individuals across social media sites.

Individuals can avoid such redundancies by selecting usernames on different sites in a way such that they are completely different from their other usernames. In other words, their usernames are so different that given one username, no information can be extracted regarding the others. Theoretically, to achieve these independent usernames, one needs to select a username with Maximum Entropy [6]. That is, a **long** username string, as long as the site allows, with characters from those that the system permits, with **no redundancy** - an entirely **random** string.

Unfortunately, all of these requirements are contrary to human abilities. Humans have difficulty storing long sequences with short-term memory capacity of 7 ± 2 items [18]. Human memory also has limited capability in storing random content and often, selectively stores content that contains familiar items, known as "chunks" [18]. Finally, human memory thrives on redundancy, and humans can remember material that can be encoded in multiple ways [18]. These limitations result in individuals selecting usernames that are generally *not long*, *not random*, and have *abundant redundancy.* These properties can be captured using specific features which in turn can help learn an identification function. In this study, we find a set of consistent behavioral patterns among individuals while selecting usernames. These behavioral patterns can be categorized as follows:

1. **Patterns due to Human Limitations**
2. **Exogenous Factors**
3. **Endogenous Factors**

The features designed to capture information generated by these patterns can be divided into three categories:

1. **(Candidate) Username Features**: these features are extracted directly from the candidate username c, e.g., its length,
2. **Prior-Usernames Features**: these features describe the set of prior usernames of an individual, e.g., the number of observed prior usernames, and
3. **Username↔Prior-Usernames Features**: these features describe the relation between the candidate username and prior usernames, e.g., their similarity.

We will discuss behaviors in each of the above mentioned categories, and features that can be designed to harness the information hidden in usernames as a result of the pattern's existence. Note that these features may or may not help in learning an identification function. As long as these features could be obtained for learning the identification function, they are added to our feature set. Later on in Section 4, we will analyze the effectiveness of all features, and if it is necessary to find as many features as possible.

3.1 Patterns due to Human Limitations

In general, as humans, we have 1) *limited time and memory* and 2) *limited knowledge.* Both create biases that can affect our username selection behavior.

3.1.1 Limitations in Time and Memory

Selecting the Same Username. As studied recently [19], 59% of individuals prefer to use the same username(s) repeatedly, mostly for ease of remembering. Therefore, when a candidate username c is among prior usernames U, that is a strong indication that it may be owned by the same individual who also owns the prior usernames. As a result, we consider the number of times candidate username c is repeated in prior usernames as a feature.

Username Length Likelihood. Similarly, users commonly have a limited set of potential usernames from which they select one, once asked to create a new username. These usernames have different lengths and, as a result, a *length distribution* $\mathcal{L}$. Let l_c be the candidate username length and l_u be the length for username $u \in U$ (prior usernames). We believe that for any new username, it is more likely to have,

$$\min_{u\in U} l_u \leq l_c \leq \max_{u\in U} l_u; \tag{3}$$

for example, if an individual is inclined to select usernames of length 8 or 9, it is unlikely for the individual to consider creating usernames with lengths longer or shorter than that. Therefore, we consider the candidate username's length l_c and the length distribution $\mathcal{L}$ for prior usernames as features. The length distribution can be compactly represented by a fixed number of features. We describe distribution $\mathcal{L}$, observed via discrete values $\{l_u\}_{u\in U}$ as a 5-tuple feature,

$$(\mathbb{E}[l_u], \sigma[l_u], med[l_u], \min_{u\in U} l_u, \max_{u\in U} l_u), \tag{4}$$

where $\mathbb{E}$ is the mean, σ is the standard deviation, and *med* is the median of the values $\{l_u\}_{u\in U}$, respectively. Note that this procedure for compressing distributions as a fixed number of features can be employed for discrete distributions $\mathcal{D}$, observed via discrete values $\{d_i\}_{i=1}^{n}$.

Unique Username Creation Likelihood. Users often prefer not to create new usernames. One might be interested in the effort users are willing to put into creating new usernames. This can be approximated by the number of unique usernames ($uniq(U)$) among prior usernames U,

$$uniqueness = \frac{|uniq(U)|}{|U|}. \tag{5}$$

Uniqueness is a feature in our feature set. One can think of $1/uniqueness$ as an individual's *username capacity*, i.e., the average number of times an individual employs a username on different sites before deciding to create a new one.

3.1.2 Knowledge Limitation

Limited Vocabulary. Our vocabulary is limited in any language. It is highly likely for native speakers of a language to know more words in that language than individuals speaking it as a second language. We assume the individual's vocabulary size in a language is a feature for identifying them, and, as a result, we consider the number of dictionary words that are substrings of the username as a feature. Similar to *username length* feature, the number of dictionary words in the candidate username is a scalar; however, when counting dictionary words in prior usernames, the outcome is a distribution of numbers. We employ the technique outlined in Eq. (4) for compressing distributions to represent this distribution as features.

Limited Alphabet. Unfortunately, it is a tedious task to consider dictionary words in all languages, and this feature can be used for a handful of languages. However, we observe that the alphabet letters used in the usernames are highly dependent on language. For instance, while the letter x is common when a Chinese speaker selects a username, it is rarely used by an Arabic speaker, since no Arabic word transcribed in English contains the letter x. Thus, we consider the number of alphabet letters used as a feature, both for the candidate username as well as prior usernames.

3.2 Exogenous Factors

Exogenous factors are behaviors observed due to cultural influences or the environment that the user is living in.

Typing Patterns. One can think of keyboards as a general constraint imposed by the environment. It has been shown [9] that the layout of the keyboard significantly impacts how random usernames are selected; e.g., `qwer1234` and `aoeusnth` are two well-known passwords commonly selected by QWERTY and DVORAK users, respectively. Most people use one of two well-known keyboards DVORAK and QWERTY (or slight variants such as QWERTZ or AZERTY) [17]. To capture keyboard-related regularities, we construct the following 15 features for each keyboard layout (a total of 30 for both):

1. (1 feature) The percentage of keys typed using the *same hand* used for the previous key. The higher this value the less users had to change hands for typing.
2. (1 feature) Percentage of keys typed using the *same finger* used for the previous key.
3. (8 features) The percentage of keys typed using each finger. Thumbs are not included.
4. (4 features) The percentage of keys pressed on rows: Top Row, Home Row, Bottom Row, and Number Row. Space bar is not included.
5. (1 feature) The approximate *distance* (in meters) traveled for typing a username. Normal typing keys are assumed to be $(1.8\text{cm})^2$ (including gap between keys).

We construct these features for candidate username and each prior username. Thus, for all prior usernames, each feature has a set of values. Adopting the technique outlined in Eq. (4) for compressing distributions as features, we construct $15 \times 5 = 75$ additional features for prior usernames.

Language Patterns. In addition to environmental factors, cultural priors such as language also affect the username selection procedure. Users often use the same or the same set of languages when selecting usernames. Therefore, when detecting languages of different usernames belonging to the same individual, one expects fairly consistent results. We consider the language of the username as a feature in our dataset. To detect the language, we trained an n-gram statistical language detector [10] over the European Parliament Proceedings Parallel Corpus [3], which consists of text in 21 European languages (*Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene,* and *Swedish*) from 1996-2006 with more than 40 million words per language. The trained model detects the candidate username language, which is a feature in our feature set. The language detector is also used on prior usernames, providing us with a language distribution for prior usernames, which again is compressed as features using Eq. (4). The *detected language* feature is limited to European languages. Our language detector will not detect other languages. The language detector is also challenged when dealing with words that may not follow the statistical patterns of a language, such as location names, etc. However, these issues can be tackled from a different angle, as we discuss next.

3.3 Endogenous Factors

Endogenous factors play a major role when individuals select usernames. Some of these factors are due to 1) personal attributes (name, age, gender, roles and positions, etc.) and 2) characteristics, e.g., a female selecting username `fungirl09`, a father selecting `geekdad`, or a PlayStation 3 fan selecting `PS3lover2009`. Others are due to 3) habits, such as abbreviating usernames or adding prefixes/suffixes.

3.3.1 Personal Attributes and Personality Traits

Personal Information. As mentioned, our language detection model is incapable of detecting several languages, as well as specific names, such as locations, or others that are of specific interest to the individual selecting the username. For instance, the language detection model is incapable of detecting the language of usernames `Kalambo`, a waterfall in Zambia, or `K2` and `Rakaposhi`, both mountains in Pakistan. However, the patterns in these words can be captured by analyzing the alphabet distribution. For instance, a user selecting username `Kalambo` most of the time will create an alphabet distribution where letter 'a' is repeated twice more than other letters. Hence, we save the alphabet distribution of both candidate username and prior usernames as features. This will easily capture patterns like an excessive use of 'i' in languages such as Arabic or Tajik [7, 11], where language detection fails. Another benefit of using alphabet distribution is that not only is it language-independent, but it can also capture words that are meaningful only to the user.

Username Randomness. As mentioned before, individuals who select totally random usernames generate no information redundancy. One can quantify the randomness of usernames of an individual and consider that as a feature that can describe individuals' level of privacy and help identify them. For measuring randomness, we consider the entropy [6] of the candidate username's alphabet distribution as a feature. We also measure entropy for each prior username. This results in an entropy distribution that is encoded as features using aforementioned technique in Eq. (4).

3.3.2 Habits

"Old habits, die hard", and these habits have a significant effect on how usernames are created. Common habits are,

Username Modification. Individuals often select new usernames by changing their previous usernames. Some,

1. add prefixes or suffixes,
   - e.g., `mark.brown` → `mark.brown2008`,
2. abbreviate their usernames,
   - e.g., `ivan.sears` → `isears`, or
3. change characters or add characters in between.
   - e.g., `beth.smith` → `b3th.smith`.

Any combination of these operations is also possible. The following approaches are taken to capture the modifications:

[3] http://www.statmt.org/europarl/

- To detect added prefixes or suffixes, one can check if one username is the substring of the other. Hence, we consider the length of the *Longest Common Substring (LCS)* as an informative feature about how similar the username is to prior usernames. We perform a pairwise computation of LCS length between the candidate username and all prior usernames. This will generate a distribution of LCS length values, quantizied as features using Eq. (4). To get values in range [0,1], we also perform a normalized LCS (normalized by the maximum length of the two strings) and store the distribution as a feature as well.
- For detecting abbreviations, *Longest Common Subsequence* length is used since it can detect non-consecutive letters that match in two strings. We perform a pairwise calculation of it between the candidate username and prior usernames and store the distribution as features using aforementioned technique in Eq. (4). We also store the normalized version as another distribution feature.
- For swapped letters and added letters, we use the normalized and unnormalized versions of both Edit (Levenshtein) Distance, and Dynamic Time Warping (DTW) distance as measures. Again, the end results are distributions, which are saved as features.

Generating Similar Usernames. Users tend to generate similar usernames. The similarity between usernames is sometimes hard to capture using approaches discussed for detecting username modification. For instance, `gateman` and `nametag` are highly similar due to one being the other spelled backward, but their similarity is not recognized by discussed methods. Since we store the alphabet distribution for both the candidate username and prior usernames, we can compare these using different similarity measures. The Kullback-Liebler divergence (KL) [6] is commonly the measure of choice; however, since KL isn't a metric, comparison among values becomes difficult. To compare distributions, we use the Jensen-Shannon divergence (JS) [13], which is a metric computed from KL,

$$JS(P||Q) = \frac{1}{2}[KL(P||M) + KL(Q||M)], \tag{6}$$

where $M = \frac{1}{2}(P + Q)$, and KL divergence is

$$KL(P||Q) = \sum\nolimits_{i=1}^{|P|} P_i \cdot log(\frac{P_i}{Q_i}). \tag{7}$$

Here, P and Q are the alphabet distributions for the candidate username and prior usernames. As an alternative, we also consider cosine similarity between the two distributions as a feature. Note that Jensen-Shannon divergence does not measure the overlap between the alphabets. To compute alphabet overlaps, we add Jaccard Distance as a feature.

Username Observation Likelihood. Finally, we believe the order in which users juxtapose letters to create usernames depends on their prior knowledge. Given this prior knowledge, we can estimate the probability of observing candidate username. Prior knowledge can be gleaned based on how letters come after one another in prior usernames. In statistical language modeling, the probability of observing username `u`, denoted in characters as $u = c_1c_2 \ldots c_n$, is

$$p(u) = \Pi_{i=1}^{n} p(c_i|c_1c_2 \ldots c_{i-1}). \tag{8}$$

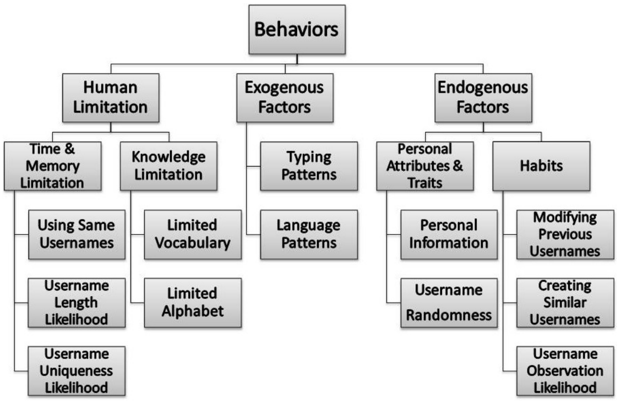

Figure 2: Individual Behavioral Patterns when Selecting Usernames

We approximate this probability using an n-gram model,

$$p(u) \approx \Pi_{i=1}^{n} p(c_i|c_{i-(n-1)} \ldots c_{i-1}). \tag{9}$$

Commonly, to denote the beginning and the end of a word, special symbols are added: $\star$ and $\bullet$. So, for username `jon`, the probability approximated using a 2-gram model is

$$p(jon) \approx p(j|\star)p(o|j)p(n|o)p(\bullet|n). \tag{10}$$

To estimate the observation probability of the candidate username using an n-gram model, we first need to compute the probability of observing its comprising n-grams. The probability of observing these n-grams can be computed using prior usernames. These probabilities are often hard to estimate, since some letters never occur after others in prior usernames while appearing in the candidate username. For instance, for candidate username `test12` and prior usernames {`test, testing`}, the probability of $p(\texttt{1}|\star\texttt{test}) = 0$ and therefore $p(\texttt{test12}) = 0$, which seems unreasonable. To estimate probabilities of unobserved n-grams, a smoothing technique can be used. We use the state-of-the-art *Modified Kneser-Ney (MKN)* smoothing technique [4], which has discount parameters for n-grams observed once, twice, and three times or more. The discounted values are then distributed among unobserved n-grams. The model has demonstrated excellent performance in various domains [4]. We include the candidate username observation probability, estimated by an MKN-smoothed 6-gram model, as a feature.

We have demonstrated how behavioral patterns can be translated into meaningful features for the task of user identification. These features are constructed to mine information hidden in usernames due to individual behaviors when creating usernames. Overall, we construct 414 features for the candidate username and prior usernames. Figure 2 depicts a summary of these behavioral patterns observed in individuals when selecting usernames.

Clearly, our features do not cover all aspects of username creation, and with more theories and behaviors in place, more features can be constructed. We will empirically study if it is necessary to use all features and the effect of adding more features on learning performance of user identification.

Following MOBIUS methodology, we compute the feature values over labeled data, and verify the effectiveness of MOBIUS by learning an identification function. Next, experiments for evaluating MOBIUS are detailed.

4. EXPERIMENTS

The MOBIUS methodology is systematically evaluated in this section. First, we verify if MOBIUS can learn an accurate identification function, comparing with some baselines. Second, we examine if different learning algorithms make a significant difference in learning performance using acquired features. Then, we perform feature importance analysis, and investigate how the number of usernames and the number of features impact learning performance. Before we present our experiments, we detail how experimental data is collected.

4.1 Data Preparation

A simple method for gathering identities across social networks is to conduct surveys and ask users to provide their usernames across social networks. This method can be expensive in terms of resource consumption, and the amount of gathered data is often limited. Companies such as Yahoo! or Facebook ask users to provide this kind of information; however, this information is not publicly available.

Another method for identifying usernames across sites is by finding users manually. Users, more often than not, provide personal information such as their real names, E-mail addresses, location, gender, profile photos, and age on these websites. This information can be employed to map users on different sites to the same individual. However, manually finding users on sites can be quite challenging.

Fortunately, there exist websites where users have the opportunity of listing their identities (user accounts) on different sites. This can be thought of as *labeled* data for our learning task, providing a mapping between identities. In particular, we find social networking sites, blogging and blog advertisement portals, and forums to be valuable sources for collecting multiple identities of the same user.

Social Networking Sites. On most social networking sites such as Google+ or Facebook, users can list their IDs on other sites. This provides usernames of the same individual on different sites.

Blogging and Blog Advertisement Portals: To advertise their blogs, individuals often join *blog cataloging* sites to list not only blogs, but also their profiles on other sites. For instance, users in BlogCatalog are provided with a feature called "My Communities". This feature allows users to list their usernames in other social media sites.

Forums: Many forums use generic Content Management Systems (CMS), designed specifically for creating forums. These applications usually allow users to add their usernames on social media sites to their profiles. Examples of these applications that contain this feature include, but are not limited to: vBulletin, phpBB, and Phorum.

We utilize these sources for collecting usernames, guaranteed to belong to the same individual. Overall, 100,179 (c-U) pairs are collected, where c is a username and U is the set of prior usernames. Both c and U belong to the same individual. The dataset contains usernames from 32 sites, such as Flickr, Reddit, StumbleUpon, and YouTube.

The collected pairs are considered as positive instances in our dataset. For negative instances, we construct instances by randomly creating pairs (c_i-U_j), such that c_i is from one positive instance and U_j is from a different positive instance ($i \neq j$) to guarantee that they are not from the same individual. We generated different numbers of negative instances (up to 1 million instances), but its effect on the accuracy of learning the identification function was negligible, so we continue with a dataset where the class balance is 50% for each label (100,179 positive + 100,179 negative ≈200,000 instances). Then, we compute our 414 feature values for this data and employ this dataset for our learning framework.

4.2 Learning the Identification Function

To evaluate MOBIUS, the first step is to verify if it can learn an accurate identification function. Given our labeled dataset where all feature values are calculated, learning the identification function can be realized by performing supervised learning on our dataset. We mentioned earlier that a probabilistic classifier can generalize our binary identification function to a probabilistic one, where the probability of a candidate username belonging to an individual is measured. Probabilistic classification can be achieved by a variety of Bayesian approaches. We select Naive Bayes. Naive Bayes, using 10-fold cross validation, correctly classifies 91.38% of our data instances.

There is a need to compare MOBIUS performance to other methods as well. To the best of our knowledge, methods from Zafarani et al. [19] and Perito et al. [15] are the only methods that tackle the same problem. The ad hoc method of Zafarani et al. employs two features: 1) exact match between usernames and 2) substring match between usernames. Perito et al.'s method uses a single feature. This feature, similar to our username-observation likelihood, utilizes a 1-gram model to compute the username observation probability. Table 1 reports the performance of these techniques over our datasets. Our method outperforms the method of Zafarani et al. by 38% and the method of Perito et al. by 18%. The key difference between MOBIUS and the methods in comparison is that MOBIUS takes a behavioral modeling approach that systematically generates features for effective user identification.

To evaluate the effectiveness of MOBIUS, we also devise three baseline methods for comparison. When people are asked to match usernames of individuals, commonly used methods are "exact username matching", "substring matching", or finding "patterns in letters". Hence, they form our three baselines, b_1, b_2, and b_3:

b_1**: Exact Username Match.** It considers an instance positive if the candidate username is an exact match to α% of the prior usernames. To set α accurately, we computed the percentage of prior usernames that are exact matches to the candidate username in each of our positive instances and averaged it over all positive instances to get α, $\alpha \approx 54\%$. To further analyze the impact, we set $50\% \leq \alpha \leq 100\%$. Among all α values, b_1 does not perform better than 77%.

b_2**: Substring Matching.** It considers an instance positive if the mean of the candidate username's normalized longest common substring distance to prior usernames is below some threshold θ. We conduct the experiment for the range $0 \leq \theta \leq 1$. In the best case, b_2 achieves 63.12% accuracy.

b_3**: Patterns in Letters.** For finding letter patterns, b_3 uses the alphabet distribution for the candidate username and the prior usernames as features. Using our data labels, we perform logistic regression. b_3 achieves 49.25% accuracy.

Our proposed technique outperforms baseline b_1, b_2, and b_3 by 19%, 45%, and 86%, respectively. The performance for Naive Bayes, other methods, and baselines are summarized

Table 1: MOBIUS Performance

Technique	Accuracy
MOBIUS (Naive Bayes)	91.38%
Method of Zafarani et al. [19]	66.00%
Method of Perito et al. [15]	77.59%
Baseline b_1: Exact Username Match	77.00%
Baseline b_2: Substring Matching	63.12%
Baseline b_3: Patterns in Letters	49.25%

Table 2: MOBIUS Performance for Different Classification Techniques

Technique	AUC	Accuracy
J48 Decision Tree Learning	0.894	90.87%
Naive Bayes	0.937	91.38%
Random Forest	0.957	93.59%
ℓ_2-Regularized ℓ_2-Loss SVM	0.950	93.70%
ℓ_1-Regularized ℓ_2-Loss SVM	0.951	93.71%
ℓ_2-Regularized Logistic Regression	0.950	93.77%
ℓ_1-Regularized Logistic Regression	0.951	**93.80%**

in Table 1. Now, we would like to see if different learning algorithms can further improve the learning performance.

4.3 Choice of Learning Algorithm

To evaluate the choice of learning algorithm, we perform the classification task using a range of learning techniques and 10-fold cross validation. The AUCs and accuracy rates are available in Table 2. These techniques have different learning biases, and one expects to observe different performances for the same task. As seen in the table, results are not significantly different among these methods. This shows that when sufficient information is available in features, the user identification task becomes reasonably accurate and is not sensitive to the choice of learning algorithm. In our experiments, ℓ_1-Regularized Logistic Regression is shown to be the most accurate method; hence, we use it in the following experiments as the method of choice. The classification employs all 414 features. Designing 414 features and computing their values is computationally expensive. Therefore, we try to empirically determine 1) whether all features are necessary, and 2) whether it makes *economic* sense to add more features, in Sections 4.4 and 4.5.

4.4 Feature Importance Analysis

Feature Importance Analysis analyzes how important different features are in learning the identification function. In other words, it finds features that contribute the most to the classification task. This can be performed by standard feature selection measures such as Information Gain, χ^2, among others. We utilize *odds-ratios* (logistic regression coefficients) for feature importance analysis and ranking features. The top 10 important features are as follows:

1. Standard deviation of normalized edit distance between the candidate username and prior usernames,
2. Standard deviation of normalized longest common substring between the username and prior usernames,
3. Username observation likelihood,
4. Uniqueness of prior usernames,
5. Exact match: number of times candidate username is seen among prior usernames,
6. Jaccard similarity between the alphabet distribution of the candidate username and prior usernames,
7. Standard deviation of the distance traveled when typing prior usernames using the QWERTY keyboard,
8. Distance traveled when typing the candidate username using the QWERTY keyboard,
9. Standard deviation of the longest common substring between the username and prior usernames, and
10. Median of the longest common subsequence between the candidate username and prior usernames.

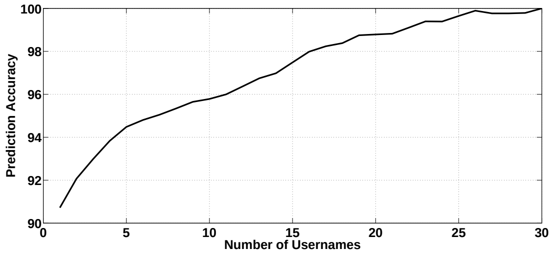

Figure 3: User Identification Performance for Users with Different Number of Usernames

In fact, a classification using only these 10 features and logistic regression provides an accuracy of 92.72%, which is very close to that of using the entire feature set. We also notice that in our ranked features,

- Numbers `[0-9]` are on average ranked higher than English alphabet letters `[a-z]`, showing that numbers in usernames help better identify individuals, and
- Non-English alphabet letters or special characters, e.g., $\hat{A}$,$\tilde{A}$,+, or &, are among the features that could easily help identify individuals across sites, i.e., have higher odds-ratios on average.

Although these 10 features perform reasonably well, it is of practical importance to analyze how we can further improve the performance of our methodology in different scenarios, such as by adding usernames or features.

4.5 Diminishing Returns for Adding More Usernames and More Features

It is often assumed that when more prior usernames of an individual are known, the task of identifying the individual becomes easier. If true, to improve identification performance, we need to provide MOBIUS with extra prior information (known usernames). In our dataset, users have from 1 to a maximum of 30 prior usernames. To verify helpfulness of adding prior usernames, we partition the dataset into 30 datasets $\{d_i\}_{i=1}^{30}$, where dataset d_i contains individuals that have i prior usernames. The user identification accuracy on these 30 datasets are shown in Figure 3. We observe a monotonically increasing trend in identification performance, and even for a single prior username, the identification is 90.72% accurate and approaches 100% when 25 or more usernames are available. Note that the identification task is hardest when only a single prior username is available.

Rarely are 25 prior usernames of an individual available across sites. It is more practical to know the minimum number of usernames required for user identification such that further improvements are nominal. The relative performance improvement with respect to number of usernames

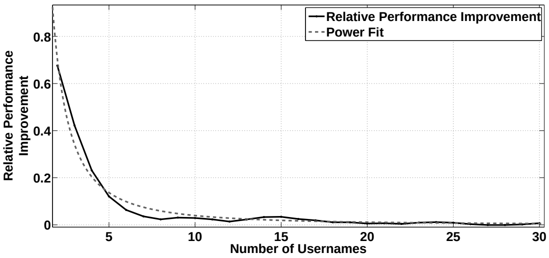

Figure 4: Relative User Identification Performance Improvement with respect to Number of Usernames

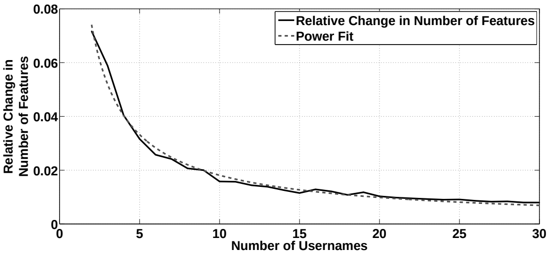

Figure 5: Relative Change in Number of Features Required with respect to Number of Usernames

can help us measure this minimum. Figure 4 shows this improvement for adding usernames. We observe a *diminishing return* property, where the improvement becomes marginal as we add usernames and is negligible for more than 7 usernames. A power function ($g(x) = 2.44x^{-1.79}$), found with 95% confidence, fits to this curve with adjusted $R^2 = 0.976$. The exponent -1.79 denotes that the relative improvement by adding n usernames is $\approx 1/n^{1.79}$ times smaller than that by adding a single username, e.g., for 7 usernames, relative identification performance improvement is $\approx 1/33$ times smaller than that of a single username.

Similar to adding more prior usernames, one can change number of features. More practically, we would like to analyze how adding features correlates with adding prior usernames. For instance, if we double the number of prior usernames, how many features should we construct (or can be removed) to guarantee reaching a required performance?

To measure this, for each number of prior usernames n, we compute the average number of features such that MOBIUS can achieve fixed accuracy θ. We set θ to the minimum accuracy achievable, independent of number of usernames (90% here). We then compute the relative change in the number of required features when usernames are added.

Figure 5 plots this relationship. We observe the same diminishing return property, and as one adds more usernames, fewer features are required to achieve a fixed accuracy. A power function ($g(x) = 0.1359x^{-0.875}$), found with 95% confidence, fits to this curve with adjusted $R^2 = 0.987$. The exponent -0.875 denotes that the number of features required for n usernames is $\approx 1/n^{0.875}$ times smaller than that of a single username.

Finally, if one is left with a set of usernames and a set of features, should we aim at adding more usernames or construct better features? Let $f(n, k)$ denote the performance

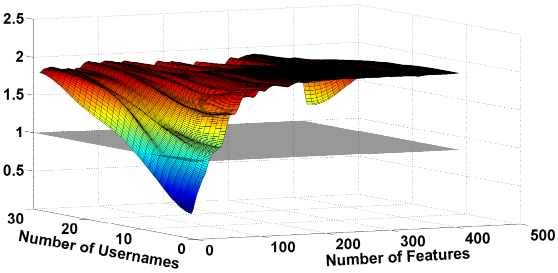

Figure 6: The $\delta(n, k)$ function, for n usernames and k features. Values larger than 1 show that adding usernames will improve performance more and values smaller than 1 show adding features is better.

of our method for n usernames and k features. Let

$$\delta(n, k) = \frac{f(n+1, k) - f(n, k)}{f(n, k+1) - f(n, k)}. \tag{11}$$

The δ function is a finite difference approximation for the derivative ratio with respect to n and k. When $\delta(n, k) > 1$, adding usernames improves performance more and when $\delta(n, k) < 1$, adding features is better. To compute $f(n, k)$, for different values of n, we select random subsets of size k. We denote the average performance over these random subsets as $f(n, k)$. Figure 6 plots the $\delta(n, k)$ function. We plot plane $z = 1$ to better show where adding features is more helpful and where usernames are more beneficial. We observe that for small values of n and k, i.e., when fewer usernames and features are available, features help best, but for all other cases adding usernames is more beneficial.

5. RELATED WORK

In this section, we focus on summarizing research related to identifying individuals in social media. We provided a review of directly relevant techniques to our study in Section 4. In addition to those methods, there exists related research about 1) *identifying content produced by an individual on the web* or 2) *identifying individuals in a single social network*.

Identifying Content Authorship. In [1], the authors look at the content generation behavior of the same individuals in several collections of documents. Based on the overlap between contributions, they propose a method for detecting pages created by the same individual across different collections of documents. They use a method called detection by compression, where Normalized Compression Distance (NCD) [5] is used to compare the similarity between the documents already known to be authored by the individual and other documents. Author detection has been well discussed in restricted domains. In particular, machine learning techniques have been employed to detect authors in online messages [20] and in E-mails [8]. Although, one can think of usernames as the content generated by individuals across sites; however, in content authorship detection, it is common to assume large collections of documents, with thousands of words, available for each user, whereas for usernames, the information available is limited to one word.

User Identification on One Site. Deanonymization is an avenue of research related to identifying individuals on a single site. Social networks are commonly represented using graphs where nodes are the users and edges are the connections. To preserve privacy, an anonymization process replaces these users with meaningless, randomly generated, unique IDs. To identify these masked users, a deanonymization technique is performed. Deanonymization of social networks is tightly coupled with the research in privacy preserving data mining or Identity Theft attacks [3]. In [2], Backstrom et al. present such process where one can identify individuals in these anonymized networks by either manipulating networks before they are anonymized or by having prior knowledge about certain anonymized nodes. Narayanan and Shmatikov in [14] present statistical deanonymization technique against high-dimensional data. They argue that given little information about an individual one can easily identify the individual's record in the dataset. They demonstrate the performance of their method by uncovering some users on the Netflix prize dataset using IMDB information as their source for background knowledge. Our work differs from these techniques, as it deals with multiple sites. Moreover, it avoids using link information, which is not always available on different social media sites.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated a methodology for connecting individuals across social media sites (MOBIUS). MOBIUS takes a behavioral modeling approach for systematic feature construction and assessment, which allows integration of additional features when required. MOBIUS employs minimal information available on all social media sites (usernames) to derive a large number of features that can be used by supervised learning to effectively connect users across sites. Users often exhibit certain behavioral patterns when selecting usernames. The proposed behavioral modeling approach exploits information redundancy due to these behavioral patterns. We categorize these behavioral patterns into (1) human limitations, (2) exogenous factors, and (3) endogenous factors. In each category of behaviors, various features are constructed to capture information redundancy. MOBIUS employs supervised learning to connect users. Our empirical results show the advantages of this principled, behavioral modeling approach over earlier methods. The experiments demonstrate that (1) constructed features contain sufficient information for user identification; (2) importance or relevance of features can be assessed, thus features can be selected based on particular application needs; and (3) adding more features can further improve learning performance but with diminishing returns; hence, facing a limited budget, one can make informed decisions on what additional features should be added.

MOBIUS can help solve the problem of age verification in a collective effort of protecting youth on the web against predators. For example, profiles of individuals across sites can be connected and inconsistencies on reported age can be checked. Detecting these inconsistencies can help provide a first line of security toward solving the age verification problem. Identifying users across social media sites opens the door to many interesting applications. Studying user behaviors across social media such as user migration [12] is an example of the many areas that can benefit from the results of this study. Future work includes analyzing these possibilities and discovering features indigenous to specific sites, beyond those constricted to usernames, and incorporating them into MOBIUS for future needs.

ACKNOWLEDGMENTS

This work was supported, in part, by the Office of Naval Research grant: N000141110527.

7. REFERENCES

[1] E. Amitay, S. Yogev, and E. Yom-Tov. Serial Sharers: Detecting Split Identities of Web Authors. In *SIGIR PAN workshop*, 2007.

[2] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore Art Thou R3579X?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In *WWW*, pages 181–190. ACM, 2007.

[3] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In *WWW*, pages 551–560. ACM, 2009.

[4] S.F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In *ACL*, pages 310–318, 1996.

[5] R. Cilibrasi and P.M.B. Vitányi. Clustering by Compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[6] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-interscience, 2006.

[7] D. Cowan. *An Introduction to Modern Literary Arabic*, volume 240. Cambridge University Press, 1958.

[8] O. De Vel, A. Anderson, M. Corney, and G. Mohay. Mining E-mail Content for Author Identification Forensics. *ACM Sigmod Record*, 30(4):55–64, 2001.

[9] C. Doctorow. Preliminary Analysis of LinkedIn User Passwords. `http://bit.ly/L5AHo3`.

[10] T. Dunning. *Statistical Identification of Language*. CR Lab, New Mexico State University, 1994.

[11] C.A. Ferguson. Word Stress in Persian. *Language*, 33(2):123–135, 1957.

[12] S. Kumar, R. Zafarani, and H. Liu. Understanding User Migration Patterns in Social Media. In *AAAI*, pages 1204–1209, 2011.

[13] J. Lin. Divergence Measures based on the Shannon Entropy. *IEEE Transaction on Information Theory*, 37(1):145–151, 1991.

[14] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *IEEE SSP*, pages 111–125, 2008.

[15] Daniele Perito, Claude Castelluccia, Mohamed Kaafar, and Pere Manils. How unique and traceable are usernames? In *PETS*, pages 1–17, 2011.

[16] N. Perlroth. Verifying Ages Online is a Daunting Task, Even for Experts. `http://nyti.ms/Tf16Gs`.

[17] Wikipedia. Keyboard Layouts. `http://bit.ly/kXso`.

[18] J. Yan, A. Blackwell, R. Anderson, and A. Grant. The Memorability and Security of Passwords-Some Empirical Results. *U. of Cambridge Tech. Rep.*, 2000.

[19] R. Zafarani and H. Liu. Connecting Corresponding Identities across Communities. In *ICWSM*, pages 354–357, 2009.

[20] R. Zheng, J. Li, H. Chen, and Z. Huang. A Framework for Authorship Identification of Online Messages: Writing-style Features and Classification Techniques. *JASIST*, 57(3):378–393, 2006.