How Unique and Traceable are Usernames?

Daniele Perito, Claude Castelluccia, Mohamed Ali Kaafar, Pere Manils

INRIA Rhone Alpes, Montbonnot, France
{perito,ccastel,kaafar,manils}@inrialpes.fr

Abstract. Usernames are ubiquitously used for identification and authentication purposes on web services and the Internet at large, ranging from the local-part of email addresses to identifiers in social networks. Usernames are generally alphanumerical strings chosen by the users and, by design, are unique within the scope of a single organization or web service. In this paper we investigate the feasibility of using usernames to trace or link multiple profiles across services that belong to the same individual. The intuition is that the probability that two usernames refer to the same physical person strongly depends on the "entropy" of the username string itself. Our experiments, based on usernames gathered from real web services, show that a significant portion of the users' profiles can be linked using their usernames. In collecting the data needed for our study, we also show that users tend to choose a small number of related usernames and use them across many services. To the best of our knowledge, this is the first time that usernames are considered as a source of information when profiling users on the Internet.

1 Introduction

Online profiling is a serious threat to users privacy. In particular, the ability to trace users by linking multiple identities from different public profiles may be of great interest and commercial value to profilers, advertisers and the like. Indeed, it might be possible to gather information from different online services and combine it to sharpen the knowledge of users identities. This knowledge may then be exploited to perform efficient social phishing or targeted spam, and might be as well used by advertisers.

Recent scraping services' activities illustrate well the threats introduced by the ability to match up user's pseudonyms on different social networks [1]. For instance, PeekYou.com has lately applied for a patent for a way to match people's real names to pseudonyms they use on blogs, OSN services and online forums [11]. The methodology relies on public information collected for an user, that might help in matching different online identities. The algorithm empirically assigns weights to each of the collected information to link different identities to the same individual. However, the algorithm is ad-hoc and not robust to false or mismatching information. In light of these recent developments, it is desirable that the research community investigates the capabilities and limits of these profiling techniques. This will, in turn, allow for the design of appropriate countermeasures to protect users' privacy.

In general, profiling unique identities from multiple public profiles is a challenging task, as information from public profiles is often incorrect, misleading or altogether missing [9]. Techniques designed for the purpose of profiling need to be robust to these occurrences. Recent works [2,3] showed how it is possible to retrieve users information from different online social networks (OSN). All of these works mainly exploit flaws in the OSN's API design (e.g., Facebook friend search). Other approaches [14] use the topology of social network friend graphs to de-anonymize its nodes.

In this paper, we propose a novel methodology that uses usernames — an easy to collect information — to tie user online identities. In this context usernames offer the advantage of being used and publicly accessible on almost all current web services (e.g., Twitter, Facebook, eBay, Skype, etc.). The techniques developed in this work can link different user profiles only knowing their associated usernames and it is widely applicable to all web services that publicly expose usernames. Our purpose is to show that users' pseudonyms allow simple, yet efficient tracking of online activities.

This paper has several contributions. First, we introduce the problem of linking multiple online identities relying only on usernames. This problem is, to the best of our knowledge, novel and has not been exlpored in the literature.

Second, we devise an analytical model to estimate the *uniqueness* of a username, which can in turn be used to assign a probability that a single username, from two different online services, refers to the same user. Our tool can correctly classify a username like `sarah82` as non-identifying, also it can identify the username `dan.perito` as probably identifying. Based on language models and Markov Chain techniques, our model validates an intuitive observation: usernames with low "entropy" (or to be precise *Information Surprisal*) will have higher probabilities of being picked by multiple persons, whereas higher entropy usernames will be very unlikely picked by multiple users and refer in the vast majority of the cases to unique users.

Third, we extend this model to cases when usernames are *different* across many online services . In essence, given two usernames (e.g., `daniele.perito` and `d.perito`) our technique returns the probability that these usernames refer to the same user. We build a classifier upon this probability estimation that, given two usernames, can classify with high accuracy whether the usernames belong to the same individual. This tool could allows to effectively link and trace users identities across multiple web services using their usernames only. These results are tested and validated on real world data, 10 million usernames collected from eBay and Google Profiles.

Fourth, by studying the usernames from our dataset, we discover that users tend to choose their usernames from a small set and re-use them across different services. Also, we discover that users tend to choose usernames that are highly related to each other, like the aforementioned example `daniele.perito` and `d.perito`. These two findings give an explanation of the high accuracy of our tool on the task of linking public profiles using usernames.

We envision several possible uses of these techniques, not all of them malicious. In particular, users might use our tool to test how unique their username is and, therefore, take appropriate decision in case they wish to stay anonymous. To this extent we provide an online tool that can help users choose appropriate usernames by measuring how unique and traceable their usernames are. The tool is available at `http://planete.inrialpes.fr/projects/how-unique-are-your-usernames`. Furthermore, spammers could gather information across the web to send extremely targeted spam, which we dub *E-mail spam 2.0*. For example, by matching a Google profile and an eBay account spammers could send spam emails that mention a recent sale to lure users into a scam. In fact, while eBay profiles do not show much personal information (like real names) they do show recent transactions indexed by username. This would enable very targeted and efficient phishing attacks.

Paper organization. In Section 2, we overview the related work on privacy and introduce the machine learning tools used in our analysis. In Section 3, we introduce our measure to estimate the uniqueness of usernames and in Section 4, we extend our model to compute the probability that two usernames refer to the same person and validate it using the dataset we collected from eBay and Google (Section 2.3). Different techniques are introduced and evaluated. Finally, in Section 5 we discuss potential impact of our proposed techniques and present some possible countermeasures.

2 Related work and Background

2.1 Related Work

Tracking OSNs users In [9] the authors propose to use what they call the *online social footprint* to profile users on the Internet. This footprint would be the collection of all little pieces of information that each user leaves on web services and OSNs. While the idea is promising this appears to be only a preliminary work and no model, implementation or validation is given.

Similarly in [3], Bilge et al. discuss how to link the membership of users to two different online social networks. Noticing that there might be discrepancies in the information provided by a single user in two social networks, the authors rely on Google search results to decide the equivalence of selected fields of interest (as for assigning uniqueness of a user). Typically, the input of their algorithm is the name and surname of a user, that is augmented by the education/occupation as provided in two different social networks. They use such input to start two separate Google searches, and if both appear in the first top three hits, these are deemed to be equivalent. The corresponding users are consequently identified as a single user on both social networking sites. Bilge et al.'s work illustrates well how challenging the process of identifying users from multiple public profiles is. Despite the usage of customized crawler and parser for each social network, the heterogeneity of information as provided by users (if correct) makes the process hard to deploy, if not unfeasible, at a large scale.

Record linkage Record linkage (RL)(or alternatively Entity Resolution) [8,4] refers to the task of finding records that refer to the same entity in two or more databases. This is a common task when databases of users records are merged. For example, after two companies merge they might also want to merge their databases and find duplicate entries. Record linkage is needed in this case if there are no unique identifiers available (e.g., social security numbers). In RL terminology two records that have been matched are said to be *linked* (we will use the same term throughout this work). The application of record linkage techniques to link public online user profiles is novel to the best of our knowledge and presents several challenges of its own.

De-anonymizing sparse database and graph data [14] proposes an identification algorithm targeting anonymized social network graphs. The main idea of this work is to de-anonymize online social graph based on information acquired from a secondary social network users are known to belong to as well. Similarity identified in the network topologies of both services allows then to identify users belonging to the anonymized graph.

2.2 Background

Information Surprisal Self-information or Information Surprisal measures the amount of information associated to a specific outcome of a random variable. If X is a random variable and x one possible outcome, we denote the information surprisal of x as $I(x)$ [5]. Information Surprisal is computed as $I(u) = -\log_2(P(u))$ and hence depends only on the probability of x. The smaller the probability of x the higher is the associated surprisal. Entropy, on the other hand, measures the information associated to a random variable (regardless of any specific outcome), denoted $H(X)$. Entropy and Surprisal are deeply related as entropy can be seen as the expected value of the information surprisal, $H(X) = E(I(X))$. Both are usually measured in bits. Suppose there exists a discrete random variable that models the distribution of usernames in a population, call this variable U. The random variable U follows a probability mass function P_U that associates to each username u a probability $P(u)$. In this context, the information surprisal of $P(u)$ is the amount of identifying information associated to a username u. Every bit of surprisal adds one bit of identifying information and thus allows to cut the population in which u might lie in half.

If we assume that there are w users in a population, then a username u identifies *uniquely* a user in the population if $I(u) > \log_2(w)$. In this sense, information surprisal gives a measure of the "uniqueness" of a username u and it is the measure we are going to use in this work. The challenge lies in estimating the probability $P(u)$, which we will address in Section 3.

Our treatment of information surprisal and its association to privacy is similar to the one recently suggested in [7] in the context of fingerprinting browsers.

2.3 The Dataset

Our study was conducted on several different lists of usernames: (a) a list of 3.5 million usernames gathered from public Google profiles; (b) a list of 6.5

million usernames gathered from eBay accounts; (c) a list of 16000 usernames gathered from our research center LDAP directory; (d) two large username lists found online used in a previous study from Dell'Amico et al. [6]: a "finnish" dataset and a list of usernames collected from Myspace.

The "finnish" dataset comes from a list publicly disclosed in October 2007[1]. The dataset contains usernames, email addresses and passwords of almost 79000 user accounts. This information has been collected from — most likely by hacking — the servers of several Finnish web forums. The MySpace dataset comes from a phishing attack, setting a fake MySpace login web page. This data has been disclosed in October 2006 and it contains more than 30000 unique usernames.

The use we made of these datasets was threefold. First, we used the combined list of 10 million usernames (from eBay and Google) to train our Markov Chain model needed for the probability estimations. Second, we used the information on Google profiles to gather ground truth evidence and test our technique to link multiple public profiles even in case of slightly different usernames (Section 4). Third, we used all the datasets to characterize username uniqueness and depict Surprisal information distributions as seen in the wild.

Notably, a feature of Google Profiles [2], allowed us to build a *ground truth* we used for validation purposes. In fact, users on Google Profiles can optionally decide to provide a list of their other accounts on different OSNs and web services. This provided us with a ground truth, for a subset of all profiles, of linked accounts and usernames. In our experiments we observed that web services differ significantly in their username policies. However, almost all services share a common alphabet of letters and numbers and the dot (`.`) character. We note that usernames in different alphabets would need a training dataset in the proper alphabet. However, most services enforce strict rules on the username that can only be Latin alphanumerical characters.

3 Estimating Username Uniqueness

As we explained above, we would like to have a measure of username *uniqueness*, which can quantify the amount of identifying information each username carries. Information Surprisal is a measure, expressed in bits, that serves this purpose. However, in order to compute the Information Surprisal associated to usernames, we need a way to estimate the probability $P(u)$ for each username u.

A naive way to estimate $P(u)$, given a dataset of usernames coming from different services, would be to use Maximum Likelihood Estimation (MLE). If we have N usernames then we can estimate the probability of each username u as $\frac{count(u)}{N}$, if u belongs to our dataset, and 0 otherwise. Where $count(u)$ is simply the number of occurrences of u in the sample. In this case we are assigning maximum probability to the observed samples and zero to all the others. This approach has several drawbacks, but the most severe is that it cannot be used

[1] `http://www.f-secure.com/weblog/archives/00001293.html`
[2] `http://www.google.com/profiles`

to give any estimation for the usernames not in the sample. Furthermore, the estimation given is very coarse.

Instead, we would like to have a probability estimation that allows us to give estimate probabilities for usernames we have never encountered. Markov-Chains have been successfully used to extrapolate knowledge of human language from small corpuses of text. In our case, we apply Markov Chain techniques on usernames to estimate their probability.

3.1 Estimating username probabilities with Markov Chains

Markov models are successfully used in many machine learning techniques that need to predict human generated sequences of words, as in speech recognition [12]. In a very common machine learning problem, one is faced with the challenge of predicting the next word in a sentence. If for example the sentence is *"The quick brown fox"*, the word *jumps* would be a more likely candidate than *car*. This problem is usually referred to as *Shannon Game* following Shannon's seminal work on the topic[15]. This task is usually tackled using Markov-Chains and modeling the probability of the word *jumps* depending of a number of words preceding it.

In our scenario, the same technique can be used to estimate the probability of username strings instead of sentences. For example, if one is given the beginning of a username like `sara`, it is possible to predict that the next character in the username will likely be `h`. Notably Markov-Chain techniques have been successfully used to build password crackers [13] and analyse the strength of passwords [6].

Mathematical treatment with Markov Chains. Without loss of generality, the probability of a given string $c_1, ..., c_n$ can be written as $\Pi_{i=1}^{n} P(c_i|c_1, ..., c_{i-1})$, where the probability of each character is computed based on the occurrence all the characters preceding it. In order to make calculation possible a Markovian assumption is introduced: to compute the probability of the next character, only the previous k characters are considered. This assumption is important because it simplifies the problem of learning the model from a dataset. The probability of any given username can be expressed as:

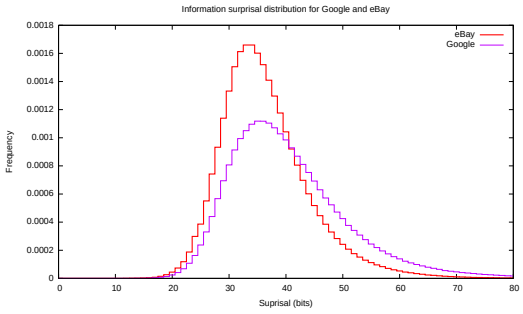
$$P(c_1, ..., c_n) = \Pi_{i=1}^{n} P(c_i|c_{i-k+1}, ..., c_{i-1})$$

To utilize Markov-Chain for our task we need to estimate, in a learning phase, the model parameters (the conditional probabilities) using a suitable dataset. In our experiments we used the database of approximately 10 million usernames populated by collecting Google public profiles and eBay user accounts (see Section 2.3). In general, the conditional probabilities are computed as:

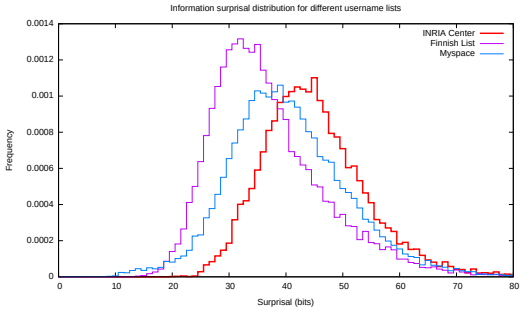
$$P(c_i|c_{i-k+1}, ..., c_{i-1}) = \frac{count(c_{i-k+1}, ..., c_{i-1}, c_i)}{count(c_{i-k+1}, ..., c_{i-1})}$$

by counting the number of n-grams that contain character c_i and dividing it by the total number of $n-1$-grams without the character c_i. Where an n-gram is simply a sequence of n characters.

Markov-Chain techniques benefit from the use of longer n-grams, because longer "histories" can be captured. However longer n-grams result into an exponential decrease of the number of samples for each n-gram. In our experiments we used 5-grams for the computation of conditional probabilities. Once we have calculated $P(u)$, we can trivially compute the information surprisal of u as $-\log_2(P(u))$. In Appendix 6 we give a different, yet related, probabilistic explanation of username uniqueness.



(a) Surprisal distribution for eBay and Google username



(b) Surprisal distribution for other services

Fig. 1. Information surprisal distribution for all the datasets used.

3.2 Experiments

We conducted experiments to estimate the surprisal of the usernames in our dataset and hence how unique and identifying they are. As explained above, our Markov-Chain model was trained using the combined 10 million usernames gathered from eBay and Google. The dataset was used for both training and

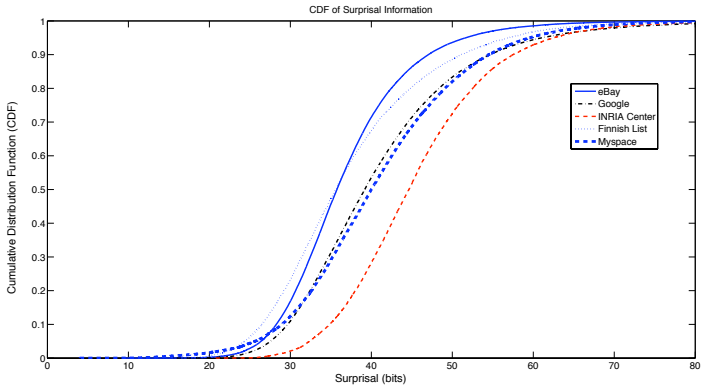


Fig. 2. Cumulative distribution function for the surprisal of all the services

testing by using leave-one-out cross validation. Essentially, when computing the probability of a username u using our Markov-Chain tool, we *excluded* u from the model's occurrence counts. This way, the probability estimation for u depended on all the other usernames but u.

We computed information surprisal for all the usernames in our dataset and the results are shown in Figure 1(a). The entropy of both distributions is higher than 35 bits which would suggest that, on average, usernames are extremely unique identifiers.

Notice the overlap in the distributions that might indicate that our surprisal measure is stable across different services. Notably, the two services have largely different username creation policies, with eBay accepting usernames as short as 3 characters from a wider alphabet and Google giving more restrictions to the users. Also, the account creation interfaces vary greatly across the two services. In fact, Google offers a feature that suggests usernames to new users derived from first and last names. Probably this is the reason why Google usernames have a higher Information Surprisal (see Figure 2). It must also be noted that both services have hundreds of millions of reported users. This raises the entropy of both distributions: as the number of users increases they are forced to choose usernames with higher entropies to find *available* ones. Overall it appears clear that usernames constitute highly identifying piece of information, that can be used to track users across websites.

In Figure 1(b) we plot information surprisal for three datasets gathered from different services. This graph is motivated by our need to understand how much surprisal varies across services. The results are similar to the ones obtained for eBay and Google usernames. The Finnish list is noteworthy, these usernames come from different Finnish forums and most likely belong to Finnish users. However, Suomi (the official language in Finland) shares almost no common roots with Roman or Anglo-Saxon languages. This can be seen as a good representative of the stability of our estimation for different languages.

Furthermore, notice that the dataset coming from our own research center (INRIA) has a higher surprisal than all the other datasets. While there are a possible number of explanations for this, the most likely one comes from the username creation policies in place that require usernames to be the concatenation of first and last name. The high surprisal comes despite the fact that the center has only around 16000 registered usernames and lack of availability does not pressure users to choose more unique usernames.

Comparing the distributions of Information surprisal of our different datasets is enlightening, as illustrated in Figure 2. This confirms that usernames collected from the INRIA center exhibit the highest information surprisal, with almost 75% of usernames with a surprisal higher than 40 bits. We also observe that both Google and MySpace CDF curves closely match. In all cases, it is worth noticing that the maximum (resp. the minimum) fraction of usernames that do exhibit an information surprisal less than 30 bits is 25% (resp. less than 5%). This shows that a vast majority of users from our datasets can be uniquely identified among a population of 1 billion users, relying only on their usernames.

4 Linking Different Username Strings

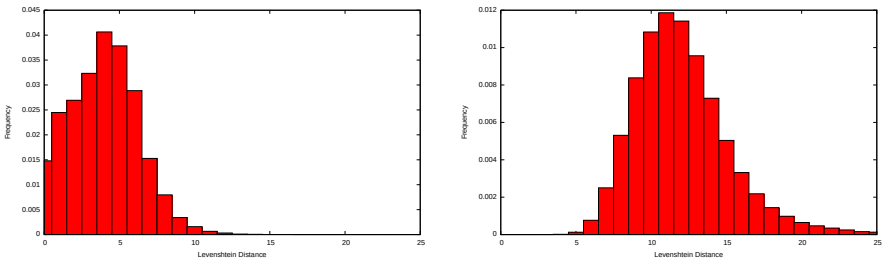
The technique explained above can only estimate the uniqueness of a single username across multiple web services. However, there are cases in which users, either willingly or forced by availability, decide to change their username. For example, the username `dan.perito` and `daniele.perito` likely belong to the same individual. Before embarking in this study, we would like to know whether users change their usernames in any predictable and traceable way. For this purpose, we use a subset of Google Profiles, in which the users explicitly gave information about the usernames of linked accounts.

In Figure 3(a) and 3(b) is plotted the distribution of the Levenshtein (or Edit) Distance for *linked* username couples. In particular, Figure 3(a) depicts the distribution for 10^4 username couples we can verify to belong to single users (we call this set L for *linked*), using our dataset. On the other hand, Figure 3(b) shows the distribution for a sample of random username couples that do not belong to a single user (we call this set NL for *non-linked*). In the first case the mean distance is 4.2 and the standard deviation is 2.2, in the second case the mean Levenshtein distance is 12 and the standard deviation is 3.1.

Clearly, linked usernames are much closer to each other than non linked ones. This suggests that, in many occurrences, users choose usernames that are related. The difference in the two distributions is remarkable and so it might be possible to estimate the probability that two different usernames are used by the same person or, in record linkage terminology, to *link* different usernames.

However, as illustrated in Section 3, and differently from record linkage, an almost perfect username match does not always indicate that the two usernames belong to the same person. The probability that two usernames, e.g. `sarah` and `sarah2`, are linked (we call it $P_{same}(sarah, sarah2)$) should depend on: (1) how "unique" is in the common part of the usernames (in this case `sarah`); and (2)

how likely is that a user will change one username into the other (in this case the addition of a 2 at the end).



(a) Levenshtein distance distribution for linked username couples (set L, $|L| = 10^4$)

(b) Levenshtein distance distribution for non-linked username couples (set NL, $|NL| = 10^4$)

Fig. 3. Levenshtein distance distribution for username couples gathered from 3.5 million Google profiles. Only couples that different at least in 1 character were considered.

Our goal is to establish a similarity measure that lies in $[0, 1]$ between two username strings u_1 and u_2, that can then be used to build a classifier to decide whether u_1 and u_2 are linked or not. We will show two different novel approaches at solving this problem. The first approach uses a combination of Markov Chains and a weighted Levenshtein Distance using probabilities. The second approach makes use of the theory and techniques used for information retrieval in order to compute document similarity, specifically using TF-IDF.

We compare these two techniques to well-known record linkage techniques for a base-line comparison. Specifically we use string-only metrics like the Normalized Levenshtein Distance (NLD) and Jaro distance to link username couples. However, because of lack of space, we will not explain them in detail.

Method 1: Linkage using Markov-Chains First of all, we need to compute the probability of a certain username u_1 being changed into u_2. We denote this probability as $P(u_2|u_1)$. Going back to our original example, $P(sarah2|sarah)$ is equal to the probability of adding the character 2 at the end of the string `sarah`. This same principle can be extended to deletion and substitution. In general, if two strings u_1 and u_2 differ by a sequence of basic operations $o_1, o_2, ..., o_n$, we can estimate $P(u_2|u_1) \equiv P(u_1|u_2) = p(o_1) \times p(o_2) \times ... \times p(o_n)$.

In order to estimate the probability that username u_1 and u_2 belong to the same person, we need to consider that there are two different possibilities on how u_1 and u_2 were chosen in the first place. The first possibility is that they were picked independently by two different users. The second possibility is that they were picked by the same user, hence they are not independent.

In the former case we can compute $P(u_1 \wedge u_2)$ as $P(u_1) \times P(u_2)$ since we can assume independence. In the latter, $P(u_1 \wedge u_2)$ equals $P(u_1) \times P(u_2|u_1)$ in case the user is the same. Note that using Markov Chains and the our estimation of $P(u_2|u_1)$, we can compute all the terms involved. Estimating the probability $P_{same}(u_1, u_2)$ is now a matter of estimating and comparing the two probabilities above.

The formula for $P_{same}(u_1, u_2)$ is derived from the probability $P(u_1 \wedge u_2)$ using Bayes' Theorem. In fact, we can rewrite the probability above as $P(u_1 \wedge u_2|S)$ where the random variable S can have values 0 or 1 and it is 1 if u_1 and u_2 belong to the same person and 0 otherwise. Hence without loss of generality:

$$P(S|u_1 \wedge u_2) = \frac{P(u_1 \wedge u_2|S)P(S)}{\sum_{S=0,1}(P(u_1 \wedge u_2|S) * P(S))}$$

which leads to $P(S = 1|u_1 \wedge u_2)$ equal to

$$\frac{P(u_1)P(u_2|u_1)P(S = 1)}{P(u_1)P(u_2)P(S = 0) + P(u_1)P(u_2|u_1)P(S = 1)}$$

where $P(S = 1)$ is the probability of two usernames belonging to the same person, regardless of the usernames. We can estimate this probability to be $\frac{1}{W}$, where W is the population size. Conversely $P(S = 0) = \frac{W-1}{W}$. And so we can rewrite $P_{same}(u_1, u_2)$ as $P(S = 1|u_1 \wedge u_2)$ equal to

$$\frac{P(u_1)P(u_2|u_1)}{W * P(u_1)P(u_2)\frac{W-1}{W} + W * P(u_1)P(u_2|u_1)\frac{1}{W}}$$

Please note that when $u_1 = u_2 = u$ then the formula above becomes

$$P_{same}(u, u) = \frac{1}{(W - 1)P(u) + 1} = P_{uniq}(u)$$

which is the same estimation we devised for the username uniqueness in Appendix.

Method 2: Linkage using TF-IDF In this case we use a well known information retrieval tool called TF-IDF. However, TF-IDF similarity measures the distance between two documents (or a search query and a document), which are set of words. We need to slightly alter the TF-IDF measure to apply it to username strings instead.

The term frequency-inverse document frequency (TF-IDF) is a weight used to evaluate how important is a word to a document that belongs to a corpus [10]. The weight assigned to a word increases proportionally to the number of times the word appears in the corpus but the importance decreases for common words in the corpus.

If we have a collection of documents D in which each document $d \in D$ is a set of terms, then we can compute the *term frequency* of term $t_i \in d$ as: $tf_{i,j} = \frac{n_{i,j}}{\Sigma_k n_{k,j}}$ where $n_{i,j}$ is the number of times term t_i appears in document d_j. The *inverse document frequency* of a term t_i in a corpus D is $idf_i = \frac{|D|}{c_i}$ where c_i is the

number of documents in the corpus that contain the term t_i. The TF-IDF is computed as $(tf - idf)_{i,j} = tf_{i,j} idf_i$. The TF-IDF is often used to measure the similarity between two documents, say d and d', in the following way: first the TF-IDF is computed over all the term in d and d' and the results are stored in two vectors v and v'; then the similarity between the two vectors is computed, for example using a cosine similarity measure $sim(d, d') = \frac{v \cdot v'}{\|v\| \|v'\|}$.

In our case we need to measure the distance between usernames composed of a single string. The way we solved this problem is pragmatical: we consider all possible substrings, of size q, of a string u to be a document d_u. Where d_u can be seen as the *building blocks* of the string u. The similarity between username u_1 and u_2 is computed using the similarity measure described above. This similarity measure is referred to in the literature as q-gram similarity [16], however it has been proposed for fuzzy string matching in database applications and its application to online profiling is novel.

4.1 Validation

Our goal is to assess how accurately usernames can be used to link two different accounts. For this purpose we design and build a classifier to separate the two sets L and NL, respectively of *linked* usernames and *non-linked* usernames.

For our tests the ground-truth evidence was gathered from Google Profiles and the size the number of linked username couples $|L|$ is 10000. In order to fairly estimate the performance of the classifier in a real world scenario we also randomly paired 10000 non-linked usernames to generate the NL set. The username couples were separated, shuffled and a list of usernames derived from L and NL was constructed. The task of the classifier is to re-link the usernames in L maximizing the username couples correctly linked while linking as few incorrect couples as possible. In practise for each username in the list our program computed the distance to any other username and kept only the link to the *single* username with highest similarity. If this value is above a threshold then the candidate couple is considered *linked* otherwise *non-linked*.

Measuring the performance of our binary classifier Binary classifiers are primarily evaluated in terms of *Precision* and *Recall*, where precision is defined in terms of true positives (TP) and false positives (FP) as follows $precision = \frac{TP}{TP+FP}$ and recall takes into account the true positives compared to false negatives $recall = \frac{TP}{TP+FN}$. The recall is the proportion of usernames couples that where correctly classified as linked (TP) out of all linked usernames ($TP + FN$).

In our case, we are interested in finding usernames couples that are actually linked (true positives) while minimizing the number of couples that are linked by mistake (false positives). Precision for us is a measure of exactness or fidelity and higher precision means less profiles linked by mistake. Recall measures how complete our tool is, which is the ratio of linked profiles that are found out of all linked ones. Precision and recall are usually shown together in a precision/recall graph. The reason is that they are often closely related: a classifier with high

recall usually has sub-optimal precision while one with high precision has lower recall. An ideal classifier has both a high precision and recall of 1.

Our classifier looks for potentially matching usernames. Once a set of potential matches is identified our scoring algorithms are used to calculate how likely it is that the two usernames represent the same individual. By using our labeled test data, score thresholds can be selected that yield a desired trade-off between recall and precision. Figure 4 shows the precision and recall of the two methods discussed in this paper and known string metrics (Jaro and NLD) at various threshold levels.

In general the metric based on Markov models outperforms the other metrics. Our Markov-Chain method has the advantage of having the highest precision values especially at recalls up 0.71. Remember that a recall of 0.71 means that 71% of all matching username couples have been successfully linked. Depending on the application, one might favor TF-IDF based approach (method 2) which has good precision at higher recalls or the Markov chain approach (method 1) which has the highest precision up to recall 0.7.

Table 1 shows specific examples of the performance of our classifier [3]. The similarity estimation is derived computing P_{same} as described in this Section. Username couples like `johnsmith` and `johnsmith82`, even though very similar, are deemed too common and therefore are correctly not linked by our classifier. Higher "entropy" usernames couples like `daniele.perito` and `d.perito` are correctly classified as likely belonging to the same person, even though there are 6 letters that differ out of 14. Example number 6 shows two slightly different usernames that contain the name Mohamed in them. However, since Mohamed is a very common Arabic first name, the model successfully deems the usernames as common and therefore does not link them.

Example #	Username 1	Username 2	Similarity	I_1	I_2	Classifier Decision
1	`ladygaga`	`ladygaga87`	9.34×10^{-9}	24.37 bits	34.63 bits	Non-linked
2	`johnsmith`	`john.smith`	5.08×10^{-9}	21.87 bits	24.34 bits	Non-linked
3	`johnsmith`	`johnsmith82`	2.94×10^{-10}	21.87 bits	30.51 bits	Non-linked
4	`mohamed.ali`	`mohamed.ali.ka`	3.28×10^{-7}	28.28 bits	38.23 bits	Non-linked
5	`daniele.perito`	`claude.castelluccia`	1.75×10^{-10}	39.76 bits	52.76 bits	Non-linked
6	`ccastel`	`claude.castelluccia`	$1.10 \times 10{-}10$	24.76 bits	52.76 bits	Non-linked
7	`johnsmith8219`	`john.smith8219`	0.73	37.74 bits	40.21 bits	Linked
8	`daniele.perito`	`d.perito`	0.006	39.97 bits	31.79 bits	Linked
9	`c.castelluccia`	`claude.castelluccia`	0.046	45.64 bits	52.76 bits	Linked
10	`uniquextrxyqm`	`kswaquniquextrxyqm`	0.999	60.09 bits	88.65 bits	Linked
11	`uniquextrxyqm`	`kswaquni1q3u4extrxyqm`	0.996	60.09 bits	130.64 bits	Linked

Table 1. Classifier similarity threshold fixed at $5.6 \times 10{-}4$ to maximize accuracy in the training set. All the username couples with a similarity above this threshold are classified as *linked*, the ones below *non-linked*. I_1 and I_2 stand for information surprisal of username 1 and 2 respectively.

[3] These examples are a combination of a set of the authors' usernames and usernames chosen to exemplify common features of the ones in our dataset.

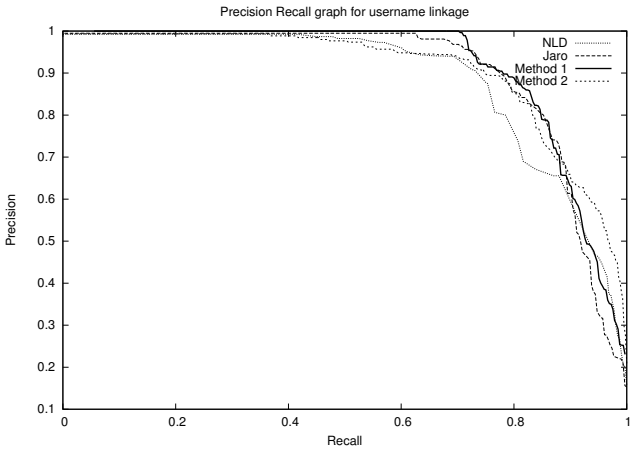


Fig. 4. Precision and recall for username Linkage

Discussion of Results Our results show that it is possible, with high precision, to link accounts solely based on usernames. This is due to the high average entropy of usernames and the fact that users tend to choose usernames that are related to each other. Clearly users could completely change their username for each service they use and, in this case, our technique would be rendered useless. However, our analysis shows that users indeed choose similar and high entropy usernames.

This technique might be used by profilers and advertiser trying to link multiple online identities together in order to sharpen their knowledge of the users. By crawling multiple web services and OSNs (a crawl of 100M Facebook profiles has already been made available on BitTorrent) profilers could obtain lists of accounts with their associated usernames. These usernames could be then used to link the accounts using the techniques underlined in the previous section.

Addressing Possible Limitations The linked username couples we used as ground truth have been gathered from Google Profiles. We have shown how that, in this sample, the users tend to choose related usernames. However, one might argue that this sample might not be sufficiently representative of the whole population. Indeed Google users might be least concerned about privacy and show a preference of being traceable by posting their information on their Profiles.

We were not able to test our tool in linking profiles of certain types of web services in which users are more privacy aware, like dating and medical websites (e.g., WebMD). This was due to the difficulty of gathering ground truth evidence for this class of services. However, even if we assume that users choose completely unrelated usernames for different websites, our tool might still be used. In fact, it might be the case that a user is registered on multiple dating websites with similar usernames. Those profiles might be linked together with our tool and more complete information about the user might be found. For example, a date

of birth on a website might be linked with a city of residence and a first name on another, leading to real world identification. A more thorough analysis is left for future work.

Finding linked usernames in a population requires time that is quadratic in the population size, as all possible couples must be tested for similarity. This might be too costly if one has millions of usernames to match. A solution to this problem is to divide the matching task in two phases. First, divide usernames in clusters that are likely be linked. For example, one could choose usernames that share at least one n-gram, thus restricting the number of combinations that need to be tried. Second, test all possible combinations within a cluster.

5 Discussion

This work shows that it is clearly possible to tie digital identities together and, most likely, to real identities in many cases only using ubiquitous usernames. We also showed that, even though users are free to change their usernames at will, they do not do it frequently and, when they do, it is in a predictable way. Our technique might then be used as an additional tool when investigating online crime. It is however also subject to abuse and could result in breaches in privacy. Advertisers could automatically build online profiles of users with high accuracy and minimum effort, without the consent of the users involved.

Spammers could gather information across the web to send extremely targeted spam, which we dub *E-mail spam 2.0*. For example, by matching a Google profile and an eBay account one could send spam emails that mention a recent sale or, by linking with Twitter, recent posts.

Countermeasures for Users Following this work users might change their username habits and use different usernames on different web services. We released our tool as a web application that users can access to estimate how unique their username is and thus take informed decision on the need to change their usernames when they deem appropriate (`http://planete.inrialpes.fr/projects/how-unique-are-your-usernames`). After its launch and following media coverage [4], our tool has already been used by more than 10000 users.

Countermeasures for Web Services There are two main features that make our technique possible and exploitable in real case scenarios. First, web services and OSNs allow access to public accounts of their users via their usernames. This can be used to easily check for existence of a given username and to automatically gather information. Some web services like Twitter are built around this particular feature. Second, web services usually allow the user pages to be crawled automatically. While in some cases this might be a necessary evil to allow search engines to access relevant content, in many instances there is no legitimate use of this technique and indeed some OSNs explicitly forbid it in the terms of service agreements, e.g., Facebook.

[4] E.g., MIT Technology Review: `http://www.technologyreview.com/web/32326/`

While preventing automatic abuse of public content can be difficult in general, for example when the attacker has access to a large number of IPs, it is possible to at least throttle access to those resources via CAPTCHAs [17] or similar techniques. For example, in our study we discovered that eBay presents users with a CAPTCHA if too many requests are directed to their servers from the same IP.

6 Conclusion

In this paper we introduced the problem of linking online profiles using only usernames. Our technique has the advantage of being almost always applicable since most web services do not keep usernames secret. Two family of techniques were introduced. The first one estimates the uniqueness of a username to link profiles that have the same username. We gather from language model theory and Markov-Chain techniques to estimate uniqueness. Usernames gathered from multiple services have been shown to have a high entropy and therefore might be easily traceable.

We extend this technique to cope with profiles that are linked but have different usernames and tie our problem to the well known problem of record linkage. All the methods we tried have high precision in linking username couples that belong to the same users. Ultimately we show a new class of profiling techniques that can be exploited to link together and abuse the public information stored on online social networks and web services in general.

Acknowledgments. We would like to thank Jean Baptiste Faddoul, Arvind Narayanan, Emiliano De Cristofaro and Abdelberi Chaabane for their invaluable suggestions and insights. We would also like to thank the reviewers for their insightful comments.

References

1. Scrapers dig deep for data on web. `http://online.wsj.com/article/SB10001424052748703358504575544381288117888.html`.
2. Balduzzi, M., Platzer, C., Holz, T., Kirda, E., Balzarotti, D., and Kruegel, C. Abusing social networks for automated user profiling. In *RAID'2010, 13th International Symposium on Recent Advances in Intrusion Detection, September 15-17, 2010, Ottawa, Canada* (09 2010).
3. Bilge, L., Strufe, T., Balzarotti, D., and Kirda, E. All your contacts are belong to us: Automated identity theft attacks on social networks. In *18th International World Wide Web Conference* (2009), pp. 551–560.
4. Cohen, W. W., Ravikumar, P., and Fienberg, S. E. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration* (August 2003), pp. 73–78.
5. Cover, T. M., and Thomas, J. A. *Elements of information theory.* Wiley-Interscience, New York, NY, USA, 1991.

6. Dell'Amico, M., Michiardi, P., and Roudier, Y. Measuring password strength: An empirical analysis.
7. Eckersley, P. How unique is your web browser? In *Privacy Enhancing Technologies, 10th International Symposium, PETS 2010, Berlin, Germany, July 21-23, 2010.* (2010), vol. 6205 of *Lecture Notes in Computer Science*, Springer.
8. Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering 19* (2007), 1–16.
9. Irani, D., Webb, S., Li, K., and Pu, C. Large online social footprints–an emerging threat. In *CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering* (Washington, DC, USA, 2009), IEEE Computer Society, pp. 271–276.
10. Jones, K. S. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation 28* (1972), 11–21.
11. JR., M. H., Baranov, P., McArdle, T., Boesenberg, T., and Duggal, B. Distributed personal information aggregator. Patent application number: 20100010993. http://www.faqs.org/patents/app/20100010993.
12. Manning, C. D., and Schuetze, H. *Foundations of Statistical Natural Language Processing*, 1 ed. The MIT Press, June 1999.
13. Narayanan, A. Fast dictionary attacks on passwords using time-space tradeoff. In *In ACM Conference on Computer and Communications Security* (2005), ACM Press, pp. 364–372.
14. Narayanan, A., and Shmatikov, V. De-anonymizing social networks. vol. 0, IEEE Computer Society, pp. 173–187.
15. Shannon, C. E. Prediction and entropy of printed english. *Bell Systems Technical Journal 30* (1951), 50–64.
16. Tata, S., and Patel, J. M. Estimating the selectivity of tf-idf based cosine similarity predicates. *SIGMOD Rec. 36*, 2 (2007), 7–12.
17. von Ahn, L., Blum, M., Hopper, N., and Langford, J. Captcha: Using hard ai problems for security. In *Advances in Cryptology - EUROCRYPT 2003*, E. Biham, Ed., vol. 2656 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2003, pp. 646–646.

Username uniqueness from a probabilistic point of view

We now focus on computing the probability that only *one* users has chosen username u in a population. We refer to this probability as $P_{uniq}(u)$.

Intuitively $P_{uniq}(u)$ should increase with the decrease in likelihood of $P(u)$. However, $P_{uniq}(u)$ also depends on the size of the population in which we are trying to estimate uniqueness. For example, consider the case of first names. Even an uncommon first name does not uniquely identify a person in a very large population, e.g. the US. However, it is very likely to uniquely identify a person in a smaller population, like a classroom.

To achieve this goal we use the $P(u)$ to calculate the expected number of users in the population that would likely choose username u. Let us denote by $n(u)$ the expected number of users that choose string u as a username in a given population W. The value of $n(u)$ is calculated based on $P(u)$ as:

$$n(u) = P(u) * W$$

where W is the total number of users in the population. In our case W is an estimation of the number of users on the Internet: 1.93 billions [5].

In case we are sure there exists *at least one* user that selected the username u (because u is taken on some web service) then the computation of $n(u)$ changes slightly:

$$n(u) = P(u) * (W - 1) + P(u|u) = P(u) * (W - 1) + 1$$

where the addition of 1 comes from the fact that we are sure that there exists *at least* one user that choses u and $W - 1$ is there to account for the person for which we are sure of.

Finally we can estimate the uniqueness of a username u by simply considering the probability that our user is unique in the reference set determined by $n(u)$, hence:

$$P_{uniq}(u) = \frac{1}{n(u)}$$

[5] http://www.internetworldstats.com/stats.htm