

# On the Reliability of Profile Matching Across Large Online Social Networks

Oana Goga  
MPI-SWS

Patrick Loiseau  
EURECOM

Robin Sommer  
ICSI

Renata Teixeira  
Inria

Krishna P. Gummadi  
MPI-SWS

Matching the profiles of a user across multiple online social networks brings opportunities for new services and applications as well as new insights on user online behavior, yet it raises serious privacy concerns. Prior literature has proposed methods to match profiles and showed that it is possible to do it accurately, but using evaluations that focused on sampled datasets only. In this paper, we study the extent to which we can *reliably* match profiles *in practice*, across real-world social networks, by exploiting *public attributes*, i.e., information users publicly provide about themselves. Today's social networks have hundreds of millions of users, which brings completely new challenges as a reliable matching scheme must identify the correct matching profile out of the millions of possible profiles. We first define a set of properties for profile attributes—Availability, Consistency, non-Impersonability, and Discriminability (ACID)—that are both necessary and sufficient to determine the reliability of a matching scheme. Using these properties, we propose a method to evaluate the accuracy of matching schemes in real practical cases. Our results show that the accuracy in practice is significantly lower than the one reported in prior literature. When considering entire social networks, there is a non-negligible number of profiles that belong to different users but have similar attributes, which leads to many false matches. Our paper sheds light on the limits of matching profiles in the real world and illustrates the correct methodology to evaluate matching schemes in realistic scenarios.

## 1. INTRODUCTION

Internet users are increasingly revealing information about different aspects of their personal life on different social networking sites. Consequently, there is a growing interest in the potential for aggregating user information across multiple sites, by matching user accounts across the sites, to develop a more complete profile of individual users than the profile provided by any single site. For instance, companies like PeekYou [26] and Spokeo [3] offer “*people search*” services that can be used to retrieve publicly visible information about specific users that is aggregated from across a multitude of websites. Some companies are mining data posted by job applicants on different social networking sites as part of background checks [31], while others allow call centers to pull up social profiles when their customers call [30]. The many applications of matching profiles across social networking sites also raise many legitimate and serious concerns about the privacy of users. A debate on the relative merits of leveraging profile matching techniques for specific applications is out of the scope of this paper.

In this paper, our goal is to investigate the *reliability* of techniques for matching profiles across *large real-world* online social networks, such as Facebook and Twitter, using only *publicly* available profile attributes, such as names, usernames, location, pho-

tos, and friends. Reliability refers to the extent to which different profiles belonging to the same user can be matched across social networks, while avoiding mistakenly matching profiles belonging to different users. Matching schemes need to be highly reliable because incorrectly matched profiles communicate an inaccurate portrait of a user and could have seriously negative consequences for the user in many application scenarios. For example, Spokeo has been recently sued over providing inaccurate information about a person which caused “actual harm” to the person employment prospects [4]. We focus on publicly available profile attributes because data aggregators today can crawl and leverage such information for matching profiles.

Recently, a number of schemes have been proposed for matching profiles across different social networks [22, 28, 34, 12, 21, 14] (we review them in §9.) The potential of these schemes to reliably match profiles in practice, however, has not been *systematically* studied. Specifically, it is not clear how or what properties of profile attributes affect the reliability of the matching schemes. Furthermore, the training and testing datasets for evaluating the matching schemes are often opportunistically generated and they constitute only a small subset of all user profiles in social networks. It is unclear whether the reliability results obtained over such datasets would hold over all user profiles in real-world social networks, where there are orders of magnitude more non-matching profiles than matching profiles (i.e., there is a huge class imbalance).

Our first contribution lies in defining a set of properties for profile attributes—Availability, Consistency, non-Impersonability, and Discriminability (ACID)—that are both necessary and sufficient to determine the reliability of a matching scheme (§3). Analyzing the ACID properties of profile attributes reveals the significant challenges associated with matching profiles reliably in practice (§4). First, data in real-world social networks is often *noisy* – users do not consistently provide the same information across different sites. Second, with hundreds of millions of profiles, there is a non-trivial chance that there exist multiple profiles with very similar attributes (e.g., same name, same location) leading to false matches. Finally, attackers create profiles attempting to impersonate other users, fundamentally limiting the reliability of any profile matching scheme.

Another key contribution lies in our method for carefully selecting the training and testing datasets for matching profiles (§5). When we evaluate the main types of matching schemes in the literature (based on binary classifiers) using a small random sample of Twitter and Facebook profiles (similar to how these schemes were evaluated originally), the schemes achieve over 90% recall and 95% precision (§6.1). Unfortunately, when we evaluate these schemes over datasets sampled carefully to preserve the reliability that the schemes would have achieved over the larger datasets (full Facebook database), their performance drops significantly (§6.2).

We could obtain only a 19% recall for a 95% precision.

We then investigate if we could improve the reliability of matching schemes in scenarios where we know that there is *at most one matching profile* (see §7). In such scenarios, we propose a new matching scheme and show that it is indeed possible to improve the recall to 29% at 95% precision. This is still considerably lower than the high recall (90%) reported in the literature.

Thus, we discover a fundamental limitation in matching profiles across existing social networks using public attributes. To further confirm the inherent limits of reliably matching profiles in practice, we compare the reliability of automated matching schemes with that of human Amazon Mechanical Turk (AMT) workers. Under similar conditions, AMT workers are able to match only 40% of the profiles with a 95% precision. Our analysis is the first to highlight that achieving high reliability in matching profiles across large real-world social networks comes at a significant cost (in terms of reduced recall).

## 2. PROBLEM DEFINITION

In this section, we define the problem of matching profiles, we present the constraints we have to consider and discuss how we approach the problem.

**The profile matching problem:** We consider that two profiles in two social networks match if they belong to/are managed by the same user. The profile matching problem is: given a profile  $a^1$  in one large social network  $SN_1$ , find all its *matching profiles* in another large social network  $SN_2$ , if at least one exists. We will denote by  $a^2$  generic profiles in  $SN_2$  and by  $\hat{a}^2$  matching profiles of  $a^1$ . For conciseness, we will also write  $a^2$ -match- $a^1$  if  $a^2$  is a matching profile of  $a^1$  and  $a^2$ -non-match- $a^1$  otherwise.

Note that we address here the problem of matching *individual* profiles, which is different from the problem of matching two entire social networks or databases. The difference is that we do not assume that we have access to all the data in  $SN_1$  but only to one profile. For example, we cannot match profiles by exploiting patterns in the graph structure of  $SN_1$  and  $SN_2$ , and we cannot optimize the matching of a profile in  $SN_1$  based on the matchings of other profiles in  $SN_1$ . Thus, we cannot take advantage of some methods proposed for de-anonymizing social graphs [23, 15] and entity matching [9].

Our problem formulation is motivated by practical scenarios. There are many people search engines such as Spokeo that allow users to search for data about a particular person. These services gather data about a person by matching the profiles a person has on multiple social networks.

We are particularly interested in two instantiations of the problem that are motivated by practical scenarios: (1) the *generic case* – a profile can have multiple matching profiles in  $SN_2$ ; and (2) the *special case* – a profile can have *at most one* matching profile. This case is suited for matching social networks such as Facebook or LinkedIn that enforce users to have only one profile.

**Features:** In this paper, we investigate the extent to which we can match profiles by exploiting the *attributes* users publicly provide in their profiles such as their *real names*, *screen names* (aka. username – name that appears in the URL of the profile), *location*, *profile photos*, and *friends*. Using this information we can ideally match any person that maintains the same persona on different social networks. Also, we choose these attributes because they are essential to find people online and they are present and usually remain public across different social networks even if users make all their other content, such as their posts and photos, private. For profile  $a^1$  (resp.  $a^2$ ), we denote by  $v^1$  (resp.  $v^2$ ) the value of a considered attribute.

From attribute values, we define a *feature* as the similarity between the values of profiles in  $SN_1$  and  $SN_2$ :  $s(v^1, v^2)$ .

**Matching scheme as a binary classifier:** Most previous works solved the matching problem by building binary classifiers that, given two profiles  $a^1$  and  $a^2$ , determine whether  $a^1$  and  $a^2$  are matching or not [21, 29, 33, 27, 34, 22, 25, 20]. The binary classifier takes as input a feature vector  $f(a^1, a^2)$  that captures the similarity between each attribute of a pair of profiles ( $a^1, a^2$ ); and then outputs the probability  $p$  of  $a^1$  and  $a^2$  to match. By selecting a cut-off threshold for  $p$  the classifier returns 1 (i.e., matching profiles) if  $p$  is larger than the threshold; and 0 otherwise. We say that a matching scheme outputs a *true match* when the matched profiles belong to the same user and outputs a *false match* when the matched profiles belong to different users. The threshold’s choice constitutes the standard tradeoff between increasing the number of true match and decreasing the number of false matches.

This solution works well for the generic case of our matching problem. Given a profile  $a^1$ , we can use the binary classifier to check, for every pair of profiles ( $a^1, a^2$ ) such that  $a^2 \in SN_2$ , whether it is matching or not. We can then output any profile  $a^2$  that the binary classifier declares as matching. In this paper, we test such approach when we represent ( $a^1, a^2$ ) with five features, each corresponding to the similarity score between  $a^1$  and  $a^2$  for each of the five profile attributes: real name, screen name, location, photo, and friends.

For the special case of our matching problem, the previous approach is vulnerable to output many false matches. For this case, instead of independently judging whether *each* pair ( $a^1, a^2$ ) is a match or not, we can compare (for a given  $a^1$ ) the probabilities  $p$  for *all* pairs ( $a^1, a^2$ ) to judge which profile is most likely the matching profile of  $a^1$ . We discuss this case in more detail in §7.

**Reliability of a profile matching scheme:** In this paper our focus is on the reliability of matching schemes. A *reliable matching scheme* should ensure that the profile it finds indeed matches with high probability, i.e., the matching scheme does not have many false matches. If there is no clear matching profile in  $SN_2$  for  $a^1$ , then the scheme should return nothing.

Many previous studies used the true and the false positive rate to evaluate their matching schemes. The true positive rate is the percentage of matching profiles that are identified, while the false positive rate is the percentage of non-matching profiles that are false matches. The goal is to have a high true positive rate and a low false positive rate. These metrics are, however, a misleading indicator of the reliability of a matching scheme because they are not suited for scenarios with high class imbalance, i.e., the number of matching profiles is much lower than the number of non-matching profiles. For example, a matching scheme with a 90% true positive rate for a 1% false positive rate might seem reliable, however, if we use it in a scenario where we have 1,000 matching and 999,000 non-matching profiles, the matching scheme would output 900 true matches and 9,990 false matches, which is clearly unreliable. In real-world social networks, the class imbalance is even higher (e.g., for each matching profile we have over 1 billion non-matching profiles in Facebook) thus the scheme would output even more false matches.

This paper argues that better metrics to evaluate the reliability of a matching scheme are the precision and recall. The recall is the percentage of matching profiles that are identified, while the precision the percentage of all pairs returned by the matching scheme which are true matches. The goal is to have a high recall and a high precision. In the previous example, we would have 90% recall for a 8% precision, which shows the low reliability of the scheme (out of all matched profiles only 8% are true matches). Thus, the best

way to show the reliability of a matching scheme is to evaluate its precision and recall with realistic class imbalance. In the rest of the paper, by *reliable* we mean a precision higher than 95%.

### 3. THE ACID FRAMEWORK

The natural question that arises when investigating the reliability of matching schemes is: what does the reliability depends on? Undoubtedly, the reliability depends on the attributes we consider for matching and on their properties. Thus, given an attribute, what properties should the attribute have in order to enable a reliable profile matching? We propose a set of four properties to help capture the quality of different attributes to match profiles: *Availability*, *Consistency*, *non-Impersonability*, and *Discriminability* (ACID).

**Availability:** At first, to enable finding the matching profile, an attribute should have its value available in both social networks. If only 5% of users provided information about their “age” across two sites, then “age” has limited utility in matching profiles. To formalize this notion, we model the attribute values of  $a^1$  and each  $a^2 \in SN_2$  as random variables and we define the availability of an attribute as:

$$A = Pr(v^1 \text{ and } v^2 \text{ available} | a^2\text{-match-}a^1).$$

**Consistency:** It is crucial that the selected attribute is consistent across matching profiles, i.e., users provide the same or similar attribute values across the different profiles they manage. Formally, we define the consistency of an attribute as:

$$C = Pr(s(v^1, v^2) > th | a^2\text{-match-}a^1, v^1 \text{ and } v^2 \text{ available}),$$

where  $th$  is a threshold parameter.

**non-Impersonability:** If an attribute can be easily impersonated, i.e., faked, then attackers can compromise the reliability of the matching by creating fake profiles that appear to be matching with the victim’s profiles on other sites. Some public attributes like “name” and “profile photo” are easier to copy than others such as “friends”. To formalize this notion, we introduce the notation  $a^2\text{-impersonate-}a^1$  to denote that profile  $a^2$  has been created by an attacker impersonating profile  $a^1$ . We denote the probability that there exists at least one profile  $a^2$  impersonating  $a^1$  by  $p_I = Pr(a^1 \text{ is impersonated})$  and the probability that there is no profile impersonating  $a^1$  by  $p_{nI} = 1 - p_I$ . The difficulty to manipulate an attribute is characterized by its non-Impersonability defined as:

$$nI = Pr\left(\max_{a^2: a^2\text{-impersonate-}a^1} s(v^1, v^2) < th\right).$$

**Discriminability:** Even without impersonations, in order to enable finding the matching profile, an attribute needs to uniquely identify a profile in  $SN_2$ . A highly discriminating attribute would have a unique and different value for each profile, while a less discriminating attribute would have similar values for many profiles. For example, “name” is likely to be more discriminating than “gender”. Formally, we define the discriminability of an attribute as:

$$D = Pr\left(\max_{a^2: a^2\text{-non-match-}a^1} s(v^1, v^2) < th | a^1 \text{ not impersonated}\right).$$

In practice, it is impossible to estimate  $D$  unless we are able to identify impersonating profiles. Instead, we estimate:

$$\tilde{D} = Pr\left(\max_{a^2: a^2\text{-non-match-}a^1} s(v^1, v^2) < th\right).$$

$\tilde{D}$  represents the “effective discriminability” taking into account possible impersonations. Since impersonators create non-matching profiles as similar as possible to the original profile, it is reasonable

to assume that  $\tilde{D} \leq D$ . Moreover, by application of Bayes formula, we can show that  $D \leq \tilde{D}/p_{nI}$  so that, if  $p_I$  is not too large,  $\tilde{D}$  gives a good estimate of  $D$ . If we assume that the impersonating profiles are independent from the other non-matching profiles, we can also prove that  $\tilde{D} = D \cdot (p_{nI} + nI \cdot p_I)$ . This clearly shows that  $\tilde{D}$  is close to  $D$  if either the attribute is hard to impersonate ( $nI$  close to one) or the proportion of impersonator is small ( $p_I$  small).

The ACID properties are clear and intuitive properties that help understand the potential of an attribute to perform reliable matching, as the following theorem formalizes.<sup>1</sup>

**THEOREM 1.** *Consider a classifier based on a given attribute that classifies as matching profiles if  $s(v^1, v^2) > th$ . The performance of the classifier is characterized by the following results.*

(i) *We have*

$$recall = C \cdot A.$$

(ii) *Assume that, for each profile  $a^1 \in SN_1$ , there is at most one matching profile in  $SN_2$ . Then,*

$$precision \leq \frac{recall}{recall + 1 - \tilde{D}}.$$

(iii) *Assume that  $p_I > 0$ . Then,  $precision = recall = 1$  iff  $A = C = nI = D = 1$ .*

In Theorem 1, the threshold parameter  $th$  must be the same as the one in the definitions of  $C$ ,  $nI$  and  $D$ . Theorem 1-(i) shows that the classifier’s recall is simply the product of consistency and availability. Theorem 1-(ii) gives a simple upper bound of the precision as a function of the effective discriminability (which itself is a function of the discriminability and of the impersonability, see above). This upper bound gives a good order of magnitude for the precision; moreover, for high precision (which is what we aim), given the small number of false positives, the true precision should be close to the bound. Finally, Theorem 1-(iii) confirms that a high value of all four ACID properties is *necessary* and *sufficient* to obtain high precision and recall.

Properties  $A$ ,  $C$  and  $nI$  are independent of the network scale, however, the discriminability very largely depends on the network scale since having more non-matching pairs decreases the probability that none of them has a high similarity score. This implies that we must estimate the precision and the recall of a matching scheme using datasets that accurately capture the ACID properties of profile attributes of the entire social network. Otherwise, the precision and the recall will be incorrect.

In practice different attributes satisfy the properties to different extent and the challenge is to combine different attributes with imperfect properties to achieve a reliable matching. The next section analyzes the ground truth for several large social networks to understand the limits of matching profiles across different sites.

## 4. LIMITS OF MATCHING PROFILES

To understand the limits of matching profiles, we analyze the ACID properties of profile attributes (screen name, real name, location, profile photo, and friends) across six popular social networks (Facebook, Twitter, Google+, LinkedIn, Flickr, and MySpace). First we present our method to gather ground truth of matching profiles and we then analyze each property separately.

### 4.1 Ground truth of matching profiles

Gathering ground truth of matching profiles spanning multiple social networks is challenging and many previous works manually

<sup>1</sup>The proof can be found in the Appendix A.



Table 1: Number of ground truth matching profiles obtained with Friend Finder (DATASET FF) and Google+ (DATASET G+) for different combinations of social networks.

	DATASET FF	DATASET G+
TWITTER - FACEBOOK	4,182	76,332
LINKEDIN - FACEBOOK	2,561	20,145
TWITTER - FLICKR	18,953	35,208
LINKEDIN - TWITTER	2,515	20,439

Table 2: Availability of attributes for DATASET FF.

Legend: Tw = Twitter, Fb = Facebook, Fl = Flickr, Lnk = LinkedIn.

	Screen Name	Real Name	Profile Photo	Location	Friends
Tw	100%	100%	69%	54%	86%
Fb	100%	100%	98%	52%	60%
Fl	100%	30%	29%	11%	40%
Lnk	100%	100%	57%	99%	0%
Fb - Tw	100%	100%	69%	30%	43%
Fb - Lnk	100%	100%	56%	54%	0%
Tw - Fl	100%	30%	24%	8%	32%
Lnk - Tw	100%	100%	44%	54%	0%

labeled profiles [18, 33, 27]. Below we describe two automatic methods that we used to obtain our ground truth.

We first obtained ground truth data by exploiting “Friend Finder” mechanisms on many social networks that allow a user to find her friends by their emails. We used a list of email addresses collected by colleagues for an earlier study analyzing spam email [16].<sup>2</sup> These email addresses were collected on a machine instructed to send spam by a large bot network. Since spammers target the public at large we believe that this list of emails catch a representative set of users. To combat abuse, some social networks limit the number of queries one can make with their “Friend Finder” mechanism and employ techniques to make an automated matching of an email to a profile ID impossible. Hence, we were only able to collect the email-to-profile ID matching for Twitter, Facebook, LinkedIn and Flickr. Table 1 summarizes the number of matching profiles we obtained using the Friend Finder mechanism (DATASET FF).<sup>3</sup>

Some previous works obtained ground truth from users that willingly provide links to their profiles in different social networks. Such users might not represent users in general because they want their profiles to be linked and probably expend the effort to keep their profiles synced. To be able to compare our results against previous works we collected DATASET G+ (see Table 1) by exploiting the fact that Google+ allows users to explicitly list their profiles in other social networks on their profile pages. Due to space constraints, for the rest of the paper, we show by default the results for profiles in DATASET FF and occasionally, for comparison, we show the results are for DATASET G+.

## 4.2 Attribute availability

The availability of attributes depends on the social network, for example Twitter does not ask users about their age while Facebook does. The availability also depends on whether users choose to input the information and make it public. Users might choose to let their location public on Twitter while make it private on Facebook.

Table 2 shows the breakdown of attribute availability per social network and pairs of social networks. The availability per social network characterizes the behavior of users, while the availability for pairs of social networks corresponds to the definition of  $A$  in §3.

First, we find that the availability of the attributes varies considerably across the different social networks. For example, users are

more likely to provide their location information on LinkedIn than they are on Facebook or Twitter. The differences in availability are presumably due to the different ways in which users use these sites. For our purposes, it highlights the additional information one could learn about a user by linking her profiles on different sites.

Second, we find that screen name and real name are considerably more available than location or friends. However, the availability of the less available attributes is not negligible – for example, location and friends are available for more than 30% of matching profiles in Twitter and Facebook.

Third, when we compare the availability using DATASET FF and DATASET G+ (not shown), we observe that the availability of attributes for profiles in the DATASET FF is much lower than the availability for profiles in the DATASET G+ (e.g., profile photo is available for only 69% of Twitter users in DATASET FF while it is available for 96% of users DATASET G+).<sup>4</sup> Thus, users in DATASET G+ are more likely to complete their profiles and consequently there is a higher bound on the recall to match them.

## 4.3 Attribute consistency

We now study the extent to which users provide consistent attribute values for their profiles on different social networks. Some users deliberately provide different attribute values either out of concerns for privacy or out of a desire to assume online personas different from their offline persona. It would be very hard to match profiles of such users by exploiting their public attributes.

Other users may input slightly different values for an attribute across sites. For example, a user might specify her work place as International Business Machines on one site and International Business Machines Corporation on another site.

**Similarity metrics for profile attributes:** We borrow a set of standard metrics from prior work in security, information retrieval, and vision communities to compute similarity between the values of attributes: the Jaro distance [10] to measure the similarity between names and screen names; the geodesic distance to measure the similarity between locations; the phash [2] and SIFT [19] algorithms to detect whether two photos are the same; and the number of common friends between two profiles. Please check our Appendix B for a full description of these metrics.

**Similarity thresholds for attribute consistency:** Clearly the more similar two values of an attribute, the greater the chance that the values are consistent, i.e., they refer to the same entity, be it a name or photo or location. Here, we want to show consistency results for a “reasonable” threshold beyond which we can declare with high confidence that the attribute values are consistent (irrespective of the tradeoff between consistency and discriminability in §3). The best to judge whether two attribute values are consistent are humans. Thus, we gathered ground truth data by asking Amazon Mechanical Turk (AMT) users to evaluate whether pairs of attribute values are consistent or not. We randomly select 100 pairs each of matching and non-matching Twitter and Facebook profiles from DATASET FF and asked AMT users to annotate which attribute values are consistent and which are not. We followed the guidelines to ensure good quality results from AMT workers [7].

For each attribute, we leverage the AMT experiment to select the similarity thresholds to declare two values as consistent. Specifically, we select similarity thresholds, such that more than 90% of the consistent values, as identified by AMT workers, and less than 10% of the inconsistent values have high similarities. Note that, we only use these thresholds to evaluate whether attribute values in

<sup>2</sup>The local IRB approved the collection.

<sup>3</sup>To test the representativeness of DATASET FF, we compare the distribution of properties such as account creation date, number of followers, and number of tweets of Twitter profiles in our dataset with the same properties of random Twitter profiles. We found that the pairs of distributions for each property matched fairly closely.

<sup>4</sup>For more results on DATASET G+, we refer the reader to [11].

Table 3: Consistency of attributes for users in DATASET FF; † in parenthesis, the consistency only when information is available in both social networks.

	Screen Name	Real Name	Location	Profile Photo	Friends
Fb - Tw	38%	77%	23% (77%†)	8% (12%)	34% (79%)
Fb - Lnk	71%	97%	44% (83%)	11% (23%)	0%
Tw - Fl	40%	25% (84%)	5% (67%)	5% (22%)	13% (42%)
Tw - Lnk	36%	83%	39% (71%)	13% (31%)	0%

matching profiles are consistent and we do not use them to actually match profiles. Thus, while it is important that the majority of consistent values pass the threshold, it is not critical if some inconsistent values also pass the threshold. Incidentally, this experiment also shows that the similarity metrics we choose are consistent with what humans think it is similar. Note that, it is impractical to use AMT workers to estimate the threshold for friends, thus we manually choose it be at least two friends in common to avoid noise.

**Attribute consistency in matching profiles:** Table 3 shows the proportion of users who provide consistent values for an attribute in a pair of social networks out of all users. This proportion corresponds to the recall we can achieve using the attribute given the threshold used, as shown in the previous section. In parenthesis, we also provide the equivalent proportion of users with consistent values only when the attribute value is available in both social networks (corresponding to the definition of  $C$ ). This proportion better illustrates how likely users are to provide consistent values, i.e., shows the users’s attempt to maintain synched profiles.

First, we find that a large fraction of users provides similar *real names* across different social networks. Put differently, most users are not attempting to maintain distinct personas on different sites. This trend bodes well for our ability to match the profiles of a user.

Second, we computed the percentage of matching profiles in Twitter and Facebook for which all public attributes in Table 3 are inconsistent. We find that there are 7% of such users. These users are likely assuming different personas on different sites and it is very hard, if not impossible, to match their profiles using only the public attributes that we consider in this paper. Thus, we can at most hope to match profiles for 93% of users. This percentage represents an upper bound on the recall for matching profiles based on public attributes.

Third, the consistency differs between different social networks. Twitter and Facebook have one of the lowest consistency for each attribute while Facebook and LinkedIn have the highest consistency. Thus, users are more likely to maintain synched profiles across Facebook and LinkedIn than other pairs of social networks.

## 4.4 Attribute discriminability

The previous section showed that a large fraction of users have consistent attribute values between their profiles. However, the number of profiles that we can match reliably is smaller because attribute values might not uniquely identify a single person.

To evaluate the discriminability of attributes, for each Twitter profile we compare the similarity of the matching Facebook profile with the similarity of the most similar non-matching Facebook profile. Figure 1 shows the CDF of similarity scores in DATASET FF (sample). Zero means no similarity and one means perfect similarity; except for location, where zero means perfect similarity because it corresponds to the distance between locations. The vertical lines represent the similarity thresholds for consistent attribute values used in the previous subsection. Given a threshold, we have perfect discriminability if there are no non-matching profiles with higher similarities. Concretely, for a given similarity threshold ( $x$  value), the  $y$  value for the distribution for the most similar non-matching profile represents an estimate of the (effective) discrim-

inability  $\bar{D}$ , whereas the  $y$  value for the distribution of matching profiles represents the complementary of the recall  $1 - C \cdot A$ .

For the *real name* and *screen name* we see a clear distinction between distributions of matching and non-matching profiles in Figure 1. The highest similarity of non-matching profiles is around 0.75 while a number matching profiles have similarities around 1. This suggests that these attributes have a high discriminability. For *photo*, the two distributions are generally similar. The photo does not appear to have a very good overall discriminability because there are not many Facebook matching profiles that use the same profile photo with the Twitter profile. However, for similarities large than 0.10, when the profile photos are consistent, there are not many non-matching profiles. As expected, the *location* does not have a good discriminability; even in a small dataset there are Facebook non-matching profiles with the same location as the Twitter profile. Finally, *friends* have a good discriminability between matching and non-matching profiles, i.e., it is uncommon to have non-matching profiles with many common friends.

We do not have access to the whole Facebook dataset to evaluate the discriminability of all attributes over an entire social network, however, we exploit the Facebook Graph Search to estimate the discriminability of real names and screen names. For each Twitter profile we use Facebook Graph Search to retrieve all the profiles with the same or similar names and screen names. This procedure samples the non-matching profiles with the highest similarity; therefore it preserves the discriminability of the entire social network. Figure 1a and 1b also presents the discriminability of real names and screen names over the entire Facebook (entire). As expected, the CDF of similarity score for non-matching profiles is much lower than it was at small scale. Furthermore, for 60% of the Twitter profiles, there is a non-matching Facebook profile that has *exactly* the same real name and for 25% exactly the same screen name. Even worse, the CDFs of Figure 1a for non-matching profiles are even below the CDFs for matching profiles which means that in many cases there are non-matching profiles that have even more similar names with the Twitter profile than the matching profile. These results show that names and screen names are actually not so discriminating in practice and consequently shows the difficulty of reliably finding the matching profile in real-world social network. This also shows the risk of evaluating matching scheme over a sampled dataset because attributes have a much higher discriminability than over entire social networks.

## 4.5 Attribute impersonability

In most social networks a user is not required to prove that her online identity matches her offline person. Since there is a lot of personal data publicly available, it is very easy for attackers to create fake profiles that impersonate honest users. Because such attacks could be a very big source of unreliability for matching schemes, we show evidence that such attacks indeed exist and they are more frequent than previously assumed.

To search for potential cases of impersonation we start with an initial set of 1.4 million random profiles in Twitter. We find that, strikingly, a large fraction of profiles could be *potential* victims of impersonation attacks: 18,662 Twitter profiles have at least another Twitter profile with consistent profile attributes. This gives a rough estimate of  $p_I$  of 1%. It is beyond the scope of this paper to thoroughly investigate such attacks but in §7 we propose a way to make matching schemes less vulnerable to impersonation attacks.

## 5. TRAINING & TESTING MATCHING SCHEMES

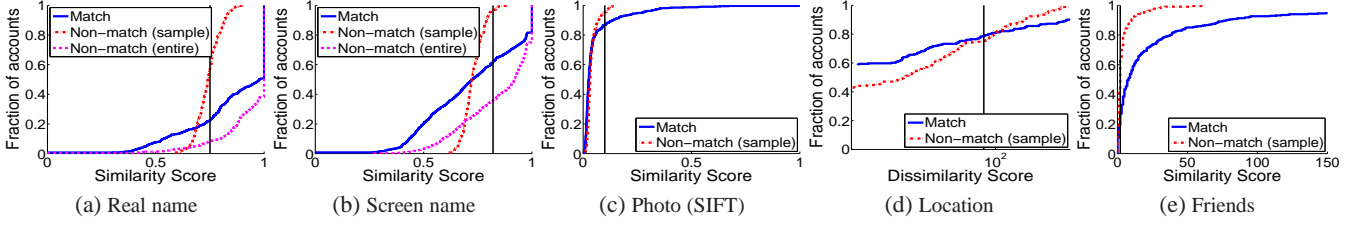


Figure 1: CDF of similarity scores for the matching Facebook profiles and for the most similar non-matching Facebook profile in DATASET FF (sample) and the entire Facebook (entire).

In this section, we focus our attention on the datasets used to train and evaluate (test) matching schemes. To estimate well the precision and recall in practice, we should test for each profile  $a^1$  the accuracy of finding the matching profiles  $\hat{a}^2$  out of all the profiles  $a^2 \in SN_2$ . If we consider large social networks like Facebook, Twitter, or Google+,  $SN_2$  has hundreds of millions of profiles. Obtaining such complete datasets is impractical, thus, we have to sample a number of profiles in the network.

Most previous studies sampled datasets by picking matching and non-matching profiles at random. Such random sampling fails to capture the precision and recall of matching schemes in practice because it severely over-estimates the discriminability of attributes found in the original social network (as seen in §4.4) and therefore it severely over-estimates precision. To estimate well the reliability of a matching scheme in practice, the sampled dataset needs to preserve the precision and recall of the original social network at least for high values of precision. The key to ensure this is to sample *all potential* false matches, i.e., all profiles that could be mistakenly matched by the matching scheme. Thus, we build two datasets: (1) a *reliability non-preserving sampled dataset* for comparison with previous techniques (as this is the standard evaluation method); and (2) a *reliability preserving sampled dataset* that strives to capture all possible false matches in a social network to better estimate the reliability of matching schemes in practice.

We generate a reliability preserving sampled dataset for matching Twitter and Facebook. Although building such dataset for other social networks is possible, the process is strenuous. Instead, we take two of the most popular social networks to show the limitations for matching profiles across real-world social networks.

**Reliability-non-preserving sampling:** We randomly sample 850 matching Twitter-Facebook profiles from DATASET FF and we use them to build 722,500 pairwise combinations of Twitter-Facebook profiles (850 positive and 721,650 negative examples). We call the resulting dataset the RANDOM-SAMPLED. The RANDOM-SAMPLED dataset preserves the availability and consistency of attributes in the original social network, but it does not preserve the discriminability and non-impersonability. Thus, the dataset does not preserve the precision of the original social network. Note that, datasets such as DATASET G+, which have been used in previous work, do not even preserve the availability and consistency of attributes because they are biased towards a particular kind of users (as seen in §4.2); hence they do not preserve recall.

**Reliability-preserving sampling:** To preserve the reliability over the original social network, our sampling strategy is to sample non-matching profiles that have a reasonably high similarity to  $a^1$  and ignore non-matching profiles that have a very small chance of matching. We note the set of most similar profiles to  $a^1$  in  $SN_2$  as  $C(a^1) \subset SN_2$ . A comprehensive  $C(a^1)$  includes all the Facebook profiles which could be potential false matches.

Given that our analysis in §4.3 shows that most Twitter-Facebook matching profiles have consistent real names or screen names, we

hope to build a comprehensive  $C(a^1)$  by exploiting the Facebook search API, which allows searching for people by name. For each Twitter profile,  $a^1$  (we sample the same Twitter profiles from RANDOM-SAMPLED), we generate  $C(a^1)$  using the Facebook search API to find profiles with the same or similar real name or screen name as  $a^1$ . The resulting dataset, which we call EMULATED-LARGE, contains over 270,000 combinations of profiles  $(a^1, a^2)$  where  $a^2 \in C(a^1)$ . Thus, for each Twitter profile the dataset contains in average 320 Facebook profiles with similar names.

Our analysis shows that the matching profile of  $a^1$  is in  $C(a^1)$  (i.e.,  $\hat{a}^2 \in C(a^1)$ ) for 70% of Twitter profiles. This implies that for 70% of cases we selected at least all non-matching profiles with higher name similarity than the matching profile. Additionally, the median similarity of the least similar real name in  $C(a^1)$  is 0.5, while the median similarity of matching profiles is 0.97. This means that we also catch many Facebook profiles with lower name similarity than the matching profiles. Thus, the only possible false matches that we miss are the ones that have very different names.

Note that the reliability preserving sampling does not sample the matching profile when there is little chance for it to match (in 30% of the cases). We actually tried to train and test matching schemes with or without including the unsampled matching profiles in the EMULATED-LARGE and the reliability did not differ significantly.

Our sampling strategy ensures that the discriminability and impersonability of real names and screen names found in the real-world datasets are preserved. It might over-estimate, however, the discriminability of location, friends, and profile photos since we do not sample in  $C(a^1)$  profiles with similar location, friends or photos if they do not also have similar names or screen names. Evaluating matching schemes over  $C(a^1)$  rather than all  $SN_2$  could lead to an under-estimation of the false matches. Thus, the precision we obtain over this dataset is an upper bound on the precision in practice. This implies that the limits of reliably matching schemes in practice can only be worse than what we show in this paper. We believe, however, that our sampling strategy gives a very good idea of the precision and recall in the real-world datasets because there will be very few false matches (if any) with very dissimilar names even if they have similar location, photo or friends.

Another limitation of the dataset is that it does not contain cases where a profile  $a^1$  has multiple matching profiles in  $SN_2$ . This is a consequence of our method to gather ground truth (§4.1) that only gives a single matching profile in  $SN_2$  for each  $a^1$ . The implications of this limitation on our evaluation is that there might be some matching profiles that we consider as false matches whereas they are not. Since Facebook enforces the policy that users should only have one profile, we believe there are not many such cases and the reliability we measure is likely close to the real-world reliability.

In practice, there are Twitter users that do not have a matching Facebook profile, but our datasets do not contain such cases. To evaluate how matching schemes perform in such scenarios, we test in §7 the reliability of matching schemes when we remove the matching profile from EMULATED-LARGE.



## 6. GENERIC MATCHING PROBLEM

This section evaluates the reliability of matching schemes based on classifiers aimed at solving the generic case of the matching problem (see §2). We build classifiers that are conceptually similar to what previous works have done. The primary difference between different previous matching schemes is the features and the datasets they used to train and test classifiers, however, they all use traditional classifier such as SVM and Naive Bayes. The goal of this section is not to build a matching scheme that is better than previous ones but to investigate the limits of such schemes in practice.

We first emulate the methodology employed by previous works: we train and test matching schemes with RANDOM-SAMPLED, using all attributes. Since some profile attributes have a high discriminability in the dataset, it is straightforward to build a matching scheme with high reliability. On top of this, there is little difference between the reliability of naive classification techniques and more sophisticated ones.

We then investigate the reliability of matching schemes in practice by testing them over EMULATED-LARGE. As expected, the precision of the previously built matching schemes drastically decreases to a point that makes them unusable. Thereafter, we investigate the reasons behind such poor reliability and we evaluate different strategies to increase the precision and recall in practice. The resulting schemes are able to achieve a good precision, but the recall is still low. These results show the inherent difficulty of matching profiles reliably in today’s large social networks.

### 6.1 Evaluation over RANDOM-SAMPLED

We use the RANDOM-SAMPLED dataset to train and test four classification techniques to match profiles: Naive Bayes, Decision Trees, Logistic Regression, and SVM. We split RANDOM-SAMPLED in two: 70% for training and 30% for testing.

There are two important aspects to handle when training classifiers to match profiles: (1) *classes are very imbalanced* – there are much more non-matching profiles than matching profiles. Previous works handled this problem by balancing the training instances by under-sampling the majority class [13]. We also adopt this technique and we randomly sample 850 non-matching profiles from the RANDOM-SAMPLED; (2) *features have missing values* – some attribute values may be unavailable hence the similarity value is missing (e.g., users may choose to omit their location or photo). Thus, we must either work with classification techniques that are robust to missing values (e.g., Naive Bayes) or identify methods to impute the missing values.

We use 10-fold cross validation on the training data to evaluate the four classifiers with different combinations of parameters and different methods for imputing the missing feature values. We call the four resulting classifiers with the best optimized parameters the LINKER-NB, LINKER-SVM, LINKER-LR and LINKER-DT.

We investigate the tradeoff between precision and recall for the different classifiers in Figure 2a. Our results show that LINKER-NB out of the box, without imputing the missing values and LINKER-SVM and LINKER-DT when we replace missing values with -1 achieve the highest reliability with a recall over 90% for a 95% precision. LINKER-LR achieves a lower recall, only 85% for the same precision. Thus, as expected, even out of the box classification techniques such as Naive Bayes are able to achieve a high precision and recall over RANDOM-SAMPLED.

**Analysis of matched pairs:** To understand what pairs of profiles the classifiers are matching, we analyze in Table 4 the availability and consistency of attributes for the *true matches*, the *false matches*, and the *missed matches* (the pairs of matching profiles

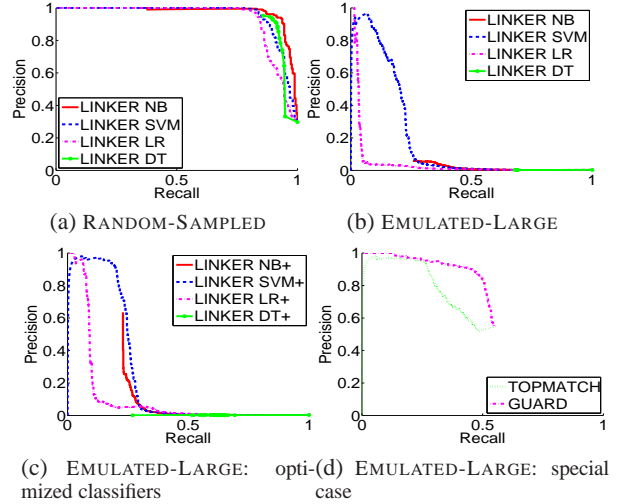


Figure 2: Precision and recall tradeoff for matching Twitter to Facebook profiles using different classifiers when evaluated over RANDOM-SAMPLED and EMULATED-LARGE.

Table 4: Fraction of true, missed and false matches that have available and consistent attributes in RANDOM-SAMPLED.

Feature	Fraction of available and consistent attributes			
	All Matches	True Matches	Missed Matches	False Matches
Real Name	0.77	0.91	0.20	0.62
Screen Name	0.38	0.46	0.07	0.09
Location	0.23	0.25	0.14	0.00
Profile Photo	0.08	0.10	0.01	0.00
Friends	0.34	0.34	0.22	0.38

that are *not* detected by the classifier). We use LINKER-SVM with a threshold on the probability  $p$  (outputted by the classifier) corresponding to a 95% precision (and 90% recall) to select the true, missed and false matches. The table shows that the only matching profiles the LINKER-SVM is not able to identify are the ones that do not have available and consistent attributes: only 20% of the missed matches have consistent names and 53% of missed matches do not have *any* consistent and available attribute (not shown in the table). The table also shows that the LINKER-SVM easily mistakes non-matching profiles for matching profiles if they have either consistent names or friends. While in this dataset this is not problematic, in practice this will lead to many false matches.

### 6.2 Evaluation over EMULATED-LARGE

Figure 2b presents the tradeoff between precision and recall when we evaluate using EMULATED-LARGE the four LINKER classifiers trained on RANDOM-SAMPLED. The figure shows that when matching profiles in practice the reliability of all four classifiers drops significantly compared to RANDOM-SAMPLED (presented in Figure 2a). The best classifier on the RANDOM-SAMPLED, LINKER-NB, achieve only a 4.5% precision for a 23% recall when tested on EMULATED-LARGE. The only classifier that achieves a satisfying 95% precision is LINKER-SVM, however, the recall is only 15%.

These results confirm our intuition that the reliability of a matching scheme over RANDOM-SAMPLED fails to capture the reliability of the matching scheme in practice. Worse, the matching scheme that has the best reliability when testing with RANDOM-SAMPLED (i.e., the LINKER-NB) can be amongst the worst in practice.

#### Optimizing the binary classifiers

**LINKER-NB:** We investigate the reasons for the low precision of LINKER-NB in EMULATED-LARGE. The results in Figure 1 show that matching profiles often have consistent names whereas non-

matching profiles (from sample) most often do not; there is no such clear distinction for the other attributes. Since Naive Bayes assumes that features are independent, the probability that two profiles match will be mainly determined by their name similarity. In a large social network, however, multiple users can have the same name, which will cause LINKER-NB to output many false matches.

One way to make the classification more accurate is to use two classifiers in cascade instead of one. The first classifier weeds out profiles that are clear non-matches (most of which have different names). Then, the second classifier takes the output of the first and disambiguates the matching profiles out of profiles with similar names. We call this improved classifier LINKER-NB+. For more details about this approach please refer to [11].

Another approach to make the classification more accurate is to use methods based on joint probabilities such as quadratic discriminant analysis. We prefer to move to SVM which also considers features jointly and is not restricted to quadratic boundaries.

**LINKER-SVM:** LINKER-SVM has a much higher precision in EMULATED-LARGE than LINKER-NB. Intuitively, this is because, as opposed to Naive Bayes, SVM considers the features jointly and hence can distinguish between pairs of profiles with high name similarity that match and pairs of profiles with high name similarity that do not match based on other features. Nevertheless, previous work has shown that SVM performs suboptimally when using under-sampling to deal with imbalanced datasets [6]. By under-sampling the majority class, we are missing informative data points close to the decision boundary.

To improve the reliability of LINKER-SVM, we take advantage of the fact that EMULATED-LARGE contains negative examples close to the decision boundary, to enrich our training set. We build a training set that contains 850 positive examples, 850 negative examples from RANDOM-SAMPLED plus 850 negative examples from EMULATED-LARGE. We call the resulting classifier the LINKER-SVM+. Note that if we only use for training negative examples from EMULATED-LARGE and not from RANDOM-SAMPLED, the resulting classifier will only be able to distinguish the matching profiles out of profiles that look similar and will not be able to distinguish the matching profile out of profiles that are clearly not similar, i.e., it will only work on datasets such as EMULATED-LARGE and not in practice. For LINKER-LR and LINKER-DT we apply the same retraining technique.

**Evaluation of optimized classifiers:** Figure 2c shows the tradeoff between precision and recall when using LINKER-SVM+, LINKER-NB+, LINKER-LR+, LINKER-DT+ on EMULATED-LARGE. We can see that LINKER-SVM+ is able to achieve a 19% recall (4% improvement over the LINKER-SVM) for a 95% precision.<sup>5</sup> Also, LINKER-NB+ achieves a 23% recall for a 88% precision, considerably better than LINKER-NB. Nevertheless, the recall is significantly lower compared with the recall obtained when testing with RANDOM-SAMPLED. Thus, even more sophisticated techniques trained to match profiles in real-word settings fail to match a large fraction of profiles.

**Analysis of matched pairs:** To understand the low recall we obtain in EMULATED-LARGE, we analyze again the availability and consistency of attributes. The precision of LINKER-SVM+ has a sudden drop, to go from a recall of 19% to 33%, the precision goes from 95% to 0.02%. To analyze the drop, we split the pairs of profiles in EMULATED-LARGE in true, missed, and false matches using first a threshold corresponding to a 95% precision (and a 19% recall) and then with a threshold corresponding to a 0.02% precision

Table 5: Fraction of true, missed and false matches that have available and consistent attributes in EMULATED-LARGE.

Feature	Fraction of available and consistent attributes					
	95% precision and 19% recall			0.02% precision and 33% recall		
	True Matches	Missed Matches	False Matches	True Matches	Missed Matches	False Matches
Real Name	0.94	0.73	0.86	0.94	0.69	1.00
Screen Name	0.60	0.33	0.57	0.64	0.26	0.89
Location	0.32	0.21	0.14	0.41	0.15	0.01
Profile Photo	0.14	0.07	0.57	0.16	0.04	0.07
Friends	0.91	0.17	0.57	0.68	0.14	0.00

sion (and a 33% recall), see Table 5.

Contrarily to our expectation, for most attributes but friends, the availability and consistency of true matches at 0.02% precision is actually slightly higher than the one at 95% precision. Only the availability and consistency of friends decreases from 91% at 95% precision to 68% at 0.02% precision. This means that, to go from 19% to 33% recall we mainly started to match profiles that do not have friends in common. The consequence is that while at 95% precision, the false matches needed to have friends in common, at 0.02% precision, false matches no longer need to have friends in common. This makes the matching scheme have orders of magnitude more false matches at 33% recall than at 19% recall. Thus, even if the features are highly available and consistent, if they are not discriminable enough, they will allow for many false matches which limits the precision and recall we can achieve in practice.

The results suggest that when matching profiles in practice, to maintain a high precision, we need features that are highly discriminable. Indeed, if we exclude friends (one of the most discriminable attributes) from the features we use for the classification, we can only achieve a 11% recall for a 90% precision.

## 7. SPECIAL MATCHING PROBLEM

The previous section showed that even fine tuned classifiers are vulnerable to output many false matches in practice. Worse, previous matching schemes are not able to protect against impersonation attacks. In this section, we propose ways to mitigate both of these problems in the special case where we know that there exists *at most one matching profile* in  $SN_2$ .

**The TOPMATCH:** The straw man approach is, for each profile  $a^1$ , to simply return the profile in  $C(a^1)$  with the highest probability  $p$  to be the matching profile given by LINKER-SVM+, provided that  $p$  is larger than a threshold. We call the most similar profile the TOPMATCH. This approach reduces the number of false matches since the matching scheme outputs at most one false match. Figure 2d displays the tradeoff between precision and recall obtained for different probability thresholds on the  $p$  of the TOPMATCH. It shows that TOPMATCH largely improves recall for a given precision: TOPMATCH in EMULATED-LARGE achieves to a 26% recall for a 95% precision.

**The GUARD:** The strategy of outputting the TOPMATCH considerably increases the recall compared to approaches in §6. However, it is still vulnerable to output false matches in practice when Twitter users who do not have a Facebook profile. Worse, the TOPMATCH is vulnerable to impersonation attacks that also hinder the reliability of the matching scheme. We propose next a simple solution that mitigates both of these problems by comparing the probability to be the matching profile of the most similar profile in  $C(a^1)$ ,  $p_{1st}$ , and the probability of the second most similar profile,  $p_{2nd}$ . The high level idea is that, to be sure that the most similar profile in  $C(a^1)$  is the matching profile,  $p_{1st}$  should be much higher than the probability  $p$  of any profile in  $C(a^1)$ , i.e.,  $p_{1st} \gg p_{2nd}$ .

Intuitively, there are two possible scenarios where the TOPMATCH

<sup>5</sup>In DATASET G+, LINKER-SVM+ has 50% recall and 95% precision.



is a false match: The first is if an attacker creates an impersonating profile on  $SN_2$  that is more similar than the true matching profile. It might be possible to detect these cases as both  $p_{1st}$  and  $p_{2nd}$  will be high and  $(p_{1st} - p_{2nd})$  will be very small. The second is when the true matching profile  $\hat{a}^2$  is in  $C(a^1)$  but a non-matching profile  $a^2 \in C(a^1)$  is chosen as output because the classifier assigns it a higher probability  $p$  of being the matching profile (due to the lack of attribute availability and/or consistency). Another case is when  $\hat{a}^2$  does not exist, forcing the scheme to choose the non-matching profile that is the most similar to  $a^1$  as the output. We might detect these cases as  $p_{1st}$  and  $p_{2nd}$  will not be very high (none of the profiles in  $C(a^1)$  are very similar to  $a^1$ ) and  $(p_{1st} - p_{2nd})$  will be again very small (none of the profiles in  $C(a^1)$  is much more similar than the rest).

To incorporate the above logic, we design the GUARD which is a binary classifier that takes as input  $p_{1st}$  and  $p_{2nd}$  and outputs the probability that the TOPMATCH is the matching profile. Figure 2d shows that the GUARD increases the recall of the matching scheme to 29% for a 95% precision. Although 29% recall is a big improvement over the recall previously obtained, the recall is still low. This shows that in practice, it is hard to achieve a high recall if we want to have a high precision.

The matching schemes in §6 decide independently for each pair  $(a^1, a^2)$  where  $a^2 \in C(a^1)$  whether it is a match or not. In contrast, the strength of the GUARD is that it exploits the structure of  $C(a^1)$  for a given  $a^1$ . In particular, since  $C(a^1)$  depends on  $a^1$ , for a given probability  $p$  to be the matching profile of  $a^1$ , the TOPMATCH profile  $a^2$  will be declared a match for some  $a^1$  if its attribute values are sufficiently unique, whereas the scheme will return nothing for other  $a^1$  if the attribute values are too common (e.g., Jennifer Clark that lives in New York). This reduces considerably the false matches and, as we have shown, increases a lot the matching recall for a given precision.

**Reliability in the absence of a matching profile:** To test the reliability of the matching scheme in the absence of a matching profile, we take the EMULATED-LARGE and we remove the matching profiles from the dataset. Then, we evaluate the GUARD over the resulting dataset. Ideally, the GUARD should not return any profile as there is no matching profile in the dataset. Indeed, the GUARD only returns a false match for 1% of the Twitter profiles. We manually investigate the 1% cases: in half the returned profile is a false match; in the other half it is actually a profile that corresponds to the same person (the returned profiles are either impersonators or people that maintain duplicate profiles on Facebook). Thus, the GUARD is reliable when there is no matching profile in  $SN_2$ .

## 8. EVALUATION AGAINST HUMANS

In this section, we confirm the inherent difficulty to obtain a high recall in matching profiles in practice by comparing our results with results obtained by asking human workers to match profiles.

For this we designed an AMT experiment. We randomly select 200 Twitter-Facebook matching profiles from DATASET FF (that are not used for training the matching schemes). In each assignment, we give AMT workers a link to a Twitter profile as well as links to the 10 most similar Facebook profiles (we shuffle their position) and we ask AMT workers to choose the matching profile. We allow workers to choose that they are unable to identify the matching profile. For each assignment we ask the opinion of three different workers. We present the results for majority agreement (two out of three workers decided on the same answer). We design two versions of the experiment: in the first one if the matching profile is not in  $C(a^1)$ , the matching profile will not be in the list of 10 Facebook profiles; and a second version, where we always put the

matching profile the list of 10 Facebook profiles.

In the first version of the experiment, AMT workers were able to match 40% of the Twitter profiles to their matching profiles and 4% are matched to the wrong Facebook profile. This means that AMT workers achieve a 40% recall for a 96% precision, which is better than the GUARD, but far from a 100% recall. In the second version of the experiment, AMT workers were able to match 58% of Twitter profiles. Thus, even humans cannot achieve a recall close to 100% to match profiles in practice.

## 9. RELATED WORKS

We review three primary lines of related research: one proposing schemes to match user profiles across different social networks; one focusing on how anonymized user graphs or databases can be deanonymized to infer user identities; and another about matching entities across databases.

**Matching profiles using private user data:** Balduzzi et al. [8] match profiles on different social networks using the “Friend Finder” mechanism that social networks provide for users to find their friends using their email addresses. In fact, this is what we use for obtaining our ground truth. Many sites, however, view Friend Finder as leaking users’ private data and have since limited the number of queries a user can make which severely limits the number of profiles one can match. In contrast, we are interested in understanding the limits of matching profiles by only using public attributes that anyone can access without assuming that we have access to more private data such as the emails of users.

**Matching profiles using public user data:** A number of previous studies proposed matching schemes that leveraged different attributes of public user data to match profiles, but without systematically understanding their limitations in real-world social networks. As a result, previous works overlooked a number of methodological aspects: (1) Most works did not train and test their matching schemes on sampled datasets that preserve the reliability of the original social network. Consequently, the reliability of these schemes drops significantly when evaluated in real-world social networks [28, 22, 29, 33, 27, 18, 36, 35]; (2) Most works used attributes without analyzing their properties and their limits to match profiles in practice, consequently, some of these studies use attributes with low availability and thus can only match a small fraction of profiles across a limited number of social networks [12, 14] or use attributes that are prone to give many false matches in practice [21]. On the contrary, we propose a framework to analyze attributes and evaluate their potential to match profiles in practice. (3) Most studies used biased sets of ground truth users that willingly publish links to their profiles on different social networks. Our analysis reveals that such datasets have attributes that are more available and consistent, consequently, the reliability results of such schemes are overly optimistic [25, 20, 28, 36]. Other studies assume that all profiles that have the same screen name are matching [17, 14]. In §4.4 we showed that 20% of profiles with the same screen name in Twitter and Facebook are actually not matching. We further split these studies according to the type of attributes used.

The closest to our work are a number of schemes that leverage information in the *profiles of users* similar to the attributes we use in this paper [22, 28, 20, 25, 5, 34, 33, 29, 24, 27, 18, 36, 35, 34, 17]. Most schemes work by training classifiers to distinguish between matching and non-matching profiles. We simulated these approaches in §6 and we saw that, because they did not consider the problems that come with matching in practice, the matching schemes are very unreliable when evaluated in real-world social networks. A few studies attempted to perform profile matching in practice [20, 25, 5]. These studies, however, just pointed out that

profile matching in practice yields a large number of false matches. In contrast, we conduct a systematic analysis of the causes of such false matches and possible ways to eliminate them.

Other schemes use attributes extracted from *user activities* (i.e., the content users generate instead of attributes of the profile) [12, 21, 14]. These schemes reveal how even innocuous activities of users can help identify a user across social networks. However, these schemes explore attributes with either low availability or low discriminability, which makes them hard to use in practice without sacrificing reliability.

**De-anonymizing user identities:** De-anonymizing user identities and matching user profiles share common methods. In fact, our work here is inspired by one of the seminal papers of Sweeney [32], which explored the uniqueness of attributes such as date of birth, postal code, and gender to de-anonymize medical records of US citizens. Other studies [23, 15] showed the feasibility to de-anonymize the friendship graph of a social network at large-scale using the *friendship graph* of another social network as auxiliary information. The structure of the social graph is certainly a powerful feature. Nevertheless, in this work, we explicitly assume that we cannot have access to the entire graph structure of the social networks since we only use public APIs to collect data. We leave as future work how to exploit partial graphs that can be obtained through APIs to improve matching schemes based on binary classifiers.

**Entity matching:** There is a large body of research in the database and information retrieval communities on matching entities across different data sources [9]. Conceptually there are many similarities between matching profiles across social networks and matching entities (e.g. the way we compute the similarities between attributes or the adoption of a supervised way to detect matches). However, matching profiles has some specific constraints (e.g., not being able to access all records in  $SN_1$ ) that the entity matching community, to our knowledge, overlooked.

## 10. CONCLUSION

In this paper, we conducted a systematic and detailed investigation of the reliability of matching user profiles across real-world online social networks like Twitter and Facebook. Our analysis yielded a number of methodological and measurement contributions.

To understand how profile attributes used by matching schemes affect the overall matching reliability, we proposed a framework that consist of four properties – *Availability, Consistency, Impersonability, and Discriminability* (ACID). Our analysis showed that most people maintain the same persona across different social networks – thus it is possible to match the profiles of many users, however, in practice there can be a non negligible number of profiles that belong to different users but have similar attribute values, which leads to false matches.

We showed that the reliability of matching schemes that are trained and tested on reliability non-preserving sampled datasets is not indicative of their reliability in practice. In fact, traditional matching schemes based on binary classifiers can only achieve a 19% recall for a 95% precision to match Twitter to Facebook profiles in practice. To avoid these pitfalls we illustrated the right assumptions we can make about the matching problem and the correct methodology to evaluate matching schemes in realistic scenarios.

Finally, we proposed a matching scheme that is able to mitigate impersonation attacks and reduce the number of false matches to achieve a 29% recall for a 95% precision. Our matching scheme exploits a special case of the matching problem, namely that there exists at most one matching profile. Although we cannot claim that 29% is a high recall, humans cannot do much better (they only detect 40% of matching profiles).

## 11. REFERENCES

- [1] Bing Maps API. <http://www.microsoft.com/maps/developers/web.aspx>.
- [2] Phash. <http://www.phash.org>.
- [3] Spokeo. <http://www.spokeo.com/>.
- [4] Spokeo lawsuit. <http://www.ftc.gov/sites/default/files/documents/cases/2012/06/120612spokeocmpt.pdf>.
- [5] A. Acquisti, R. Gross, and F. Stutzman. Faces of facebook: Privacy in the age of augmented reality. In *BlackHat*, 2011.
- [6] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ECML*, 2004.
- [7] Get better results with less effort with Mechanical Turk Masters – The Mechanical Turk blog. <http://bit.ly/112GmQI>.
- [8] M. Balduzzi, C. Platzer, T. Holz, E. Kirda, D. Balzarotti, and C. Kruegel. Abusing social networks for automated user profiling. In *RAID*, 2010.
- [9] P. Christen. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-centric systems and applications. Springer, 2012.
- [10] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, 2003.
- [11] O. Goga. *Matching User Accounts Across Online Social Networks: Methods and Applications*. PhD thesis, UPMC, 2014.
- [12] O. Goga, H. Lei, S. Parthasarathi, G. Friedland, R. Sommer, and R. Teixeira. Exploiting innocuous activity for correlating users across sites. In *WWW*, 2013.
- [13] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE TKDE*, 2009.
- [14] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff. Identifying users across social tagging systems. In *ICWSM*, 2011.
- [15] N. Korula and S. Lattanzi. An efficient reconciliation algorithm for social networks. *PVLDB*, 2014.
- [16] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, V. Paxson, and S. Savage. Spamcraft: An inside look at spam campaign orchestration. In *LEET*, 2009.
- [17] S. Labitzke, I. Taranu, and H. Hartenstein. What your friends tell others about you: Low cost linkability of social network profiles. In *SNA-KDD*, 2011.
- [18] J. Liu, F. Zhang, X. Song, Y.-I. Song, C.-Y. Lin, and H.-W. Hon. What's in a name?: An unsupervised approach to link users across communities. In *WSDM*, 2013.
- [19] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.
- [20] A. Malhotra, L. Totti, W. Meira, P. Kumaraguru, and V. Almeida. Studying user footprints in different online social networks. In *CSOSN*, 2012.
- [21] M. A. Mishari and G. Tsudik. Exploring linkability of user reviews. In *ESORICS*, 2012.
- [22] M. Motoyama and G. Varghese. I seek you: searching and matching individuals in social networks. In *WIDM*, 2009.
- [23] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE S&P*, 2009.
- [24] C. T. Northern and M. L. Nelson. An unsupervised approach to discovering and disambiguating social media profiles. In *MDSW*, 2011.
- [25] P. K. Paridhi Jain and A. Joshi. @i seek 'fb.me': Identifying users across multiple online social networks. In *WoLE*, 2013.
- [26] Peekyou. <http://www.peakyou.com/>.

- [27] O. Peled, M. Fire, L. Rokach, and Y. Elovici. Entity matching in online social networks. In *SocialCom*, 2013.
- [28] D. Perito, C. Castelluccia, M. Ali Kâafar, and P. Manils. How unique and traceable are usernames? In *PETS*, 2011.
- [29] E. Raad, R. Chbeir, and A. Dipanda. User profile matching in social networks. In *NBiS*, 2010.
- [30] R. Schmid. Salesforce service cloud – featuring activation, 2012.  
http://www.youtube.com/watch?v=eT6iHEdnKQ4&feature=relmfu.
- [31] Social Intelligence Corp. http://www.socialintel.com/.
- [32] L. Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine, and Ethics*, 1997.
- [33] J. Vosecky, D. Hong, and V. Shen. User identification across multiple social networks. In *NDT*, 2009.
- [34] G.-w. You, S.-w. Hwang, Z. Nie, and J.-R. Wen. Socialsearch: enhancing entity search with social network matching. In *EDBT/ICDT*, 2011.
- [35] R. Zafarani and H. Liu. Connecting corresponding identities across communities. In *ICWSM*, 2009.
- [36] R. Zafarani and H. Liu. Connecting users across social media sites: A behavioral-modeling approach. In *KDD*, 2013.

## APPENDIX

### A. PROOFS FROM SECTION 3

#### A.1 Effective discriminability formula

In this section, we justify that, if we assume that the impersonating profiles are independent from the other non-matching profiles, we have  $\tilde{D} = D \cdot (p_{nI} + nI \cdot p_I)$ . We first apply the complete probability formula:

$$\begin{aligned}\tilde{D} &= Pr\left(\max_{a^2:a^2\text{-non-match-}a^1} s(v^1, v^2) < th\right) \\ &= Pr\left(\max_{a^2:a^2\text{-non-match-}a^1} s(v^1, v^2) < th \mid a^1 \text{ not impersonated}\right) p_{nI} \\ &\quad + Pr\left(\max_{a^2:a^2\text{-non-match-}a^1} s(v^1, v^2) < th \mid a^1 \text{ impersonated}\right) p_I. \quad (1)\end{aligned}$$

Then, we observe that the max on all non-matching profile is smaller than  $th$  iff both the max on all non-matching profile that are not impersonating  $a^1$  and the max on the impersonators of  $a^1$  are smaller than  $th$ . That is:

$$\begin{aligned}&Pr\left(\max_{a^2:a^2\text{-non-match-}a^1} s(v^1, v^2) < th \mid a^1 \text{ impersonated}\right) \\ &= Pr\left(\max_{\substack{a^2:a^2\text{-non-match-}a^1 \\ a^2\text{-non-imperso-}a^1}} s(v^1, v^2) < th, \quad (2) \\ &\quad \max_{a^2:a^2\text{-impersonate-}a^1} s(v^1, v^2) < th \mid a^1 \text{ impersonated}\right).\end{aligned}$$

If the impersonating profiles are independent from the other non-matching profiles, then the joint probability equals the product of probabilities:

$$\begin{aligned}&Pr\left(\max_{a^2:a^2\text{-non-match-}a^1} s(v^1, v^2) < th \mid a^1 \text{ impersonated}\right) \\ &= Pr\left(\max_{\substack{a^2:a^2\text{-non-match-}a^1 \\ a^2\text{-non-imperso-}a^1}} s(v^1, v^2) < th \mid a^1 \text{ impersonated}\right) \cdot \\ &\quad Pr\left(\max_{a^2:a^2\text{-impersonate-}a^1} s(v^1, v^2) < th \mid a^1 \text{ impersonated}\right) \\ &= D \cdot nI;\end{aligned}$$

which, given (1), shows that  $\tilde{D} = D \cdot (p_{nI} + nI \cdot p_I)$ .

#### A.2 Proofs of Theorem 1

In this section, we give a proof of Theorem 1. Recall that  $th$  is a threshold parameter in  $[0, 1]$  such that the classifier declares a match between  $a^1$  and  $a^2$  if  $s(v^1, v^2) > th$ .

To show (i), first recall the definition of recall:

$$recall = Pr(s(v^1, v^2) > th \mid a^2\text{-match-}a^1).$$

Then, we have

$$\begin{aligned}recall &= Pr(s(v^1, v^2) > th \mid a^2\text{-match-}a^1, v^1 \text{ and } v^2 \text{ available}) \\ &\quad \cdot Pr(v^1 \text{ and } v^2 \text{ available} \mid a^2\text{-match-}a^1) \\ &\quad + Pr(s(v^1, v^2) > th \mid a^2\text{-match-}a^1, v^1 \text{ or } v^2 \text{ not available}) \\ &\quad \cdot Pr(v^1 \text{ or } v^2 \text{ not available} \mid a^2\text{-match-}a^1).\end{aligned}$$

By convention,  $s(v^1, v^2) = 0$  if  $v^1$  or  $v^2$  not available (a pair is never declared a match by the classifier if either value is missing). Therefore,  $Pr(s(v^1, v^2) > th \mid a^2\text{-match-}a^1, v^1 \text{ or } v^2 \text{ not available}) = 0$  and we have  $recall = C \cdot A$  by definition of  $C$  and  $A$ .

To show (ii), first recall the definition of precision:

$$precision = Pr(a^2\text{-match-}a^1 \mid s(v^1, v^2) > th).$$

To ease the equations reading, we simplify the notation of  $a^2\text{-match-}a^1$  into simply match and similarly for  $a^2\text{-non-match-}a^1$ . By application of Bayes formula, we compute

$$\begin{aligned}precision &= \frac{Pr(\text{match}, s(v^1, v^2) > th)}{Pr(\text{match}, s(v^1, v^2) > th) + Pr(\text{non-match}, s(v^1, v^2) > th)} \\ &= \frac{recall \cdot Pr(\text{match})}{recall \cdot Pr(\text{match}) + Pr(\text{non-match}, s(v^1, v^2) > th)}.\end{aligned}$$

Let  $n_2$  denote the number of profiles in  $SN_2^6$ . By the assumption of Theorem 1-(ii), we have  $Pr(\text{match}) \leq 1/n_2$ , so that, since  $Pr(\text{non-match}, s(v^1, v^2) > th) \geq 0$ , we get

$$precision \leq \frac{recall}{recall + n_2 \cdot Pr(\text{non-match}, s(v^1, v^2) > th)}.$$

Moreover, by definition of  $\tilde{D}$ , we have

$$Pr(\text{non-match}, s(v^1, v^2) > th) \geq \frac{1 - \tilde{D}}{n_2},$$

which gives

$$precision \leq \frac{recall}{recall + 1 - \tilde{D}}$$

and concludes the proof of Theorem 1-(ii).

We now show (iii), by making three observations:

- a. First, observe that, from (i), we directly get that  $recall = 1$  iif  $A = C = 1$ .

<sup>6</sup>Note that  $n_2$  includes impersonating profiles and hence formally is a random variable. Rigorously, we should condition on the value of  $n_2$  and then take the expectation; however the result would be unchanged hence we omit this detail for a lighter presentation.



- b. Second, observe that  $\text{precision} = 1$  iff  $\tilde{D} = 1$ . Indeed, we have  $\text{precision} = 1$  iff  $\Pr(a^2\text{-non-match-}a^1, s(v^1, s^2) > th) = 0$ , which is equivalent to  $\Pr(\max_{a^2, a^2\text{-non-match-}a^1} s(v^1, v^2) > th) = 0$  and hence to  $\tilde{D} = 1$ .
- c. Third, observe that  $D = nI = 1$  implies  $\tilde{D} = 1$  and that, if  $p_I > 0$ , the converse holds too. We show the two separately.  
 $(\Rightarrow)$  : Assume that  $D = nI = 1$ . The result follows from the following facts: if  $D = 1$  then the first term of (1) multiplying  $p_{nI}$  is 1; and from (2), if  $D = 1$  and  $D = 1$  and  $nI = 1$ , then the second term of (1) multiplying  $p_I$  is 1. Therefore,  $\tilde{D} = 1$ .  
 $(\Leftarrow)$  : Assume that  $\tilde{D} = 1$  and  $p_I > 0$ . If  $D < 1$ , then both terms of (1) multiplying  $p_{nI}$  and  $p_I$  are strictly smaller than one which contradicts  $\tilde{D} = 1$ . Therefore  $D = 1$ . If  $nI < 1$ , the second term of (1) multiplying  $p_I$  is strictly smaller than one which contradicts  $\tilde{D} = 1$  since  $p_I > 0$ . Therefore  $nI = 1$ .

The combination of these three observations implies Theorem 1-(iii). (In fact, these three observations give more detailed results on the impact of ACID on precision and recall than what is summarized in Theorem 1-(iii).)

## B. ATTRIBUTE SIMILARITY METRICS

**Name similarity:** Previous work in the record linkage community showed that the *Jaro string distance* is the most suitable metric to compare similarity between names both in the offline and online worlds [10, 28]. So we use the Jaro distance to measure the similarity between real names and screen names.

**Photo similarity:** Estimating photo similarity is tricky as the same photo can come in different formats. To measure the similarity of two photos while accounting for image transformations, we use two matching techniques: (i) *perceptual hashing*, a technique originally invented for identifying illegal copies of copyrighted content that works by reducing the image to a transformation-resilient “fingerprint” containing its salient characteristics [2] and (ii) *SIFT*, a size invariant algorithm that detects local features in an image and checks if two images are similar by counting the number of local features that match between two images [19]. We use two different algorithms for robustness. The perceptual hashing technique does not cope well with some images that are resized, while the SIFT algorithm does not cope well with computer generated images.

**Location similarity:** For all profiles, we have the textual representations of the location, like the name of a city. Since social networks use different formats for this information, a simple textual comparison will be inaccurate. Instead, we convert the location to latitude/longitude coordinates by submitting them to the Bing API [1]. We then compute the similarity between two locations as the actual geodesic distance between the corresponding coordinates.

**Friends similarity:** The similarity score is the number of common friends between two profiles. We consider that two profiles have a common friend if there is a profile with the same screen name or real name in both friend lists. A more complex but potentially more accurate method would have been to apply a matching scheme for each friend recursively taking other features beside screen name and real name into account. As we will see, however, given two small lists of profiles on different social networks, real names and screen names alone can accurately identify matching profiles. Complementary, we could divide the number of common friends by the total number of friends. Preliminary results showed no particular improvement in doing so.