Matching user accounts across online social networks : methods and applications

Oana Goga

THÈSE

en vue d'obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE - PARIS 6 (UPMC)

spécialité : informatique
Laboratoire d'Informatique de Paris 6
École Doctorale Informatique, Télécommunications et Électronique

présentée et soutenue publiquement le 21 Mai 2014
par **Oana Goga**

Titre :

Matching User Accounts Across Online Social Networks: Methods and Applications

Corrélation des profils d'utilisateurs dans les réseaux sociaux : méthodes et applications

Directeur de thèse :

Mme.	Renata Teixeira	Inria Rocquencourt

Après avis de :

Mme.	Anne-Marie Kermarrec	Inria Rennes
M.	Jon Crowcroft	University of Cambridge

Devant la commission d'examen formée de :

M.	Jon Crowcroft	University of Cambridge
M.	Krishna Gummadi	MPI-SWS
Mme.	Clémence Magnien	CNRS
Mme.	Dina Papagiannaki	Telefónica I+D
Mme.	Renata Teixeira	Inria Rocquencourt

Acknowledgments

I would like to first express my special appreciation and thanks to my advisor Renata Teixeira for her advice and mentoring during my PhD. I am immensely grateful that Renata allowed me to change PhD topic after one year of PhD and gave me the liberty to explore research I am passionate about while supporting me when I needed her help. For me, changing topic was one of the best decisions in my life. It allowed me to work on projects I believe in and projects that I want to continue pursuing in the years to come as a researcher. It also allowed me to discover the focus I want to give to my career. Finally, Renata taught me how to communicate research results effectively and how to make a good presentation. Even if it is still scary, I love giving presentations thanks to you.

I would also like to thank Nina Taft and Robin Sommer. Nina advised me at a very confusing time of my PhD, your encouragements helped me in my decision to change PhD topic. I was extremely lucky to meet Robin at the perfect time and start working with him on my first project on privacy and social networks. It was a very exciting and scary time, but Robin gave me his constant support, without you none of the work presented in this thesis would have been possible. Also, Robin sees research projects with an apparent easiness and clearness. This inspired me and I often think "what would Robin say?" whenever I am stuck in a project.

I was also super lucky to meet and work with Krishna Gummadi during my PhD. My time at MPI-SWS was the most amazing part of my PhD. Krishna played a big role in my development as a researcher, Krishna taught me how to be bold and go for crazy research projects, how to constantly push my limits and go deeper and understand the essence of a problem. Krishna has amazing ideas and all our discussions were very enriching. Moreover, Krishna's depth in the field of social networks allowed me to put my work into perspective. Krishna provided for me, and all his students, a very nurturing environment perfect to rise high quality researchers. I will try to provide the same kind of environment for everyone I will work with in the future.

I hope I am worthy of the time and efforts Renata, Robin and Krishna invested in me.

I want to also thank Anne-Marie Kermarrec and Jon Crowcroft for accepting to review my thesis. Your reports are very precise and your remarks very pertinent. I also want to thank Clémence Magnien, Krishna Gummadi and Dina Papagiannaki for accepting to be in my thesis committee. I am very honored that such distinguished researchers took time to evaluate the work I did during my PhD.

I also want to thank my collaborators Gerald Friedland, Howard Lei, Sree Hari Krishnan, Diana Joumblatt, Jaideep Chandrashekar, Giridhari Venkatadri and Bimal Viswanath. It was very fun working with you and I learned a lot from our interactions. I am very grateful to have met amazing people during my stay at LIP6, LINCS, MPI-SWS, ICSI and Technicolor. I build many friendships that I hope will last a lifetime. I want to thank Jean Bolot and Christophe Diot for hosting me in their Technicolor lab in Palo Alto. Thanks to all students and researchers I have met during my stay in different labs I had an excellent time during my PhD.

Finally, I would like to thank my affectionate and supportive family for its constant support. I would like to thank my parents for encouraging me to go abroad to continue my education even if this meant being far away from them. Thank you also for inspiring me and showing me that there are no limits and I can pursue whatever I am passionate about. I want to thank "my other half", Patrick, even if it is hard to do this in only a few sentences. Patrick is the most amazing person I have met and I am very blessed to have him near me. Patrick constantly cheered me when I was down and helped me when I needed advice and aid on my research. Patrick taught me to be strong and push forward when it is hard. I thank you from all my heart for your help in all aspects of my PhD. My warmest thanks to Patrick's parents, Isabelle and Bernard, who hosted me in their home during my PhD when I had to travel back and forth between Nice, Paris and Saarbrucken. Thanks to you I was able to spend more time with Patrick during my internships.

Contents

Abstract

The proliferation of social networks and all the personal data that people share brings many opportunities for developing exciting new applications. At the same time, however, the availability of vast amounts of personal data raises privacy and security concerns. These opportunities are even better (and the concerns more serious) if we can correlate the data that a single individual publishes in different social networks. Additionally, within a single social network, the terms of service specify that users can only have a single account. Nevertheless, some users create multiple accounts and malicious users often impersonate honest users. Both for applications that correlate user info across social networks and for social networks that fight multiple accounts of a single user, we need matching techniques to identify the accounts of a single individual.

The main contribution of my thesis is the development and analysis of scalable and reliable matching schemes to match the accounts that correspond to the same individual in today's social networks. Matching accounts across social networks allows applications to work on more complete user profiles. Aggregating personal data across social networks, however, also rises privacy concerns, in particular, when we can match the accounts of users that deliberately change the information in their profiles to maintain separate personas. Finally, matching accounts within a social network is a powerful tool to detect impersonators.

First, we study how we can exploit the public profiles (e.g., users' real name, screen name, location, bio, and profile photo) that users maintain in different social networks to match their accounts. We identify four important properties – Availability, Consistency, non-Impersonability, and Discriminability (ACID) – to evaluate the quality of different profile attributes to match accounts. Exploiting public profiles has a good potential to match accounts because many users keep their names and other personal information consistent across different social networks. To demonstrate that matching accounts in real social networks is feasible and reliable enough to be used in practice, we focus on designing matching schemes that achieve low error rates even when applied in large-scale networks with hundreds of millions of users. Matching accounts in real social networks is very challenging because we have to deal with very large datasets and there are only a limited number of attributes that we can use to detect the matching account out of more than one billion. We develop a scheme in three-steps to achieve a good accuracy at scale. Instead of using one classifier to detect matching and non-matching accounts (which performs poorly at scale), we use two classifiers sequentially, trained with separate datasets, which better leverage the power of different attributes. Furthermore, we take advantage of the fact

that there can only be one matching account on a second social network to increase the reliability of the matching.

Then, we show that we can still match accounts across social networks even if we only exploit information about user posts, i.e., their activity on different social networks. Specifically, we show that by only exploiting the location, timing and writing style of a user's posts we can match his accounts across social networks. For example, if we use the location from where users post, we can match 60% of Flickr accounts with their corresponding Twitter accounts, while only introducing a small percentage of falsely matching accounts. This demonstrates that, even if users are privacy conscious and maintain distinct profiles on different social networks, we can still potentially match their accounts.

Finally, we show that, we can detect impersonators in social networks by firstly identifying accounts that portray the same person inside the social network and then using a classifier to detect which are the impersonating accounts in the returned list. Traditional methods to detect fake accounts perform poorly for detecting impersonators. Our study shows that detecting impersonators requires to build methods that exploit features that characterize pairs of accounts rather than features that characterize single accounts as done so far.

Chapter 1

Introduction

Today, over 2.4 billion users have access to the Internet and a large fraction of them have an active account on a social network. There are more than 1.2 billion active users on Facebook alone (728 million of these users log in daily), 540 million on Google+, 259 million on LinkedIn, and 232 million on Twitter [171]. Back in 2009, a study by Anderson Analytics [178] showed that 91% of Twitter users and 82% of LinkedIn users also have a Facebook account. We expect this overlap to be even higher in 2014. Users share all kinds of information on social networks at an enormous rate. For example Facebook users share 4.75 billion pieces of content daily. Every 60 seconds, users post 510 comments, update 293,000 statuses, and upload 136,000 photos on Facebook [141]. Users often engage in different activities and reveal information about different aspects of their lives on different social networks. On Facebook, users communicate with their friends and families and share aspects of their personal lives. On LinkedIn, users give details about their professional evolution and aspirations. On Twitter, users tend to post things they are passionate about.

There is a growing interest in identifying multiple accounts that correspond to a single individual. First, organizations are interested in correlating user activities and aggregating information across multiple social networks to develop a more complete profile of individual users than the profile provided by any single social network. Second, social networks are interested in finding all the accounts corresponding to a single individual inside a single social network. Users are supposed to open only one account in a social network (as stipulated in the Terms of Service), however some users create multiple accounts. Furthermore, malicious users often impersonate honest users. For both cases, we need matching techniques to find the accounts of a single individual.

We already see legitimate business models based on such correlation techniques. Many emerging companies try to automatically match and mine users profiles across different social networks to help recruiters in their decisions [166]. Modern sales portals combine crowdsourced phone information with social networking posts to present a customer profile to sales representatives and telephone agents in assisting hotline callers more effectively [159]. Furthermore, some companies like PeekYou [146] and Spokeo [4] market themselves as *"people search engines"*: starting with basic information such as screen names (i.e., a user's login) and real names, these services return user profiles collected from different social networks. Matching accounts across social networks also has application for many research problems. There is a lot of fundamental research on online communities

such as user influence estimation [188], user expertise estimation [117, 162], community structure and link analysis [139] and opinion mining [144]. Most prior research, however, has focused on single social networks and did not analyze social networks in aggregate. For example, researchers have studied the behavior of users on Facebook or LinkedIn separately [10, 18, 182]. This only provides, however, a partial view of a user. Interactions in Facebook will most likely only characterize interactions with friends, and interactions in LinkedIn will most likely only characterize interactions with co-workers. Knowing the matching accounts on multiple social networks provides the opportunity to build a better portrait of a user. Having a deeper understanding of a user can thereafter lead to better personalized services or better estimations of users' expertise. One important building block for any research based on cross-site account matching is to have reliable techniques to match accounts across social networks.

While the creation of such complete profiles of users has many applications in industry and research it also rises legitimate and serious concerns about the privacy of users online. While on a per-site basis, a user may deem fine what she posts to her Facebook, Twitter, and LinkedIn accounts, she might be revealing much more than she realizes when considering them in *aggregate*. As an example, a social engineering attack could first identify employees of a victim organization on LinkedIn, and then examine their Facebook accounts for personal background to exploit while also following their tweets to understand travel patterns.

In this thesis, we develop methods to identify the social networks accounts of a given user. We first study how we can exploit the public profiles (e.g., users' real name, screen name, location, bio, and profile photo) users maintain in different social networks to match their accounts. We identify four important properties – Availability, Consistency, non-Impersonability, and Discriminability (ACID) – to evaluate the quality of different profile attributes to match accounts. Exploiting public profiles has a good potential to match accounts because a large number of users have the same names and other personal information across different social networks. Yet, it remains challenging to achieve practically useful accuracy of matching due to the scale of real social networks. To demonstrate that matching accounts in real social networks is feasible and reliable enough to be used in practice, we focus on designing matching schemes that achieve low error rates even when applied in large-scale networks with hundreds of millions of users. Then, we show that we can still match accounts across social networks even if we only exploit what users post, i.e., their activity on a social networks. This demonstrates that, even if users are privacy conscious and maintain distinct profiles on different social networks, we can still potentially match their accounts. Finally, we show that, by identifying accounts that correspond to the same person inside a social network, we can detect impersonators.

1.1 Personal data sharing in social networks

A large part of the data shared by users is public as the purpose of having a profile in many social networks is to make an individual more visible. The purpose of a LinkedIn profile is to have a better visibility to potential recruiters, while the purpose of a Twitter profile is to reach a large audience to promote ideas and interests. Simply put, the reason

to make content public is either to reach or be reached more easily by other users. As a result, however, anyone can easily access this content and mine it in ways that are out of the control of the user generating it.

Generally, we can find three types of information about users in a social network. First we have the *user profile* which is the information users provide about themselves. The profile information can include attributes like the name of the user, the city where he is currently living, where he went to school, his current employer, his birthday, or the profile photo. Beside the user profile we also have information about the *user activities* in the social network such as what a user posts or what books, movies, or sports teams a user likes. The posts of users often come with metadata information. For example, in Facebook and Twitter posts can be tagged with the exact geo-location from where the message was sent. On Yelp, users review restaurants, so we have information about the location and type of restaurant. Other types of metadata include hashtags, the language of the post or what application was used to post. Finally, we have information about the *social graph*: who the user is following and who are his friends.

All social networks allow users to make parts of their content private. Even with the most restrictive settings, however, there are always some pieces of information that have to be public. The minimum amount of information that always stays public in any social network is the the real name, the screen name, and the profile photo of a user.

1.2 Current techniques to match accounts

Today a number of organizations already aggregate information of a single user across multiple sites. There are four current approaches for matching accounts: (i) sites that allow users to explicitly show links to their profiles on different social networks; (ii) sites that support single sign-on services which implicitly match accounts; (iii) people search engines that use proprietary algorithms that exploit user names; and (iv) algorithms proposed by researchers that exploit different kinds of informations in users' profiles. We discuss these approaches in more detail.

Social networks like Google+ allow users to list their accounts on other social networks. People, brands, and companies that want to have a strong web presence are often advised to maintain active accounts on different social networks and link from one account to another. Linking accounts together increases their visibility and can also increase their position in Google search [161]. Besides Google+, there are also dedicated sites such as `itsmyurls.com`, `about.me`, and `hi.im` that allow users to create a page where they publish links to all their accounts. These sites allow users to better manage their online footprint. Besides some users that want to maintain a strong web presence, there are not many other users that use these services to list their accounts.

Single sign-on is a type of user authentication that allows a user to enter one name and password to connect to multiple sites or applications. OpenID [155] is the traditional web single sign-on solution. It failed, however, to get large adoption from both sites and users mainly because of the lack of business incentives [172]. In the past years, Facebook, Google+, Twitter, and LinkedIn started to allow users to connect to other sites with their

social networks accounts. This is part of an effort to allow third-party sites and applications to integrate with social networks. We can see the integration between Facebook, Google+, Twitter and LinkedIn with third-party sites as an *implicit matching* of accounts between the two sites. Facebook integration is already very popular, more than 24.3% of the top 10,000 websites have some form of integration [150]. This solution received a much larger adoption because the integration is beneficial for both site owners and social networks. When users log in with their Facebook accounts on a third-party site, Facebook shares some information about the user such as their age, current city or likes. At the same time, Facebook benefits by allowing and making it very easy for users to share on Facebook any activity they do from anywhere on the Internet. Consequently, a few main social networks could become both aggregators and dispatchers of large parts of data generated by users online. However, there is no implicit matching between different social networks.

Current people search engines such as `peekyou.com`, `spokeo.com`, `wink.com`, `pipl.com`, and `zabasearch.com` aggregate information about users without their explicit consent or knowledge. Most people search engines aggregate data from public records, surveys, and social networks. Usually, when queried, they just return all the accounts of people sharing the same name. Since in most cases, the names of people are not unique the results can lead to many false matches. Spokeo explicitly stipulates in their Terms of Use that they do not guarantee the accuracy of their data. Some people search engines use more sophisticated algorithms. PeekYou filed a patent [74] for matching people's names to their accounts on blogs, social networks, and forums. The algorithm consists mainly in using the information collected on different sites and assigning empirically different weights to different pieces of information to match identities. These heuristics are not reliable as pointed out by Perito et al. [148]. To build effective services it is crucial to understand the limits and the capabilities of such matching techniques.

Finally, there is a flurry of recent research efforts directed towards matching accounts across different social networking sites [6, 16, 76, 122, 128, 134, 137, 145, 148, 195]. These efforts leveraged a variety of account features ranging from public user profile attributes to user generated content and private data to match accounts. This work allude to the potential for conducting large-scale account matching, however, most of these studies have not evaluated their schemes at scale [76,128,134,148,195] and the few that did, found that their matching schemes tend to be very unreliable [6,122,145], i.e., they have large numbers of false matches. So the problem of reliable and scalable account matching remains an open challenge.

1.3 Privacy and security threats in social networks

People are increasingly interested in online privacy with the mediatization of the possible risks of sharing personal content. We can see strong reactions whenever there is a data breach and many lawsuits have been launched against Google, Yahoo or Target [176]. As a consequence, many governments and organizations are proposing ways to regulate online privacy [175].

The problem of protecting the privacy of people online is hard. Every day, we discover new attacks on people's online privacy and we do not have yet a clear view of all possible

attacks. Thus, it is hard to know what measures to take and how comprehensive they are. Typically, much of the attention of governments and organizations focuses on how big players such as Facebook or Google and advertisers such as DoubleClick track our browsing patterns through cookies. Privacy advocates also monitor for changes in the properties of individual sites, such as specific sharing settings on Facebook and the new Google's terms of service.

What has been overlooked so far is a broader threat of attackers correlating personal information *across site boundaries*. Indeed, we can learn much more about a user when we know his accounts on several social networks. On Facebook, we can learn personal detail about an individual, for example, his birthday, his favorite movies, and where he went to school. We can also infer information. For example, on Twitter we can infer the interests of a person by analyzing the text of his tweets or who he follows [127, 162]. On LinkedIn, we have all the professional present and past of a user, we can learn the companies he worked for and what his competences are. On Yelp, we can learn what kind of cuisine the user likes and where he likes to go out. Hence, it is even more challenging to protect users' online privacy when attackers can correlate information across site boundaries and infer new information about users. The first step to mitigate this threat is to be able to measure the online footprint of a user. Understanding how easily we can match the accounts of an individual across social networks can provide users with tools to measure their online footprint and better understand possible privacy risks.

Our discussion so far has focused on the privacy threat of aggregating users' information across social networks. Another class of threats for user's online image comes from impersonators. There are more and more anecdotal evidences of celebrities and important people being impersonated [126, 160] but other people are potential impersonation targets too. Since an impersonator can seriously affect the online image of a user, it is very important to detect such attacks. So far, however, most of the attention of researchers and social network administrators has focused on detecting fake accounts or spam [17, 25, 169, 183, 185, 196, 200]. Currently, there is no framework to automatically detect impersonators and the only solution is for the victims to manually report the accounts who are impersonating them [75].

1.4 Contributions

In this thesis, we develop and analyze methods to identify the accounts that refer to the same person across different social networks or inside a social network. We exploit the techniques to match accounts inside a social network to detect impersonators online. We summarize the contributions of the thesis as follows.

1.4.1 Reliable and scalable account matching across social networks

There is lot of interest and concern, both in research and industry, about the potential for matching the accounts of a user across multiple online social networking sites. We focus on the challenge of designing account matching schemes that leverage publicly visible attributes to achieve high *reliability*, i.e., low error rates, in matching accounts, even

when applied in *large-scale* networks with hundreds of millions of users. The main challenge in achieving a reliable and scalable matching comes from the noise inherent from publicly available data. We identify four important properties – Availability, Consistency, non-Impersonability, and Discriminability (ACID) – to evaluate the quality of different attributes to match accounts. Public attributes like name, location, profile photo, and friends satisfy the ACID properties to different extents which makes the detection of matching accounts using simple machine learning classifications techniques inaccurate when applied at scale.

We show that it is possible to leverage multiple attributes to build a reliable and scalable matching scheme by doing the classification in three-steps: first filter out accounts that are clearly different, then disambiguate the true matching account out of similar looking accounts, and finally ensure reliability by measuring how distinct the true matching accounts is from other similar looking accounts. Previous techniques to match entities are not effective on matching accounts because of the constraints and specifics of our scenario: (1) we have to deal with very large datasets; (2) we only have a limited number of features to discriminate the matching account out of more than one billion; and (3) we are sure that there can only be one true matching account in a social network. We evaluate the performance of matching accounts from Twitter and Facebook, two of the largest real-world social networks. Our results show that matching accounts at large scale is challenging. Still, our techniques can match 30% of Twitter accounts with their corresponding Facebook accounts with 92% precision. Our findings reflect the potential as well as the limits of reliably matching accounts at scale using only public attributes of user accounts. Besides the analytical contributions, we also developed an online service that takes as input an account in Twitter and searches in real-time the matching account on Facebook, which can be found at at http://matchingaccounts.app-ns.mpi-sws.org/.

In the pursuit of building a reliable and scalable matching scheme, we also make some methodological contributions. First, we develop an unbiased method to gather ground truth on matching accounts. We believe this method gives a representative sample of users in general. Second, we propose a systematic evaluation of matching schemes that reveals that the accuracy of a scheme at small scale is not indicative of the accuracy of the scheme at large scale. Last, we evaluate how well humans can match accounts. The accuracy of humans is a more realistic standard of the accuracy we can expect an automatic matching scheme to achieve.

1.4.2 Account matching by exploiting innocuous user activities

We study how potential attackers can identify accounts on different social network sites that all belong to the same user, exploiting only innocuous activity that inherently comes with posted content. We examine three specific features on Yelp, Flickr, and Twitter: the geo-location attached to a user's posts, the timestamp of posts, and the user's writing style captured by language models. We show that amongst these three features the location of posts is the most powerful feature to identify accounts that belong to the same user in different sites. When we combine all three features, the accuracy of identifying Twitter accounts that belong to a set of Flickr users is comparable to that of existing attacks that exploit screen names. Our attack can identify 37% more accounts than using screen names

when we instead match Yelp and Twitter. Our results have significant privacy implications as they present a novel class of attacks that exploit users' tendency to assume that, if they maintain different personas with different names, the accounts cannot be matched together; whereas we show that the posts themselves can provide enough information to match the accounts.

1.4.3 Detection and characterization of impersonators

People are aware that attackers impersonate accounts in social networks. Apart from some anecdotal evidence, however, there has been no in depth characterizations of impersonation accounts in today's social networks. We propose a technique in two steps that detects impersonating accounts. The first step returns accounts that portray the same person in a social network. The second step detects which account is an impersonator. Traditional methods to detect fake accounts perform poorly for detecting impersonators. We show that for detecting impersonating accounts we have to build methods that exploit features that characterize pairs of accounts rather than features that characterize single accounts as has been done so far. We do a characterization study of about 5,693 cases of impersonation attacks we catch on Twitter. We found that impersonation attacks do not only target celebrities but also target less popular Twitter users. Furthermore, their main goal is to evade Twitter fake account detection rather than use the accounts for social engineering attacks. Our findings reveal a new type of impersonation attacks that can impact negatively the online image of any user, not just that of celebrities.

1.5 Organization of the thesis

This dissertation is organized as follows. Chapter 2 presents related work on analyzing social data, privacy and security threats in social networks and methods to match entities. Chapter 3 presents the account matching framework and the ground truth data for matching accounts. Chapter 4 presents our method to reliably and salably match accounts. Chapter 5 shows how we can match accounts using only innocuous user activities. Chapter 6 characterizes impersonators and presents ways to automatically detect them. We conclude in Chapter 7.

Chapter 2

State of the Art

The problem of matching accounts across social networks is related to problems tackled in different research communities ranging from security and privacy to database and data mining. In this chapter, we first review the state of the art on measuring and characterizing the information users leave on different social networks. We then give an overview of the privacy implications of sharing content online. Finally, we review the main matching techniques proposed in the database, information retrieval, and data mining communities. Even if there are not many studies that focus on the particular problem of matching accounts across social networks, these communities have worked on closely related problems like matching records across databases and anonimizing/de-anonimizing databases. Finally, we review some of the current efforts to detect fake social networks identities.

2.1 Social networks

This section reviews works on characterizing the amount of information we can learn about users from their posts and provides the motivation for investigating techniques to match accounts across social networks. We first discuss related work that measured how much and what kind of information users leave online, then overview studies that showed how this information can be exploited to further infer information that is missing or private and we finally review studies that measured the users' footprint across different social networks.

2.1.1 Measurement and characterization of social networks

Many studies have analyzed the content, the structure, and the evolution of social networks [7, 10, 11, 18, 65, 96, 98, 107, 109, 112, 129, 190]. We first review studies that measured the type and amount of content users share online and then studies that analyzed the interactions and the graphs of social networks.

Gross and Acquisti [65] were the first who studied what kind of information users share on Facebook and what are their privacy implications. Back in 2005, they observed that users willingly provide many kinds of information ranging from their names, location and photos to interests (books, music, movies), political views and sexual orientation, including

even date of births, phone numbers and email addresses. Similarly, Humphreys et al. [73] analyzed the personal information users provide on Twitter. A quarter of tweets include information about where a user is or what activity he is doing and most of this information is publicly accessible. Lampe et al. [111] studied the effect of the types of profile attributes users provide on Facebook on the number of friends they have. They found that the presence of profile attributes that help users share common references (e.g., school, employer) is strongly associated with the number of friends. The association is weaker for attributes related to the user interests. After almost 10 years, the well publicized debate about users' online privacy caused users to limit the access to some of their personal data online [119]. However, large parts of personal data still remain accessible to public. So far, most of the studies have only looked at social networks separately and did not consider them in aggregate. In contrast, in Chapter 4, we investigate what kind of information users provide on different social networks and how consistent this information is across social networks. Our study is useful for both understanding the extent to which we can match accounts across social networks and also gives insight into the behavior of users on different social networks.

A number of studies have looked at the structure of social network graphs and patterns in users activities [10, 18, 98, 129]. More recently, there has been an increased interest in augmenting the social graph with user attributes – Social Attribute Network (SAN) [62]. Such augmented networks are useful for link prediction, attribute inference, community detection and potentially accounts matching [16]. In this thesis, we only exploit one-hop friendship links to match accounts without exploiting the full social graph. We leave such study for future works.

2.1.2 Inferring additional information about users in social networks

Social networking sites allow users to hide parts of their personal profiles from the public, however there are always some pieces of information that remain public. This mix of private and public information can be exploited to infer private attributes of users. In this section, we first review studies that inferred different kinds of information about users by exploiting friendship links or other kinds of data in social networks and then we discuss studies that inferred the location of posts and photos.

Several studies exploited the social network graph to infer private information [71, 83, 115, 116, 130]. The studies are based on the *homophily* [125] assumption that users tend to relate to other users sharing similar traits. For example, political affiliations of friends tend to be similar, or students also have other students as friends. Gayo-Avello [56] showed that we can determine the political orientation, religious affiliation, race, ethnicity and sexual orientation of Twitter users with a 95% confidence by exploiting the network neighborhood. Similarly, Backstrom et al. [13] showed that the location of users can be inferred from the location of their friends. Zheleva et al. [201] showed that the inference accuracy can be improved if we also consider group membership besides network neighbors. Finally, Gong et al. [61] proposed to jointly predict links (friendships between users) and infer attributes in social network. By combining the two problems, both the accuracy of link prediction and the accuracy of inferring attributes increases.

Besides the social graph, researchers have also exploited other types of information present inside social networks. Chaabane et al. [29] leveraged interests and likes on Facebook to infer otherwise hidden information about users, such as gender, relationship status, and age. Calandrino et al. [24] exploited the outputs of collaborative filtering to infer customer transactions. Popescu et al. [152] used Flickr tags to find the gender and the home location of users. Lieberman et al. [114] determined the location of Wikipedia users from the articles they edit. Cheng et al. [32] use the content of tweets to determine the city-level location of a user. Finally, Staddon [168] showed that we can learn the hidden connection of a LinkedIn user with a very simple attack using a fake account. More generally, researchers have inferred extra information from publicly available records. Griffith et al. [64] showed that it is possible to infer the mother's maiden name from public records. Farkas and al. [46] discussed the inference problem across databases. While this body of work is not directly related to our matching method, it shows that, when we know a piece of information about a user, we can always infer more. This explains why matching accounts is appealing to industries and attackers.

In another line of work, researchers used publicly available information from a social network site and other external sources to infer the location of users posts and photos. Hecht et al. [72] derived user locations from tweets using basic machine learning techniques that associate tweets with geotagged articles on Wikipedia. Similarly, Kinsella et al. [91] leveraged tweets with geotags to build language models for specific places; they found that their model can predict country, state, and city with similar performance as IP geolocation, and zip code with much higher accuracy. Crandall et al. [39] located Flickr photos by identifying landmarks via visual, temporal and textual features. In Chapter 5, we will show that we can match accounts across social networks by exploiting the location of the users' posts. So far, our method only uses the posts that have geotags, however we could potentially expend it to include posts that do not have geotags if we can infer their location.

2.1.3 Aggregating users' data across social networks

Prior work also studied the aggregate footprint users leave across multiple social networks [31, 77]. Irani et al. [77] showed that, in average, users reveal four personal information fields (e.g., names, location, school) in one social network. However, users reveal different attributes on different social networks, thus if we know their accounts on multiple social networks, we can learn more about users. To create aggregate profiles of users, Pontual et al. [151] proposed to build a crawler that, given a name, is able to collect information from different social networks and sites. While inferring and aggregating information across social networks is appealing for building applications and services, it can also breach the privacy of individual users. We will review in the next section possible privacy threats.

2.2 Online privacy

In this section we start by overviewing privacy research in two areas: tracking users online and the privacy of online services. These research areas are not directly related with the

privacy implications of sharing content online, but, they expose different privacy threats users encounter online. We then focus on privacy implications of sharing location data which is related to our work in Chapter 5 that exploits location data to match accounts. Finally, we overview privacy threats caused by matching data about users across different sources.

2.2.1 Online tracking and advertising

Discussions on online tracking and advertising became lately very popular in both media and research communities because users are generally bothered by the fact they do not have control over what data companies are collecting and aggregating about them.

Websites such as lemond.fr or nytimes.com authorize other third-party websites such as Google Analytics or DoubleClick to track their users through cookies. Third-party websites allow (first-party) website to easily implement advertising, provide site analytics or provide integration with social networks. While third-party websites provide tremendous benefits for first-party websites, they can also severely affect the privacy of users. Third-party sites aggregate the browsing activities of users across unrelated first-party sites to create aggregate browsing profiles for better targeted advertising. Even if the aggregate browsing profiles are not directly linked to the users real identities, many users consider them a privacy breach. The creation of aggregate browsing profiles has been criticized by consumer advocates, policymakers and even marketers themselves. Numerous research efforts have measured and analyzed the ecosystem of advertising and tracking users online [67, 69, 99–102, 124, 131, 194]. While technology researchers have provided tools to block such tracking [156], policymakers proposed laws to limit or disclose tracking [132]. If companies start to massively aggregate and exploit the data users share online it is possible that users and policymakers will react the same way and claim their privacy rights.

Because online advertising supports the free content on the Internet, blocking all tracking will significantly affect the economics of the ecosystem. To overcome this issue, researchers are working on privacy-preserving tracking [49, 68, 179].

2.2.2 Privacy of online services

We look at the privacy of online services in two situations. First, we assume that users trust and use a service (e.g., Facebook, Yelp), however, the trusted service may leak personal information to other untrusted services, either intended or unintended. Second, we assume users have the ‘big brother syndrome’ and do not trust the service, i.e. they do not feel comfortable to give their private data to service providers such as Facebook or Twitter. Personal information can leak from a trusted party to an untrusted party in different ways: through Wi-Fi networks, from applications (e.g., mobile applications, software, or browsers) and from social networks. We first review studies that measure such leaks and we then review studies that propose privacy preserving services.

Whenever we connect to a Wi-Fi network, we are susceptible to leak private information. Large portions of the network traffic generated by a computer are unencrypted, and someone connected to the same Wi-Fi network can see what information is transmitted.

Consolvo et al. [37] tried to increase the awareness of this type of leakage by proposing a tool that alerts a user whenever private information such as email address or credit card number leaves the computer unencrypted. Cunche et al. [41] showed that it is possible to infer that two users live together by exploiting messages containing the SSIDs of users' preferred wireless networks in the active service discovery phase. To protect the privacy of users in wireless networks, Greenstein et al. [63] proposed a link layer protocol that obfuscates MAC addresses.

Users can grant access to mobile applications to many kinds of private information ranging from their exact location to the contacts in their address book or their IMEI (International Mobile Station Equipment Identity). After giving permission, users lose the control of what the application is doing with this information. Several studies analyzed the application and network traces generated by smart phones to quantify such leackages [45, 87, 158, 189]. For example, Cox et al. [189] found that half of the applications they tested were sending location information to advertisers. Browsers are also susceptible to information leakage. Studies showed that the browsing history of a user can be sniffed through side channel attacks or caching [79, 187].

Private information can leak inside a social network or from a social network to third parties. Thomas et al. [177] showed that information made private by a user can be inadvertently made public through conflicting privacy policies of users. Krishnamurthy and Willis showed that social networks can leak informations such as screen names, IDs and locations to third parties through referral headers or request-URLs [103, 104]. This has important privacy implications since advertisers could potentially link the anonymous tracking profiles they hold to social network identities. This line of research is complementary to ours and shows alternate ways in which the privacy of users can be compromised.

A few research efforts proposed distributed social networks to avoid giving up private information to companies [8, 14, 23, 40, 42]. While these solutions will protect the personal data of users from big companies, it is still possible for a third-party to match the accounts users have on different 'private' social networks. Furthermore, creating aggregate profiles of users can potentially be easier as there is no central service that detects information harvesting.

2.2.3 Privacy of location data

Real-time access to users exact location has lead to many innovative and useful services. Fine-grained location information enabled the creation of applications to recommend restaurants around the current user location, call taxis or map the photos taken during a trip. Sharing location info, however, yields serious privacy concerns. In Chapter 5, we will show that we can match accounts belonging to the same individual by exploiting only the location information that comes with posts and photos. In this section, we review studies that looked at different aspects of location privacy: (1) how to share location data with service providers in a privacy preserving way; and (2) how to anonymize or de-anonymize mobility traces.

To guarantee privacy while providing a reasonable level of service when sharing location information with sites or applications, researchers have proposed both cryptographic [50,

57] and non-cryptographic [19, 66, 88, 123, 138] techniques. Non-cryptographic techniques include sharing the location at a coarser grain, cloaking or using a trusted third-party service that provides k-anonymity. For a comprehensive picture we refer the reader to existing surveys [82, 106, 133, 157]. Finally, Shokri et al. [163, 164] proposed a framework to formally quantify the location privacy (being able to predict where a user is at a particular moment) in order to compare the accuracy of different privacy preserving location sharing techniques. While these techniques were aimed to protect users privacy with respect to service providers, such obfuscation techniques could be also used to protect users' privacy online when publishing posts or photos with geotags. Such obfuscation techniques will not have a strong impact on our matching scheme because we only need coarse-grain location data.

A different line of research focuses on privacy implication of publishing anonymized location datasets. Sharing such datasets is tremendously useful and allows to study human mobility and behavior. Consequently, many researchers have gathered such datasets and make them public [97]. Usually, the mobility of a user is represented as series of location-time pairs. The granularity of the location can be at either cell level or exact latitude and longitude depending how the data is collected. Researchers have shown that we can use these datasets to build a location profile for each user in the form of a random walk or a Markovian model and we can use this location profile to de-anonymize mobility traces very easily with the help of some auxiliary location data [43, 121]. The techniques proposed to de-anonymize mobility traces do not apply in our case. Mobility traces contain thousands of location-time pairs while the median number of such pairs is less than 10 on Twitter for example. The lack of data makes it impossible to model location profiles with random walks or Markovian models. Consequently identifying location profiles that correspond to the same user in social networks is more challenging than de-anonymizing mobility traces. Other studies showed that we can learn the home and work address of a user from these mobility traces [51, 105]. The home and work location of a user can then be matched with census databases to find the real identity of a user. Our matching scheme that exploits location data is based on a similar concept: we identify the main locations from where a user is posting and we match them across social networks. Closer to our work is the study of Zang et al. [199] who studied the k-anonymity of the top locations from where users make phone calls. They found that, at zip code level, the top three locations are unique for most of the people. Our results in Chapter 5 confirms their results, but when studying where users post form. Zang et al. [199] did not further explore how these top three locations could be matched with social identities. Finally, Srivatsa et al. [167] explored how mobility traces can be de-anonymized by correlating their contact graph with the friendship graph of a social network. Rather than correlating the location profiles of users in mobility traces with location profiles of users in social networks, Srivatsa et al. [167] chose to exploit mobility traces in a different way by creating a contact graph. We cannot use contact graphs to match accounts because we only have locations where users post which is a small subset of the locations a user passes by.

2.2.4 Privacy implications of matching and exploiting users' data

Prior work has investigated privacy implications of exploiting users' data. We first review works that exposed privacy implications of matching data across different data sources (e.g., public records, health records) and across different social networks. We then discuss works that showed different privacy and security implications of sharing content online.

Researchers have realized long ago that matching data across different data sources can pose privacy problems [34]. For example, matching different health databases could lead to the conclusion that a celebrity has a contagious disease [30]. Traditionally, data matching has been used for "connecting the dots" [174], i.e., for identifying terrorist threats. While this has clear benefits, falsely matched pairs of records might have severe privacy implications. If an individual is falsely detected as being involved in a crime or terrorist attack both his life and credit worthiness could suffer [85]. Furthermore, criminals can match records to collect enough identifying data to commit identity fraud [89, 143]. In this context, researchers rose a series of concerns about the privacy of public administrative and medical records [47, 55, 113, 165].

Focusing on social networks data, Fiedland et al. [52, 54] showed that a malicious user can use data that is publicly available on the Internet to mount social engineering attacks (cybercasing). They show that it is technically possible to use seemingly innocuous information to create correlation chains that tell much more about the individuals than they realize. A possible attack could first identify on Craigslist photos of precious objects that have geotags attached to them. From geotags we can infer the address of the owner while on Facebook we can detect when the owner is on vacation. An attacker could use this online information to mount a real-word robbery. Similar techniques can be used for economic profiling, espionage targeting, cyberframing or cyberstalking [52].

Several websites highlight privacy risks of sharing data on social networks.[1] `Sleeptime.org` estimates sleep patterns of Twitter users from their posts. `Stolencamerafinder.co.uk` crawls for digital camera serial numbers in online photos in order to find pictures taken with stolen cameras. `Icanstalku.com` publishes geotags found in tweets, and `pleaserobme.com` uses status updates form social networks to locate users who were currently not at home but had published their home address. The `cree.py` application uses geolocation data from social networks and media hosting services to track a person's movements.

Finally, researchers showed that personal data can help to crack passwords. Irani et al. [78] suggested to use personal data collected across multiple social networks to gather answers for password recovery questions. Castelluccia et al. [27] showed that personal data can also be used to reduce the number of attempts of brute force password cracking. Finally, Jagatic et al. [80] show that phishing attacks [81] have a significantly higher success rate when they consider the victim's social context.

This section only described a few examples of how public data that users share online can improve real world attacks. Automated techniques to match user data across social networks will likely lead to even more sophisticated attacks.

[1] Some of the sites are no longer working.

2.3 Matching entities across data sources

There is a large body of research in the database and information retrieval communities on matching entities across different data sources. Even if they are not directly matching accounts across social networks, they address similar problems. In this section, we first present methods used for matching in entity resolution and record linkage, we then discuss research on anonymization and de-anonymization of databases.

2.3.1 Entity resolution

Entity resolution or record linkage is the task of detecting records that refer to the same individual across different databases. Because this task has many different application domains, different research fields adopted different terms for the same task, for example, duplicate detection, deduplication, reference matching, object identification, merge/purge, object consolidation, or reference reconciliation. Entities most commonly corresponds to people (e.g., patient, customers, tax payers, travelers), businesses, consumer products or publications and citations. The two traditional applications domains that started to use entity resolution a few decades ago are the health sector and census databases. More recently, entity resolution has been used to match entities after two companies merge, to detect national security threats, to build online digital libraries and for e-Commerce, to find the records referring to the same product. The main challenge in entity resolution is that there is no unique identifier and that records are not consistent across databases. Consequently, researchers from database, data mining, and knowledge engineering communities proposed a number of methods to accurately match entities based on approximate similarities of records. Most of the techniques work for a specific application or data type, but there are a few concepts that hold across different research lines.

Records are composed of different fields such as name, address, and date of birth. Methods to match entities have usually three main steps: (1) pre-process data to put it in a comparable format, (2) measure the similarity between records, (3) decide whether the records match according to the previous computed similarity. Researchers have proposed many methods for each of these steps, please refer to the book by Peter Christen [34] or the many surveys in the area [20, 35, 44, 94, 191, 192] for more details. Metrics to measure the similarity between records can be defined per record or per field. A typical per field metric is the Jaro distance [36] to measure the similarity between names. A typical metric to compare records is the number of common words between records. To decide whether two records match or not, researchers proposed three classes of methods: supervised, unsupervised, and rule-based (made by a field expert). Our strategy to match accounts across social networks is similar: we pre-process the data, we compute account similarities per field, and we use a supervised approach to identify whether two accounts match.

Unfortunately, the methods proposed in entity resolution are not directly applicable for matching accounts across social networks for three main reasons. First, we have to deal with very large datasets, for example Facebook has over 1 billion users, thus the data contains one billion matching accounts and one quintillion non-matching accounts. Second, the number of features we can use to match accounts is relatively small because there are not many profile attributes available across multiple social networks. The most common profile

attributes are: name, location, friends, and profile photo. Because of these reasons, the accuracy of traditional matching methods is not high enough to provide reliable matching. To overcome this problem, instead of using one classifier to detect matching and non-matching accounts, we use two classifiers sequentially that achieve better accuracy. Finally, given the social networks policies, there can only be one account per user per social network. We can take advantage of this to increase the reliability of the matching. Chapter 4 details our algorithm and strategies.

2.3.2 Anonymization and de-anonymization of user identities

One could view de-anonymizing user identities as matching accounts between unanonymized and anonymized user accounts. In fact, our work is inspired by Sweeney's [173] seminal work on de-anonymization, which explored the uniqueness of attributes such as date of birth, postal code, and gender to de-anonymize medical records of US citizens. Similarly, we leverage public attributes for matching accounts. Nevertheless, we have to deal with a noisier environment, where data is not always available or consistent and accounts can be impersonated. We first review prior work that studied the anonymity of different demographics when sharing anonymized administrative datasets and then we discuss techniques to de-anonymize social networks accounts through either the graph structure of the social networks or other kinds of information present in users' accounts.

A number of studies investigated the anonymity of different demographics in public data [58, 59]. Golle et al. [60] studied the anonymity of home and work locations of US citizens and found that the median anonymity set, i.e., number of people with the same home and work locations, at census block[2] is one (i.e., can uniquely identify a person). This is exactly what we want to leverage for matching accounts by exploiting the location metadata in users' posts.

Backstrom et al. [12] were the first to propose methods to de-anonymize social networks. They created sybil accounts that link to the identities that the attacker wants to identify. Because it requires the creation of many sybil accounts this approach is not scalable. Narayanan et al. [137] showed the feasibility of de-anonymizing the friendship graph of a social network at large-scale using the friendship graph of another social network as auxiliary information. They were able to match 30% of the accounts with a 12% error rate. In a different scenario, Srivatsa et al. [167] explored how mobility traces can be de-anonymized by matching their contact graph with the friendship graph of a social network. While we use data about users' friends to match individual user accounts, we do not leverage the social network graph as a whole. The structure of the social network graph is certainly a very powerful feature to match accounts. However, assuming that we have access to the whole social graph is not practical if we want to build a real-time and on-demand service that takes as input one account on a social network and searches for the matching accounts on other social networks. Nevertheless, combined with other features, these techniques might improve the matching accuracy. We leave this for future works.

[2] A census block may correspond to a city block in urban areas, however in rural areas where there are fewer roads, blocks may be limited by other features.

Researchers have also exploited other features for de-anonymizaiton. Wondracek et al. [193] identified users who visit a malicious web site by matching their browser history against *group memberships* in Facebook. Language models have been used for data de-anonymization. For example, Nanavati et al. [135] used language distribution at the n-gram level to de-anonymize reviews in an anonymous review process. Two other recent studies showed that text posted on blogs can be de-anonymized [136] and that online reviews could be matched across different sites [128]. We use the language models as an additional feature to match accounts.

2.4 Matching accounts across social networks

We now review the line of research more closely related to our work. Studies from different research communities have tackled the problem of matching accounts across social networks based on different features extracted from users' accounts. Most of these studies, however, were not evaluated at large scale on real social networks where the challenges reside. Consequently, even if most prior studies achieved good matching performance when evaluated at small scale, their methods will behave poorly when applied to today's social networks with millions to a billion of users. Moreover, no prior work analyzed the robustness of their schemes to impersonation attacks. In this section we classify matching schemes by the type of information they exploit.

2.4.1 Matching accounts using private user data

Balduzzi et al. [15] matched accounts on different social networks using the "Friend Finder" mechanism that social networks provide for users to find their friends using their email addresses. The study matched accounts with a list of 10 million *e-mail* addresses. As e-mail addresses satisfy all the four ACID properties[3] (Availability, Consistency, non-Impersonability and Discriminability), this is a simple yet powerful technique for matching accounts. In fact, this is what we use for obtaining our ground truth in Chapter 3. Many sites, however, view Friend Finder as leaking users' private data and have since limited the number of queries a user can make. Therefore, we focus on matching accounts using only public attributes.

2.4.2 Matching accounts using public user data

A number of previous studies have leveraged different profile features to match accounts without a systematic understanding of their ACID properties. Consequently, some of these studies use features with low availability and thus can only match a small fraction of accounts across social networks. Even the studies that used features with good ACID properties fail to achieve a reliable and scalable matching. In addition, most of the studies use ground truth users that willingly publish links to their accounts on different social networks. Our analysis in §4.1 reveals that such datasets have a higher attribute availability

[3]Remember that to guarantee a scalable and reliable matching an ideal features needs to satisfy all four properties.

and consistency, consequently the accuracy results of such schemes are overly optimistic. We split these studies according to the type of attributes they use.

Leveraging user profile attributes

A number of schemes leverage *user profile attributes* [6, 122, 134, 140, 145, 147, 148, 154, 184, 195] similar to the attributes we use for matching. Irani et al. [77] showed that one could potentially match accounts by searching for accounts where the screen names (the user loggin) were simple variations of the users's name. Perito et al. [148] showed that accounts can be matched by exploiting the similarity between their screen names. Liu et al. [118] used a very similar strategy to match users across different forums. Since the same screen name can correspond to different users, both Liu et al. [118] and Perito et al.'s [148] exploited the uniqueness of screen names to increase the accuracy of matching. Zafarani et al. [197, 198] proposed more sophisticated methods to detect if two screen names correspond to the same person by assuming that users follow the same practices in choosing their screen names across social networks. This assumptions is not valid on social networks, like Facebook and LinkedIn, that automatically generate screen names from the real names of users. The work by Motoyama et al. [134] is the first that showed the potential for matching accounts using *profiles attributes* (e.g., location, occupation, university, gender) aiming to assist users to find their friends when they join a new social network. The scheme considers attributes as bags of words and computes the similarity between two accounts as the number of common words between profile attributes. The bag of words technique is known to have a low recall because it cannot account for common entities that have slightly different names. For example, it cannot detect that the Bay Area and Berkeley are actually referring to the same area. Other studies [147, 154, 184] defined more specialized text based metrics for measuring the similarity between different fields and used classifiers to distinguish between matching and non-matching accounts. These techniques [118, 134, 147, 148, 154, 184, 197, 198] have only been evaluated using small-scale datasets. We simulate these approaches in §4.2, and we see that they are prone to give many false matches when used at scale.

A few recent studies performed account matching at scale [6, 122, 145]. These studies pointed out that account matching at large scale yields a large number of false matches. They do not, however, propose any mechanism to tackle the low reliability of their schemes. In contrast, we conduct a systematic analysis of the causes of such false matches and our scheme presented in Chapter 4 eliminates them.

Acquisti et al. studied the power of *face* recognition algorithms to match accounts in a dating site to Facebook accounts [6]. Even though the study showed that face recognition algorithms can match 10% of the accounts, it also acknowledged that face detection algorithms need to lower their false positives to be usable at large scale. Our matching scheme uses photo similarity to detect if two photos are the same but it does not use face recognition. Face recognition algorithms work very good when we have access to multiple photos to train the classifiers, however, unfortunately, in many social networks we only have access to one training instance, the profile photo. Adding face recognition could lead to improvements in matching accuracy, however, the challenge is to build face recognition algorithms that work well with only one training instance.

Leveraging user friends

To give more relevant information when searching people on the web, You et al. [195] proposed a scheme to link people's names to their social identities. Their scheme discriminates between candidate social identities by matching the *relational graph* of co-occurrences of names, extracted from EntityCube[4], to the *friends graph*. Similar with our matching scheme, they propose a notion of confidence in the match. The algorithm, however, only works for people with notable web presence that have an entry in the relational graph thus the scalability is limited. In contrast, our scheme in Chapter 4 finds matching accounts for arbitrary users.

Labitzke et al. [110] showed that the overlap between friends can be used to match accounts on social networks. Our results confirm that friendship overlap is an important feature for detecting matching accounts and we use it to complement the other features extracted from user accounts. Similar with efforts in de-anonymizing graphs, Korula and Lattanzi mathematically formalize the problem of matching accounts across social networks and propose an algorithm that uses the friendship graph to match accounts [95]. As we discussed in §2.1.2, researchers have propose algorithms to jointly predict links and infer attributes in social networks. Bartunov et al. [16] took this approach further and proposed a *joint link-attribute* algorithm to match two social networks. The structure of the networks can be used to infer missing attributes; while profile attributes can be used to link matching nodes. The authors, however, only evaluated the algorithm on one-hop neighbors. As we already discussed, the structure of the social graph is certainly a powerful feature and, combined with other features, it might improve the overall matching accuracy. Nevertheless, we leave this for future work.

Leveraging user activity traces

Other schemes use attributes extracted from *user activities* (i.e., the content users generate instead of attributes of the profile) [76, 128]. These schemes reveal how even innocuous activities of users can help identify a user across social networks. Mishari et al. [128] showed that the authors of online reviews could be linked across different sites by exploiting the *writing style* of the authors. Finally, Iofciu et al. [76] used *tags* to match accounts between Delicious and Flickr. In Chapter 5, we also exploit information from user activities to match accounts. We show that the locations from where users post and the time when users post are much better at matching accounts than the writing style when considering social networks where users do not post long sentences such as Twitter and Flickr.

To conclude, there are two main points to consider when building and evaluating a scalable and reliable matching scheme. First, for a scheme to be scalable it has to disambiguate the matching account out of a list of similar-looking accounts. Most of the current schemes build (and evaluate) their classifiers using data that contains half matching accounts and half non-matching accounts chosen at random. Non-matching accounts chosen at random are likely very different, and thus very easy to identify as non-matching. Even if such classifiers show good results their performance drops drastically, when they have to disambiguate between similar looking accounts (they will return a high number of false positives).

[4] http://entitycube.research.microsoft.com/

In Chapter 4, we show that, to obtain an accurate classifier, one has to train the classifier with pairs of non-matching accounts that are not easy to identify. Second, from a reliability point of view, a scheme has to differentiate between cases where it is confident that the most similar account is the matching account and cases where it is not confident. We propose, in Chapter 4, a method to evaluate this confidence in the match.

2.5 Security threats in social networks

In this thesis we want to identify impersonation attacks in social networks. Since attackers are actually creating fake accounts to impersonate people, we review in this section the research done in detecting fake accounts online. We then focus on identity theft attacks.

Today, most online services including social networking services, allow users to create accounts for free or at very little cost. All these services are based on a weak notion of identity where it is hard to map an identity to a real entity. In such a setup, it is a challenge to distinguish trustworthy from untrustworthy users (e.g., spammers). This task is especially hard because an attacker can exploit weak identities to create multiple fake identities and manipulate the functioning of the system. For example, an attacker can create multiple fake accounts to manipulate content ratings or to send spam. It is well known that spam and service abuse is a standing problem [25]. Below we discuss three broad existing approaches to detect fake identities.

Behavioral profiling

Is one of the most widely used approach today where given some ground-truth information about known trusted and untrusted users, the service provider can build a behavioral profile for each class of users. A behavioral profile for an account approximately characterizes the activities of an account within that service and may also characterize the content generated or consumed by the account. Activity within a service can include, for example, the act of sending messages to other identities, viewing pages, or rating a particular piece of content. Once we have constructed such a behavioral profile, we can use it to classify unknown accounts as being either trusted or untrusted. The typical way to build behavioral profiles of accounts is through machine learning approaches. While researchers have proposed several techniques [17, 200], the industry has also adopted this strategy. Facebook has built a tool called the Immune System [169] to identify attackers. All techniques used so far to detect fake accounts rely solely on information available about a single user identity. We will see in Chapter 6 that such techniques fail to detect the fake accounts created by impersonators. A better approach to detect such accounts is to use features that characterize pairs of accounts (the impersonating and the real account) rather than features of that characterize only the fake account.

Leveraging trust between identities

To assess trustworthiness of identities, one type of information typically available on social networks is trust relationship *between* identities. This relationship can be explicitly defined when two identities mutually trust each other and form a friendship relationship, or it could be implicitly estimated based on the amount of interaction between two identities (e.g., identities that talk to each other frequently or tag each other in photos might trust each other more). We can build a trust network between identities using this information, where each edge in the network represents trust. Researchers have proposed a variety of schemes such as SybilGuard [196] and SybilRank [25] that analyze such trust networks to assess trustworthiness of identities and thus identify Sybil attackers [181,183,196]. The key assumption here is that an attacker cannot establish an arbitrary number of trust edges with honest or good users in the network. It should be noted that such a trust network is entirely built based on information available within a single site. This assumption breaks when we have to deal with impersonating accounts as for them it is much easier to link to good users.

Crowdsourcing misbehavior detection

A third approach to identify suspicious identities is to crowdsource this task. Many social networking services rely on end users to report suspicious profiles or actions taken by users (e.g., message postings) to the service provider. End users make the judgement to report suspicious actions or profiles solely based on information available to them within that site. Furthermore, a recent study that examined crowdsourced misbehavior detection by end users in a social network reported that such reports tend to be false alarms most of the time [25].

Another variation of this approach is to crowdsource this task to experts who are familiar with identifying suspicious profiles or actions. Social networking services typically have a tiered approach where suspicious profiles reported by end users are further verified by a group of experts before taking a decision to suspend the account or show Captchas to those suspicious users [25]. In fact, researchers recently explored the possibility of using online crowdsourcing services such as Amazon Mechanical Turk (AMT) to crowdsource the task of detecting sybil identities in a social network [185]. In Chapter 6, we will see that AMT workers have a very poor accuracy in detecting impersonating accounts because they are extremely similar to real accounts.

Identity theft

Traditional methods to detect fake accounts fail to detect impersonators. Recently, researchers have become interested in detecting identity theft attacks. Identity theft attacks are also known as profile cloning or identity cloning. The attacker creates a fake account that clones the information of a victim account. The attacker then uses the fake account to send friend requests to the victim's friends. The goal of such attack is to launch phishing attacks or to harvest sensitive information about the victim. Bilge et al. [22] demonstrated the feasibility of automated profile cloning attacks both inside the same social network

and across different social networks. We consider identify theft attacks as a subclass of impersonation attacks. By impersonation attacks, we mean all the attacks that create fake accounts impersonateing another person, whatever the reason. The fake accounts can be either used to acquire sensitive information about the victim, to spread wrong rumors about a celebrity, or just to avoid fake account detection systems.

A few studies made some initial investigation about detecting profile cloning [84,93]. They hinted at the fact that cloned profiles can be identified by searching for profiles with similar attribute values. None of the prior works, however, has done a systematic study of how to search for profiles with similar attributes and what is the accuracy of the approach for detecting cloned profiles in real social networks. To protect against friend requests coming from cloned profiles, He et al. [70] proposed three strategies: (1) use a challenge – upon receiving a friend request, the users can chat to confirm their identities; (2) use a single account such as OpenID to connect to multiple social networks; (3) verify the number of mutual friends. The detection of impersonation attacks shares common techniques with the detection of identity theft attacks. However, we do not stop at detecting a group of similar looking accounts with a victim account, we propose an algorithm that detects with high accuracy when two accounts correspond to the same person or not. Furthermore, our proposed method can discriminate between pairs of accounts that are avatars (multiple honest accounts maintained by the same person inside a social network) and pairs of accounts where one account corresponds to an impersonator and one account corresponds to an honest user.

Chapter 3

Account Matching Framework

In this chapter, we set the general framework for matching accounts. We define the problem of matching accounts across social networks (§3.1), and we then list the constraints imposed by our scenario (§3.2). One important design consideration for our matching problem is to build reliable and scalable matching schemes that perform well even when applied in *large-scale* networks with hundreds of millions of users. For this we identify four crucial properties – *Availability, Consistency, non-Impersonability, Discriminability* (ACID) – that features need to satisfy to enable reliable and scalable account matching (§3.3). This chapter also presents our ground truth data in §3.4 and our strategies to evaluate the matching schemes in §3.5. Chapters 4 and 5 will go into the technical details of building the matching schemes.

3.1 The matching problem: definition

Given an account in one large social network, $a \in SN_1$, the account matching scheme attempts to find its *matching account* in another large social network, $\hat{a} \in SN_2$, if one exists. We consider two accounts to be *matching* if they are managed by the same person, i.e., $user(a) = user(\hat{a})$. We assume that there is at most one matching account to a user in any given social network, an assumption that is rooted in the usage policies of many social networks like Facebook [142].

To match accounts across social networks, we will to exploit all kinds of information users *publicly* provide in their profiles such as their real names, screen names, location, profile photos, friends as well as what they post. We call the information users provide about themselves in their profiles (e.g., real names, screen names, location, profile photo, bio) the *public attributes* and everything that is related with their activity on a social networks (e.g., what they post, what they like) the *user activity*.

An alternative could be to match accounts based on private information in user profiles. For example, an attacker could gain illegal access to users' private information when users inadvertently access a malicious site or application. Such approaches raise some serious privacy and security concerns. However, gaining illegal access to users accounts requires users to visit some malicious sites. These techniques are usually specific to a single social network and cannot be reused on other social networks. As soon as the social network

owners discover the security vulnerability that is exploited by the attacker they fix it. Therefore, this thesis only considers public information. Public information is essential for the purpose of the social network.

We tackle two specific variations of the problem:

- Is it possible to match accounts across different social networks only using the information users provide in their public profiles such as their real name, screen name, profile photo, friends and location?
  $\longrightarrow$ **Chapter 4**

- If we ignore all profile attributes, can we still match accounts by exploiting the user activity (e.g. from where users post, at what time users post and what are the specificities of their writing style)?
  $\longrightarrow$ **Chapter 5**

The matching scheme that exploits public attributes will be mainly able to match accounts of users that maintain the same persona across different social networks and do not necessarily want to hide. This matching scheme can be used in the future as a building block for applications that exploit cross-site information. On the other hand, the matching scheme that exploits the user activity can be used to target users that want to hide and maintain different personas across social networks. For example, this matching scheme can be part of a suite that helps users measure their online footprint to better understand their privacy risks.

3.2 The matching problem: challenges

There are several challenges to match accounts in todays social networks that make the problem both interesting and difficult:

1. **Information users provide is noisy.** First, users might not provide the same kind of information on different social networks. For example one user might choose to show his location on Facebook while he keeps his location private on Flickr. Second, even if the users provide the same kind of information across different social networks, they might not be consistent. For example, one user who moved, might update his location only on Facebook while leaving his old location on Twitter.

2. **At large scale, the chances that two users have similar profiles grows.** While in a small dataset there might be just one Jennifer Clark in San Francisco, if we consider all the accounts in a social network, there might be tens or hundreds of Jennifer Clarks.

3. **Attackers create accounts attempting to impersonate other users.** If a matching scheme does not handle such cases, it risks matching the account of a real person to the account of an impersonator.

4. **Social networks have restricted APIs to access/query user accounts.** If we want to build a matching scheme that works in practice, we have to take these restrictions into account.

5. **Data imbalance:** There are many more possible non-matching accounts than matching accounts. If not handled correctly, data imbalance creates problems for both training and evaluating classifiers.

3.3 The matching problem: scalability and reliability

A matching scheme that aims to match accounts over large real-world social networks like Facebook and Twitter must be *reliable* and *scalable*. By scalable, we refer to the challenge of finding a single account (say on Facebook) that matches a given account (say on Twitter) from potentially hundreds of millions of user accounts. By reliable, we refer to the challenge of matching accounts with a very low error rate, i.e., very few false positives. It is crucial for an account matching scheme to be reliable, as in most application scenarios (e.g., profiling a prospective employee) incorrect matches could be worse than simply failure to match. Thus, it is better for a matching scheme to return nothing rather then to return an account when it is not highly confident that the account is a true match. The difficulty of achieving highly reliable matching gets magnified with the scale (size) of the social network. In a social network with hundreds of millions of accounts, there is a non-trivial chance that there exist multiple accounts with very similar features leading to false matches.

3.3.1 The ACID test for scalable and reliable matching

Any scheme for matching accounts works by matching *features* of the accounts on different sites. The reliability and scalability of the scheme depends crucially on the features selected for matching the accounts. To achieve reliability and scalability for matching accounts, we identified four key properties that an ideal feature should have:

- **Availability:** To scale account matching, the selected features should be available for a large fraction of user accounts across the different sites. For example, if only 5% of users provided information about their 'age' across two sites, then 'age' has limited utility in matching accounts.
- **Consistency:** It is crucial that the selected feature is consistent across different matching accounts, i.e., that users provide the same feature values across the different accounts they manage. If users provided different values for their "name" across the different sites, then "name" would not be useful in matching accounts.
- **non-Impersonability:** If a feature can be easily impersonated, i.e., faked, then attackers can compromise the reliability of the matching by creating fake accounts that appear to be matching with the victim's accounts on other sites. Some public features like 'name' and 'profile photo' are easier to copy and impersonate than others such as 'friends'.
- **Discriminability:** A highly discriminating feature would have a unique and different value for each account, while a less discriminating feature would have similar values for many accounts. For example, 'name' is likely to be more discriminating than 'location' and 'gender'. The more discriminating a feature, the lower the chances

of a false match with other non-matching accounts and the higher the reliability of matching.

To enable reliable correlation, the selected feature should be highly *Available, Consistent, non-Impersonable, and Discriminable (ACID)*. If a feature satisfies these four ACID properties, then matching accounts using the feature is trivial. Unfortunately, in practice, we will see that no feature exhibits all four ACID properties. In Chapter 4, we will study the ACID properties of attributes from public profiles and in Chapter 5 we will study the ACID properties of user activity. Different account features satisfy the properties to different extents and the key challenge in reliable account matching schemes lies in designing matching algorithms that can leverage multiple features with imperfect ACID properties to achieve high reliability and scalability.

3.4 Ground truth data

Our analysis requires *ground truth* data of matching accounts, i.e., accounts belonging to the same user, across different popular social networks. To evaluate our techniques in the most challenging environments, we gather ground truth data on seven out of the most popular and largest social networks today: Facebook, Twitter, Google+, LinkedIn, Flickr, Yelp, and MySpace. Gathering ground truth data of matching accounts spanning multiple social networks is difficult. Below we describe two 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 10 million email addresses collected by colleagues for an earlier study analyzing spam email.[1] 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 an account ID impossible. Hence, we were only able to collect the email-to-account ID matching for Twitter and Flickr. In addition, we obtained from colleagues pairs of matching accounts among Twitter, Facebook, and LinkedIn, which they collected for an earlier study using the same technique before the restrictions were introduced in the query APIs [15].

Table 3.1 summarizes the number of matching accounts we obtained using the Friend Finder mechanism (Dataset 1). Since spammers target the public at large, we believe that this list of emails catches a representative set of users, but we are limited to only five social networks. Therefore we complement Dataset 1 with a second, orthogonal set that contains more social networks. We exploit the fact that Google+ allows users to explicitly list accounts they have on *other* social networks on their Google+ profile pages. We randomly crawled about 3 million Google+ accounts and arrived at the ground truth set summarized in column Dataset 2 of Table 3.1, which reports the number of matching accounts for different combinations of social networks.[2] Google+ users, who voluntarily reveal their accounts on other sites, might not represent users in general but we can use

[1] The local IRB approved the collection.

[2] Google+ proves easy to crawl as it provides an initial starting point in the form of a comprehensive directory of all accounts, and does not block crawlers.

Table 3.1: Number of ground truth matching accounts obtained with Friend Finder (Dataset 1) and Google+ (Dataset 2) for different combinations of social networks.

	Dataset 1	Dataset 2
Twitter - Facebook	4,182	76,332
LinkedIn - Facebook	2,561	20,145
Twitter - Flickr	18,953	35,208
LinkedIn - Twitter	2,515	20,439
Twitter - Yelp	1,889	5,130
Flickr - Yelp	1,119	2,899
LinkedIn - Flickr	-	8,503
Twitter - Google+	-	205,709
Twitter - Myspace	-	9,015
Facebook - Google+	-	164,333
Facebook - Myspace	-	9,610
LinkedIn - Google+	-	32,827
LinkedIn - Myspace	-	2,283
Myspace - Google+	-	36,440
Yelp - Google+	-	5,620
Twitter - Facebook - Google+	-	76,332
Twitter - Facebook - Myspace	-	4,207
Twitter - Google+ - Myspace	-	9,015
Facebook - Google+ - Myspace	-	9,610
Twitter - Flickr - Yelp	559	2,753

these users to extend our study to other social networks and to a larger dataset. Note that most of previous works used this kind of users as ground truth. In the rest of the manuscript, by default, we show the values for accounts in Dataset 1 because they are more representative. Generally, the matching accuracy for users in Dataset 2 is higher than users in Dataset 1. We will pinpoint throughout the manuscript some of the most interesting differences.

3.5 Evaluation method

In this section we present the evaluation methodology that we will use in Chapter 4 and Chapter 5. We evaluate our matching schemes from three complementary perspectives. First, we evaluate the accuracy over a small dataset. This evaluation method was used by most of previous studies. However, we believe that the evaluation over a small dataset is a misleading indicator of the accuracy over entire social networks. Thus, we also evaluate the matching schemes at scale over entire social networks. Finally, we evaluate the matching schemes against human workers.

3.5.1 Evaluation metrics

We say that a matching scheme outputs a *true match* when it correctly identifies the matching account and the scheme outputs a *false match* otherwise. To measure the accuracy of a scheme we use two pairs of metrics: true and false positive rates, and precision and recall. The recall and the true positive rate represent the proportion of true matches, i.e., the proportion of matching accounts detected by the scheme out of all the tested matching accounts. The false positive rate is the proportion of non-matching accounts the scheme mistakenly identifies as matching accounts out of all the non-matching accounts tested. The precision is the number of true matches divided by the sum of true and false matches the scheme returns. An accurate matching scheme has a high true positive rate and a low false positive rate, or else, a high recall and precision. Even if the two pairs of metrics are similar, when dealing with unbalanced datasets (i.e., the number of non-matching accounts is much higher than the number of matching accounts), the precision/recall is a much better indicator of the accuracy of a matching scheme than the true/false positive rate as we will see later. We present both pairs of metrics to allow comparison with previous work.

3.5.2 Evaluation over a small dataset

The evaluations over a small dataset has been the typical evaluation method for most of the previous works that attempted to match accounts across social networks [118, 134, 147, 148, 154, 184, 197, 198]. Here we sample data to simulate this approach. We select n random accounts on SN_1 from DATASET 1 for which we know the corresponding n accounts on SN_2. We evaluate the accuracy of the matching scheme to identify the n pairs with matching accounts out of the n^2 possible combinations of pairs.

3.5.3 Evaluation at scale

The evaluation of matching schemes at scale requires access to all the accounts in a social network, i.e., hundreds of millions of accounts, on large social networks like Facebook, Twitter, or Google+. Obtaining such large datasets is very difficult. Instead, we resort to an optimization where, for a given account in SN_1, we preselect a *candidate set* of accounts that are most similar to the given account on SN_2 and evaluate the efficacy of our schemes over the candidate set. Our assumption is that, if a matching scheme is able to disambiguate the matching account out of a small set of very similar accounts, it will most likely be able to disambiguate the matching account out of a larger set of less similar accounts. Evaluating matching schemes over the candidate set rather than the set of all accounts leads to an under-estimation of both the true positive rate (when the matching account is missed from the candidate set) and the false positive rate (when one or more non-matching accounts that are similar to the given account are missed). Our insight is that such a candidate set of accounts can be generated by leveraging the search APIs provided by social networks like Facebook.

We build next two separate datasets for the evaluation of matching schemes using user profiles (Chapter 4) and for the evaluation of matching schemes using user activities (Chap-

ter 5).

Large scale dataset for matching schemes based on user profiles: We evaluate the accuracy of matching schemes to find the corresponding Facebook account for a given Twitter account. The matching method as well as the evaluation are not specific to these social networks.

To build a test dataset to evaluate the scheme at large scale we generate, for each Twitter account, a, the candidate set of the most similar Facebook accounts, $C(a)$. A *perfect candidate set* would include the Facebook matching account as well as all the Facebook accounts that are similar with the target Twitter account, a. The Facebook search API allows to search for people by name, however it is not possible to search by other attributes.[3] Given that 84% of Twitter-Facebook matching accounts have consistent real names and 40% consistent screen names we can build a comprehensive candidate set by exploiting the Facebook search API to find Facebook accounts with the same or similar real name or screen name as the Twitter account. The Facebook search API mostly returns accounts with the same name and only very few accounts with similar names. To get more accounts with similar names, we apply a number of heuristics to the real name and the screen name so that we can find accounts with simple variations of the name (for example, if the real name is Franklin Delano Roosevelt, we will search for Franklin Delano, Delano Roosevelt, and Franklin Roosevelt; if there is only one name as FranklinRoosevelt, we will try to split it). Finally, we test whether there is a Facebook account with the exact same screen name as the Twitter account. When the real name of the Twitter account is a single word, we also check whether it is a screen name in Facebook.

We build Facebook candidate sets $C(a)$ for 1,000 Twitter accounts from Dataset 1 (the same we choose for the small dataset). The averages size of accounts in $C(a)$ is 320. The Facebook matching account is included in the candidate set for 70% Twitter accounts.[4] This percentage is lower than the percentage of Facebook-Twitter matching accounts with consistent real names, 84%, because Facebook mostly returns accounts with the same name rather than similar names. This limitation is out of our control and has the following implications on our system design and evaluation method: 1) if we build a real-time service that finds the Facebook matching account for a given Twitter account, 70% will be an upper bound on the percentage of accounts we can match regardless of the accuracy of the matching scheme; 2) the true positive rate we estimate when matching at scale in the rest of the paper is underestimated by a percentage up to 30%.

Large scale dataset for matching schemes based on user activities: For our case study, we evaluate matching schemes to match Flickr or Yelp accounts to Twitter accounts.

As for the previous dataset, we have to generate, for each account $a \in SN_1$, a candidate set of the most similar accounts in SN_2, $C(a)$. However, the dataset used to evaluate matching scheme at scale using user profiles is not good for evaluating matching schemes based on

[3] Facebook actually allows to search for people in a particular location but the number of results returned is limited to a few hundreds.

[4] This percentage is 82% for accounts in Dataset 2.

user activities for two reasons. First, it mainly contains users that have consistent real names across social networks, while we focus here on matching the accounts of users that do not have consistent names. Second, since we use features extracted from user activities such as the location and time of users' posts and the users' writing style, the dataset previously generated will not contain the accounts in SN_2 that are the most similar with the target account a. The key to generate comprehensive candidate sets is to select the accounts in SN_2 based on the features considered for the matching. For example, if we aim to match accounts by exploiting the location from of posts, we can assume that users who post regularly within a certain region will most likely live there, and thus their posts on other sites will originate from there as well. For each account a in SN_1 we can build the candidate set of accounts $C(a)$ in SN_2 by extracting all users of SN_2 who have posted from that region where a lives.

To build the dataset for evaluating the matching schemes at large scale to match Flickr to Twitter accounts, we could, for each Flickr account, determine the dominant region (the region where a user posts the most) and then collect all the Twitter accounts that have at least one post in this region. Such approach requires us to collect *separate* sets $C(a)$ for each a, which will likely lead to gathering users from all around the world. Since there is no Twitter API call that gives the list of users who posted in a certain region (as there is a call to get all the users with a given name, which we used for the previous large scale dataset) we resort to an optimization. We limit our evaluation to match users living in five urban areas in the US (San Francisco, San Diego, New York, Chicago, and Los Angeles), which allows us to use a *single* set C^{area} for all the ground truth accounts in each of these areas. We define the subset of ground truth users, $GT^{\text{area}} \subset$ Dataset 1, as those who have more posts inside the respective area than outside it. Table 3.2 shows the number of matching ground truth accounts in Dataset 1 in the five areas previously mentioned. We filtered out all accounts that have no posts or no locations attached (considering addresses for Yelp, and geotags otherwise).

For the evaluation, we focus on matching Flickr or Yelp accounts to their corresponding Twitter accounts. We obtain the candidate sets C^{area} by crawling Twitter for users from each of the five areas. We use the Streaming API[5] to collect in real-time all the tweets tagged with a location in one of the five areas between August and November of 2012. We then extract all users that have at least one tweet in this collection. For the San Francisco area, the Twitter matching account is included in the candidate set C^{SF} for 75% of the Flickr accounts in GT^{SF} (a set of users taken from the San Francisco for one year achieves 95% coverage). This shows that our method to gather a comprehensive list of accounts that are similar with a given target account based on location works well and has a comparable coverage with what we previously had when using real names. Table 3.3 presents the number of users we collected for each area.

We evaluate the accuracy of the matching scheme to detect the Twitter matching account $\hat{a}$ of a given Yelp or Flickr account, a in GT^{area}, out of the candidate set, C^{area}. We focus on matching Yelp and Flickr to Twitter accounts, but the same methods can be applied to other social networks. It is also possible to gather candidate sets in Flickr and Yelp. The Flickr API allows to search all photos with geotags in a certain region (defined

[5]While the Streaming API generally returns only a sample of tweets, limiting a query to a region the size of, e.g., the San Francisco area seems to indeed return the complete set.

Table 3.2: Number of Yelp and Flickr users in DATASET 1 (for which we know their matching Twitter account) with more posts inside a given area than outside it, for 5 selected areas.

	Number of users in				
	SF	**SD**	**NY**	**C**	**LA**
FLICKR - TWITTER	474	152	427	236	284
YELP - TWITTER	160	45	106	50	117

Table 3.3: Number of users in the C^{area} for five selected areas. † Users with at least one post inside a given area; users may belong to multiple areas.

	C^{area} in				
	SF†	**SD†**	**NY†**	**C†**	**LA†**
Twitter	75,747	35,068	89,219	54,774	77,402

as a latitude/longitude bounding box), which makes the collection of the candidate set straightforward. For Yelp, we can retrieve a list of all restaurants in each of the areas and then extract users that reviewed one of them.

We build the candidate set based on the feature used for matching. In principe, then, to match accounts based on timing, we can select $C(a)$ as those accounts for which we find a temporal overlap with posts from $a \in SN_1$. Instead, however, we also limit the language and timestamp analyses to the datasets gathered using location so that we can evaluate the combination of multiple features on the same set of users.

3.5.4 Evaluation against human workers

The evaluation over a small dataset and at scale measure the accuracy of the matching schemes against the ground truth, which is the standard way to evaluate such schemes. However, we can obtain a more interesting and different perspective by evaluating their accuracy against human workers. The practice of employing human workers to do simple tasks such as detecting fake or spam accounts is becoming more and more popular and is used by companies like Facebook and Twitter. Humans are particularly good at such tasks and are often much better than machines [186]. We can consider human workers as a gold standard for matching accounts because they are very good at identifying persons. Thus, in this evaluation, we focus on understanding: (1) how well humans can detect matching accounts; and (2) which are the cases where the matching scheme performs worst than humans and where it performs better.

3.6 Summary

In this chapter, we setup the matching account framework: we defined the problem of matching accounts, we described the challenges in matching accounts in todays' social networks and we described our ground truth data and evaluation method. We proposed a set of

four properties to evaluate the quality of features used for the matching in order to achieve reliable and scalable matching schemes – *Availability, Consistency, non-Impersonability* and *Discriminability*. We also proposed a novel evaluation method that allows to put in perspective the accuracy of matching schemes by comparing results against humans as well as comparing the results at small scale vs. large scale. Chapter 4 and Chapter 5 will use this framework to build and evaluate matching schemes.

Chapter 4

Matching Accounts using Public Profile Attributes

In this chapter we investigate ways to match accounts across social networks by exploiting the information users provide in their public profiles. One important design consideration for our matching problem is to build reliable and scalable matching schemes that perform well even when applied in *large-scale* networks with hundreds of millions of users.

Given an account in one social network, say a, we leverage multiple public attributes of the account (specifically, real name, screen name, location, profile photo, and friends) to find its matching account in another social network. We first study the ACID properties of these features (see §4.1). Our analysis reveals that they satisfy the ACID requirements to different extents, but none of the features alone satisfies all requirements. We then investigate the challenges of leveraging multiple public attributes to build a reliable matching scheme.

We present a scheme that matches accounts using a binary classifier, called the Linker, trained to distinguish between matching accounts and *random* non-matching accounts (§4.2). This training approach is similar to what prior work has done. Unfortunately, we show that the Linker has limited scalability. We then propose a new matching scheme (§4.3) that proceeds in three steps to work at scale. First, we filter the majority of accounts that are clearly different from a. Second, we build a classifier, called Disambiguator, that is able to distinguish between matching accounts and other *similar* non-matching accounts. The Disambiguator is solving a harder problem than the Linker, because it is trying to distinguish among accounts that are highly similar. Third, based on the assumption that there is only one matching account on the second social network, we process the results from the Disambiguator to ensure that the matching scheme is reliable to impersonators and to achieve a better precision. The key distinguishing features of this three-step matching scheme are its *reliability* and *scalability*. This scheme can accurately estimate its confidence in the match and abstains from producing an output when it is uncertain. Contrary to the one-step matching scheme, the three-step matching scheme has good accuracy at scale and is therefore usable in practice.

We evaluate the performance of the three-step matching scheme on two of the largest real-world social networks: Twitter and Facebook. Our results show that the three-step matching scheme can correctly detect 21% of matching accounts with a 98% precision (or

alternatively, a 30% recall for a 92% precision). These numbers reflect the real fraction of accounts we can match, at scale, without shortcuts and considering all the limitations imposed by the query interfaces of social networks. We further confirm the inherent difficulty in reliably matching accounts by comparing the performance of our automated matching scheme with that of human Amazon Mechanical Turk (AMT) workers. Under similar conditions, AMT workers are able to match 25% of the accounts with a 98% precision. Interestingly, when we evaluated the matching scheme over small subsets of Facebook accounts containing a few thousands accounts as in previous work, it detects 83% with 89% precision. Our analysis highlights the true costs of achieving high reliability in account matching schemes at large-scale.

It is certainly possible to improve detection by using more account features (beyond the public attributes we considered) and more sophisticated matching techniques (e.g., better face matching techniques) in the account matching schemes. We consider, however, that our main contributions consist in identifying the challenges in achieving a scalable and reliable matching and proposing a set of fundamental design considerations to meet these challenges.

4.1 ACID test of public attributes

In this section we analyze the extent to which public attributes satisfy the ACID test requirements. We perform this analysis for six large scale social networks to understand the potential of matching accounts across different sites.

4.1.1 Attribute availability

We begin by examining the availability of public attributes across the six social networks in our ground truth. We focus on five attributes that are supported by all sites: screen name (aka. username, nickname – name that appears in the URL of the profile), real name, location, profile photo, and friends. The availability of these attributes depends on whether users choose to provide information about them. In some sites, users can choose to make some of these attributes private, which also affects their availability.

For all users in our ground truth datasets, we collected their publicly visible attributes from their accounts on the six social networks. Table 4.1 shows the breakdown of attribute availability per social network. We highlight three implications of attribute availability results for matching accounts.

First, we find that some attributes like screen name and real name are considerably more available than other attributes like location or friends. The more available attributes are, the more useful they are in matching a larger number of accounts. However, the availability of the less available attributes is not negligible – for example, location and friends are available for more than 50% of accounts in Twitter and Facebook. In later sections, we show how these partially available attributes can be used in conjunction with highly available attributes to improve the accuracy of matching accounts.

Table 4.1: Availability of attributes for a single social network and for pairs of social networks for Dataset 1 and Dataset 2.
Legend: Tw = Twitter, Fb = Facebook, G+ = Google+, M = Myspace, Fl = Flickr, Lnk = LinkedIn; $X\%/Y\%$ = availability in the Dataset 2 availability in Dataset 1. Whenever there is only one value it is for Dataset 2.

	Real Name	**Profile Photo**	**Location**	**Friends**
Twitter	100%/100%	96%/69%	80%/54%	91%/86%
Facebook	100%/100%	99%/98%	-	58%/60%
Google+	100%	80%	72%	-
Myspace	20%	98%	26%	-
Flickr	76%/30%	95%/29%	53%/11%	68%/40%
LinkedIn	100%/100%	54%/57%	98%/99%	-
Fl - Tw	76%/31%	88%/24%	46%/8%	72%/35%
Fb - Lnk	100%/100%	53%/56%	-	-
Fb- Tw	100%/100%	94%/69%	-	38%/48%
Lnk- Tw	100%/100%	52%/44%	84%/54%	-
Fb - G+	100%	91%	-	-
Fb - Fl	77%	88%	-	20%
Fb - M	17%	89%	-	-
Tw - G+	100%	90%	55%	-
Tw - M	16%	85%	23%	-
Lnk - Fl	79%	47%	56%	-
Lnk - G+	100%	52%	74%	-
Lnk - M	16%	40%	37%	-
Fl - G+	76%	87%	39%	-
Fl - M	12%	81%	21%	-
G+ - M	20%	75%	22%	-

Second, 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 here, it highlights the additional information one could learn about a user by matching her accounts on different sites. Our observation also hints at the potential for iteratively matching accounts across different sites, i.e., building a more complete profile of a user by matching her accounts across some sites and useing the resulting profile with more attributes to match her accounts on other sites.

Third, when we compare the availability using Dataset 1 and Dataset 2, we observe that the availability of attributes for accounts in the Dataset 1 is much lower than the availability for accounts in the Dataset 2. This has implications on the representativeness of users in Dataset 2, and consequently the representativeness of the results in previous works [122, 136, 145, 148, 198].

4.1.2 Attribute consistency

We now study the extent to which users provide similar attribute values for their accounts 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 the matching accounts of such users by matching their attributes.

Amongst users who do not deliberately set different attribute values on different sites, many users may not spend the effort needed to set their attributes to exactly the same values across all sites. For example, a user might specify their work place as International Business Machines on one site and International Business Machines Corporation on another site. To profile for such scenarios where the attribute matching is not exact, we first propose metrics for estimating similarity in attribute values for matching accounts and determine thresholds for determining when two values are sufficiently *similar*.

Similarity metrics for attributes

We borrow a set of standard metrics from prior works in security, information retrieval, and vision communities to compute similarity between values of the five public attributes we study.

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 [36,148]. 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 profile photo can come in different formats on multiple social networks. 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 [3] 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 [120].

We use two different algorithms because 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. Thus, having two algorithms makes the detection more robust.

Location similarity: For all accounts, we have the textual representations of the profiles' location such as the name of a city. However, as social networks use different formats for this information, we cannot just perform a textual comparison. Instead, we convert the location to latitude/longitude coordinates by submitting them to the Bing API [1]. We

then compute the similarity score 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 accounts. We consider that two accounts have a common friend if there is an account 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 our approach recursively by taking other features beside screen name and real name into account. We will see later, however, that given two small list of accounts on different social networks, real names and screen names are already very good alone to identify matching accounts. Thus, we limit ourselves to the simpler approach in this thesis.

Selecting 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. Our goal is to select thresholds for attribute similarity beyond which the chance of two values being consistent is quite high (say larger than 90%).

To this end, we gathered ground truth data by asking humans (AMT users) to evaluate whether pairs of attribute values are consistent or not. We randomly selected 100 pairs each of matching and non-matching Twitter and Facebook accounts from Dataset 1 and asked AMT users to compare their corresponding attribute values for consistency. We followed the guidelines [9] to ensure good quality results from AMT workers. For each pair of attribute values, workers had to choose between 3 answers: (i) *Yes, they match*; (ii) *No, they do not match*; and (iii) *Cannot say, the information is not available.* For each attribute value pair, we ask the opinion of three different AMT workers and consider that the values are consistent if there is a majority agreement that these values match.

We leverage the AMT experiments 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 higher similarities. We only use thresholds to study the consistency and not to match accounts. The matching algorithm uses the similarity scores between attributes directly.

Attribute consistency in matching accounts

We now use the similarity metrics and thresholds defined in the previous section to evaluate whether attribute values in matching accounts are consistent.

Table 4.2 shows how likely users are to provide consistent values for an attribute in a pair of social networks. We highlight three implications of our consistency analysis for matching accounts.

First, we find that large fraction of users provide similar *real names* across the different social networks. Put differently, most users are not attempting to maintain distinct personas

Table 4.2: Consistency of attributes for pairs of social networks for DATASET 1; † consistency extracted from DATASET 2.

	Screen Name	**Real Name**	**Location**	**Profile Photo**
Fb - Tw	40%	84%	22%	19%
Fb - Lnk	71%	97%	17%	23%
Tw - Fl	40%	84%	32%	22%
Tw - Lnk	36%	83%	28%	31%
Fb - Tw†	60%	88%	77%	26%
Fb - Lnk†	80%	98%	81%	23%
Tw - Fl†	56%	89%	71%	31%
Tw - Lnk†	57%	89%	70%	36%
Fb - Fl †	48%	94%	30%	20%
Fb - G+ †	59%	94%	65%	20%
Fb - M †	46%	77%	43%	13%
Tw - G †	50%	85%	62%	31%
Tw - Lnk †	57%	89%	29%	36%
Lnk - Fl †	48%	95%	33%	29%
Lnk - G †	65%	97%	67%	35%
Lnk - M †	54%	76%	56%	23%
Fl - G †	42%	94%	63%	31%
Fl - M †	43%	78%	49%	27%
G - M †	44%	70%	51%	20%

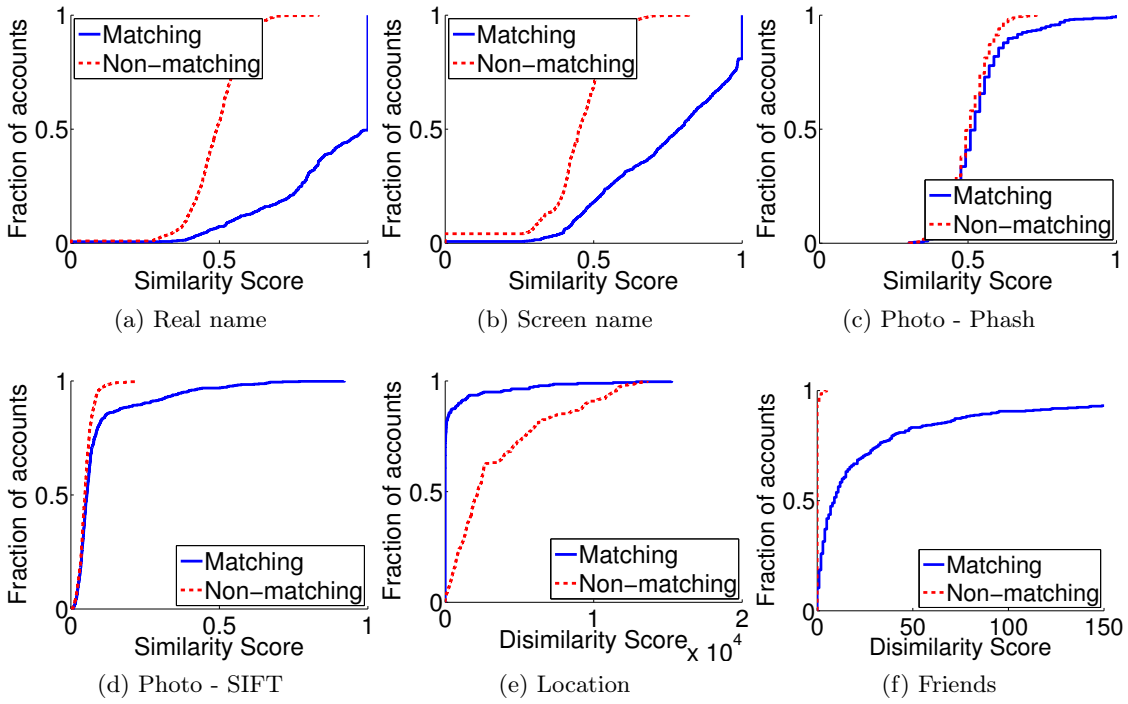
on different sites. This trend bodes well for our ability to match the different accounts of a user. Note that even if real names have such high consistency, we will see that they cannot be used alone to match accounts at scale because there can be multiple people sharing the same name. We also computed the percentage of matching accounts in Twitter and Facebook for which all public attributes in Table 4.2 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 accounts using only the public attributes that we consider in this chapter. Thus, we can at most hope to correlate accounts for 93% of users. This percentage represents an upper bound on the performance of public attribute-based matching schemes.

Second, some attributes such as real name are considerably more consistent than others such as location.

Third, the consistency differs between different social networks. Twitter and Facebook have consistencies among the lowest.

4.1.3 Attribute discriminability

We now investigate the *discriminability* of attributes, by which we refer to the extent to which different values of an attribute can be used to distinguish a single account from all other accounts in a social networking site.



(a) Real name (b) Screen name (c) Photo - Phash

(d) Photo - SIFT (e) Location (f) Friends

Figure 4.1: Discriminability of attributes: CDFs of similarity scores between Twitter and Facebook matching accounts and non-matching accounts from DATASET 1.

We can evaluate the discriminability of an attribute by comparing the similarity in attribute values between accounts that belong to different users with the similarity in attribute values between accounts that belong to the same user.

We randomly sample 1,000 Twitter and Facebook matching accounts and 1,000 non-matching accounts from DATASET 1 and we compute attribute similarity scores between all the account pairs. Figure 4.1a shows the CDF of real name similarity scores for matching and non-matching accounts between Twitter and Facebook. The X-axis represents the similarity scores, where zero means no similarity and one means perfect similarity. Figures 4.1b to 4.1f show the CDFs for other features. For location, zero means perfect similarity because it corresponds to locations that have the same latitudes and longitudes. For friends, zero similarity means no friends in common. Results for other pairs of social networks are similar, and are therefore omitted.

For the *real name* and *screen name* we see a clear distinction between distributions of matching and non-matching accounts. Non-matching accounts systematically have similarities around 0.5 while matching accounts have similarities around 1. This suggests that these features have a high discriminability, i.e., are efficient at separating matching and non-matching accounts. Note that, even if the discriminability of real names is very high, when considering entire social networks, there can be a non negligible number of users with the same name. For *photo*, the two distributions are generally similar, with many scores around 0.5 for Phash algorithm (this is a typical score returned by the perceptual hashing method for unrelated photos). The photo does not have a very good overall discriminability

because there are not many matching accounts that use the same profile photo. *Location* and *friends* also have a good discriminability between matching and non-matching accounts although it is lower than the discriminability of real names and screen names.

4.1.4 Attribute impersonability

There is an emerging trend of malicious attackers and spammers that create fake accounts to impersonate honest users. Since one can use any values to fill the profile information when creating an account on a social networks, it is very easy to impersonate users. Attributes such as real name, location, bio or profile photo are all very easy to impersonate. Screen names are harder to impersonate because they are unique in a social network, so the impersonator cannot have the same screen name with the honest user. Finally, friends are the hardest attribute to impersonate, out of the ones we consider in this chapter, because for an attacker it is very hard to establish links to good users [26].

4.2 One-step matching scheme

The ACID test, described in §4.1, helps us understand the quality, from a matching point of view, of public attributes. None of the public attributes satisfy all the ACID properties, however each property is satisfied by at least one attribute. In this section, we make a first attempt to build a solution to our matching problem by combining all the attributes together so that we can obtain a matching scheme that satisfies all the ACID properties. The one-step matching scheme aims to simulate previous approaches to match accounts. However, instead of just using previous algorithms, we opt to improve them and build the best possible configuration for the one-step matching scheme.

4.2.1 Design of the scheme

Recall from §3.1 that our problem definition is: given a account in one large social network, $a \in SN_1$, find its matching account in another large social network, $\hat{a} \in SN_2$, if it exists. We can reduce this problem to an easier problem: given two accounts on two social networks, $a \in SN_1$ and $b \in SN_2$, determine whether b is a's matching account. We design the Linker to solve this problem. Then, given an account $a \in SN_1$, we can use the Linker to check, for every pair of accounts (a, b) such that $b \in SN_2$, whether b is a's matching account. We can then return any account b that the Linker declares as matching as the result of our general account matching problem.

4.2.2 The Linker

We design the Linker as a binary classifier that determines whether two accounts on different social networks, $a \in SN_1$ and $b \in SN_2$, belong to the same user or not.

Input: An account $a \in SN_1$ and an account $b \in SN_2$.

Output: A binary answer whether b is a's matching account, and the probability they are matching accounts.

Objective: Correctly identify matching accounts without mistakenly identifying non-matching accounts as matching accounts, i.e., maximize true positive rate while keeping false positive rates very low.

Conception: We build a binary classifier that takes as input a feature vector $f(a, b)$ that captures the similarity between each attribute of a pair of accounts (a, b), where $a \in SN_1$ and $b \in SN_2$; and then outputs 1 if b is a's matching account; and 0, otherwise. We represent (a, b) with five features each corresponding to the similarity score between a and b for each of the five account attributes: real name, screen name, location, photo, and friends.

Realization: DATASET 1 contains ground truth on matching accounts, which we use to train the binary classifier. We build a training set with 1,000 samples of Twitter and Facebook matching accounts from DATASET 1 and with 1,000 samples of non-matching accounts chosen at random from DATASET 1. The Twitter and Facebook accounts used for training are different than the accounts used for evaluating the matching schemes (i.e, accounts used for testing). We use 10-fold cross validation to train the classifiers.

Matching accounts across social networks is very challenging for standard classification methods, because (as discussed in §4.1.2) user accounts may be incomplete (users may choose to omit their location or photo). We must either work with classification techniques that are robust to missing values or identify methods to impute the missing values. We evaluate how well four classification techniques handle missing values: Naive Bayes, Decision Trees, Logistic Regression, and SVM:

Naive Bayes decides if two accounts match based on the probability that each attribute's similarity score belongs to the matching class, assuming that the distribution of attribute scores in each class is based on a kernel density estimation. The Naive Bayes classifier has a natural way of handling missing values of a attribute: during training, attribute instances with missing values will not be included in the attribute-value-class probability computation. During testing, if a particular attribute vector has a missing attribute value, then that attribute will be omitted from the prediction calculation.

Decision Trees decide if two accounts match by traversing a tree of questions until they reach a leaf node; the leaf node then specifies the result. In our setting each node represents a threshold for a given attribute; the classifier tests the input account against that value and takes the appropriate branch. The most popular way to handle missing attributes is at training time to only create branches on present values, and at testing to take all the branches of the node representing the attribute whose value is missing and then select the class with the highest frequency among the leafs. Decision Trees prove useful for eliminating redundant attributes, and they allow to directly interpret results by

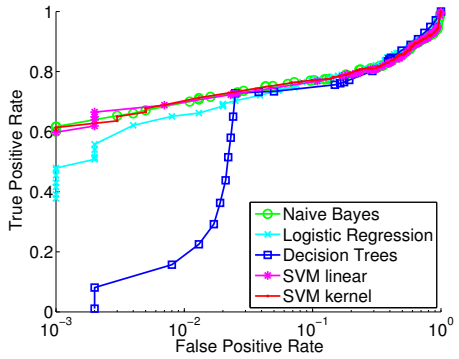


Figure 4.2: Comparison of the performance of different classifiers (matching Flickr to Twitter with combination of all attributes).

following the decision process. The drawback is that the decision boundaries are rough because Decision Trees can only make horizontal and vertical splits.

Logistic Regression is a linear classifier that bases its decisions on a linear combination of all the similarity scores of each attribute. Logistic Regression does not have a native way of handling missing values, so they must be inputted. The most common way is to replace missing values with the median, the mean of all existing attribute values or a values that does not exists in the dataset (e.g., -1). We tested both methods and replacing with the median value gives higher accuracy.

SVM is a large margin classifier that obtains the decision boundary with the largest distance between matching and non-matching observations. Boundaries can either be linear or not (kernel). Missing values are inputted in the same way as for Logistic Regression. The imputation of missing values with -1 gives the best results.

Figure 4.2 presents ROC curves to compare the performance of different classifiers using the Google+ dataset. The Naive Bayes and SVM classifiers (both linear and kernel) perform the best, Logistic Regression is close to the first two while Decision Trees exhibit the lowest performance for small false positive rates because of its rough decision boundaries to split matches and non-matches. Since Naive Bayes takes much less time to compute than SVM on our large data set, we use Naive Bayes for computing all results in this chapter.

SVM usually performs better than Naive Bayes in scenarios where all values are available, however its accuracy decreases due to missing values. Note that previous works on matching social networks did not evaluate different classifiers with respect to missing values.

Given a pair of accounts (a, b), our Naive Bayes classifier outputs the probability p that b is a's matching account. We select a cut-off threshold for p and the classifier returns 1 (i.e., matching accounts) if p is larger than the threshold; and 0 otherwise. The threshold's choice constitutes the standard tradeoff between true and false positive rates. We use 10-fold cross validation on the training set to analyze the tradeoff. There is a clear distinction between matching accounts and non-matching accounts: p is close to one for most matching

accounts, whereas p is close to zero for the vast majority of non-matching accounts. If we use a threshold of 0.94 for p and we consider all the pairs of accounts that have a probability higher than 0.94 as being matching accounts and all the rest as non-matching accounts, we detect 89% of the matching accounts with a false positive rate of 1%. We choose this operational point because most of the previous works focused on false positive rates at this scale.

4.2.3 Evaluation over a small dataset

To test the one-step matching scheme on the small dataset, we use the LINKER to predict whether each pair of Twitter-Facebook accounts match or not. As expected, the scheme correctly identifies 89% of the matching accounts with 1% false positive rate. At a first glance, these results are promising. However, at a closer look, these results translate into a 89% recall for a 8% precision, i.e., only in 8% of the accounts output by the LINKER are actually matching accounts. This is due to the fact that the number of possible non-matching accounts ($\sim$ 1 million) is much larger than the number of matching accounts 1,000. Thus, even for a small dataset we have to choose a threshold for p that corresponds to smaller false positive rates. For a threshold corresponding to a 10^{-3} false positive rate, we get a 85% true positive rate and a 45% precision while for a 10^{-4} false positive rate, we get a 83% true positive rate and a 89% precision. Thus, even for small datasets, to obtain a reliable matching we need to focus on very small false positive rates.

Unfortunately, in large social networks like Facebook, Twitter, and Google+, where the number of users is larger than 500 million, even a low false positive rate of 10^{-4} would still lead to several hundreds false positives. Since accounts have a single matching account on another site, distinguishing this single account from the haystack of hundreds of millions of accounts poses a serious scalability challenge.

Due to the small size of the dataset, we cannot estimate true positive rates with false positive rates lower than 10^{-4}. To evaluate the scheme at scale, we use in next section the method described in §3.5.3.

Takeaway: The one-step matching scheme can identify 83% of the matching accounts with a 89% precision on a small-scale dataset.

4.2.4 Evaluation at scale

To evaluate the one-step matching scheme at scale, we use the LINKER to determine the corresponding Facebook account for a given Twitter account a. More precisely, we use the LINKER to determine whether every pair of accounts (a, b) where $b \in C(a)$ match or not where the candidate set C(a) is built as described in §3.5.3. We return all accounts detected as matching as output. Ideally, the LINKER should correctly identify the single matching account and reject all the other accounts in the candidate set.

For the evaluation, we set a threshold probability for the LINKER to declare an account as matching at 0.99. For the small dataset of the previous section, this threshold corresponds

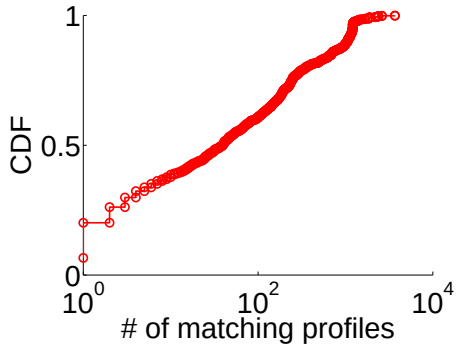


Figure 4.3: CDF of the number of Facebook matching accounts identified by the LINKER for each Twitter account.

to a zero false-positive rate due to the small number of non-matching account. Here, it will give a small number of false matches (see below).

True matches: LINKER successfully identifies 66% of all matching Facebook accounts. While this true positive rate is lower than the expected rate observed in §4.2, 83%, it is due to our limitations in generating the candidate set. Recall that, the candidate sets we generated contains the matching account only 70% of the time. With a more extensive candidate set containing a larger fraction of matching accounts, the true positive rate would likely improve.

False matches: Figure 4.3 shows the cumulative distribution of the number of Facebook accounts in the candidate set that LINKER identifies as matching to the Twitter accounts. For more than 62% of Twitter accounts, LINKER identifies 100 or more Facebook matching accounts. Note that the number of matching accounts is likely to be higher with a more extensive candidate set. Since there is only one matching account, our finding indicates that more than 99 out of 100 accounts are false matches, i.e., the LINKER has a $\approx 1\%$ precision. The high number of false matches highlights the reliability problem with applying LINKER to match accounts at scale.

The maximum precision the LINKER can achieve, i.e., when we set the probability threshold for detecting matches to 1 (the maximum possible), is 2.5% for a 50% recall. Note that, while such precision is very low for a reliable matching scheme, we could potentially use the LINKER as a filtering step.

Takeaway: The maximum achievable precision of the LINKER when evaluated at scale is 2.5% for a 50% recall. Such precision is unsatisfying.

4.2.5 Evaluation against human workers

To investigate the detectability of matching account by humans we set up an Amazon Mechanical Turk (AMT) experiment where we ask AMT workers if they think that a

Table 4.3: Fraction of matching and non-matching accounts detected by AMT workers out of the true matching and non-matching accounts; † $X\%(Y\%)$ = fraction of accounts detected with majority agreement (fraction of accounts with full agreement).

	True matching	**True non-matching**
Detected matching	58% (45%)†	0%(0%)
Detected non-matching	15% (7%)	85% (67%)
No Consensus	27%	15%

Twitter and a Facebook account belong to the same individual or not. We randomly select 100 pairs of matching and non-matching Twitter and Facebook accounts from DATASET 1. In each assignment, we give AMT workers a link to a Facebook and a link to a Twitter account and we ask them to choose between three options: *'the accounts belong to the same individual'*, *'the accounts do not belong to the same individual'*, and *'cannot say'*. For each assignment we ask the opinion of three different AMT workers. We say that we have a *full agreement* when all the AMT workers chose the same answer, a *majority agreement* when at least two AMT workers chose the same answer, and *no consensus* when all workers chose different answers or at least two AMT workers were unable to say if the accounts belong to the same individual.

Table 4.3 shows the fractions of accounts matched by humans out of the tested matching and non-matching accounts. The AMT workers were able to detect with a majority agreement 58% of the matching accounts and 45% with full agreement. For 27% of the matching accounts the AMT workers are not able to detect if they belong or not to the same individual, and for 15% of the matching accounts AMT workers said that they belong to different individuals. We manually investigated the latter cases and the accounts indeed correspond to different identities of the same user (i.e, either the accounts correspond to different personas, or one corresponds to a person and the other to an organization or interest group). The AMT workers were also able to detect with a very high accuracy the non-matching accounts: 85% are detected with a majority agreement; no accounts are mistakenly identified as belonging to the same individual; and for 15% there was no consensus. We can conclude that humans can achieve a 58% true positive rate for a 0% false positive rate (100% precision). We will see later in the chapter, that in more complex scenarios where AMT workers have to choose the matching account out of a list of similar looking accounts, they sometimes choose the wrong account (i.e., they have a lower precision).

Humans are not able to obtain a higher true positive rate not because they are doing a bad job but because the matching accounts that are not detected do not have enough available information to make the AMT workers confident that the accounts correspond to the same person. To ensure the quality of their work, AMT workers where asked to describe in free text the reasons they think the accounts correspond to the same person. They gave very detailed descriptions showing they are doing a thorough job (you can find some examples of their descriptions in Table 4.4).

The AMT workers' detailed descriptions allowed us to identify a number of attributes AMT workers found helpful for matching accounts. There are 6 main attributes AMT workers

Table 4.4: Examples of descriptions AMT workers gave about why they think two accounts belong to the same individual or not.

Detected Matching (True Matching)
· same account picture
· It is a different picture, but he has the same look on his face, same glasses and same hairdo.
· He likes the cubs and is a pilot, picture on fb has him with a plane and hes from illinois
· Names are the same. Twitter references corkscrews as does the facebook account under "groups". Locations are also the same.
· Both have the same name, they both have a baseball picture
· same tweet as post on Facebook.
· it is an unusual last name and for two people to have the same first name spelled in an uncomventional way and the same last name would be a rare coniciennce to me
Detected Non-matching (True Matching)
· Not a match, one is for music channel, other is for person.
· different name location information pictures
· No because they look totally different and have different names.
Detected Non-matching (True Non-matching)
· Two different photos, names and sex.
· The Twitter account appears to be of a woman, The Facebook account, a man.
No Consensus (True Matching and Non-matching)
· Other than the name of the accounts there is not enough information on either account to compare them
· twitter account has no tweets or pictures to help with identifcation
· Twitter account belong to american, facebook to hispanic in guatemala.
· Twitter account does not provide much information

used to identify matching accounts: if the *names*, *screen names*, *locations*, or *account photos* are the same on the two accounts; if it appears to be the *same person* in the photos and if any information in their *bio* matches. In some cases AMT workers went even further and said that from analyzing the posts and photos they think that the two accounts show the same *interests* or that they *link* to the same sites. Some of these attributes can be verified automatically with a low effort, such as identifying if two accounts have the same name, screen name, location or account photo. However, other are much harder to check by machines. For example, it is very easy for humans to identify that the same person appears in two photos whereas it is much more harder for a machine [6]. The average time an AMT worker took to decide if two accounts correspond to the same person is 6 minutes and 49 seconds.

Takeaway I: Human workers are able to detect 58% of the matching accounts without mistakenly mis-detecting any non-matching account.

Table 4.5: Proportion of detected and undetected matching accounts by AMT workers and by the LINKER.

		LINKER	
		Detected	Undetected
AMT	Detected	55%	3%
	Undetected	34%	8%

Takeaway II: Not all the matching accounts in the ground truth dataset correspond to the same person even if they are managed by the same person.

Takeaway III: These results suggest that a true positive rate of 58% is more indicative of the expected fraction of accounts a matching scheme could detect instead of 100% of the ground truth.

To get a better understanding of the accuracy of the LINKER, we compare it with the accuracy of human workers. We use the LINKER to match the accounts used for the AMT experiment to compare the LINKER against AMT workers. The LINKER is able to detect 89% of the matching accounts without any misclassification (§4.2.3) while AMT workers are able to detect 58% of the matching accounts with majority agreement, suggesting that the LINKER is doing a better job than AMT workers.

Table 4.5 details the proportions of matching accounts that AMT workers and the LINKER detect or not: 55% of the matching accounts are detected by both; 34% of the accounts are detected by the LINKER but not by AMT workers. Most of the accounts AMT workers miss and the LINKER detects have no location information nor photos but they have very similar real names or screen names. The LINKER puts more weight on the similarity between real names and screen names (and less on the other attributes) when it makes a decision, and consequently it often decides that two accounts match only because they have similar real names. AMT workers are more cautious when deciding if two accounts match and they check for other attributes besides the real name or screen name to match. The LINKER gives such a high weight to the similarity between real names because the real name is the most discriminative, available and consistent attribute and when looking at all accounts in a social network this is the attribute that best distinguishes matching accounts.

Furthermore, 3% of the accounts are detected by AMT workers but not by the LINKER. AMT workers are able to identify these accounts because the same person appears in the photos but the LINKER misses them because it does not do face recognition. The 8% of the accounts that are not detected by either AMT workers or the LINKER are accounts where the real name does not match and other attributes are not available. Note that this fraction is consistent with the 7% fraction of accounts with inconsistent values for all attributes we found in §4.1.2.

The LINKER has such high true and false positive rates because it often decides that two accounts match only because they have similar real names. AMT workers have a better precision because they check for other attributes besides real names to declare that two accounts belong to the same person.

Takeaway IV: The LINKER has a higher true positive rate than AMT workers because it often decides that two accounts match only because they have similar real names whereas AMT workers need other attributes beside real names to match. This comes at the cost, however, of a large false positive rate.

Takeaway V: AMT workers are not able to detect a higher percentage of matching account because, other than real names, the accounts have a low availability of attributes and/or low consistency.

4.3 Three-step matching scheme

Our analysis in the previous section reveals reliability issues with the one-step matching scheme when applied at scale. This section designs a three-step matching scheme to address this issue. We combine three classifiers that together find matching accounts with very high accuracy and abstains when its confidence in the output is low.

4.3.1 Design of the scheme

To design a scalable and reliable matching scheme, we leverage four observations: (1) the LINKER is not able to accurately detect matching accounts at scale, however it is very good at weeding out non-matching accounts while retaining a small set of possible matching accounts that contains with high probability the true matching account; (2) the LINKER often decides that two accounts match only because they have similar real names; (3) however, humans need multiple attributes to match between accounts to be confident that they correspond to the same person; and (4) an impersonator can copy most of the public attributes of an account, and trick the matching scheme to output the impersonator account instead of the matching account.

Figure 4.4: Three-step matching scheme.

The reason why the LINKER cannot achieve a good precision at scale is because we have to deal with very large datasets. For example Facebook has over 1 billion users, thus the data contains one billion matching accounts and one quintillion non-matching accounts. Moreover, the number of features we can use to distinguish the matching account out of hundreds of millions of non-matching accounts is small. To overcome this problem, instead of using one classifier to detect matching and non-matching accounts, we use two classifiers in sequence that achieve a better accuracy.

We propose a simple but efficient methodology based on three steps to build a scalable and reliable matching scheme, as shown in Figure 4.4. The FILTER takes as input a target account $a \in SN_1$ and *all* the accounts in SN_2 and returns a small candidate set $C(a)$ of accounts in SN_2 that are similar to a. The DISAMBIGUATOR takes the output of the FILTER and further disambiguates the matching account out of the candidate set. Furthermore, we take advantage of the fact that there can only be one matching account on a second social network to build the GUARD. The GUARD adds a second level of confidence. It checks the account retuned by the DISAMBIGUATOR and returns the account only if it is confident that is indeed the true matching account and abstains otherwise. Part of the GUARD's job is to detect impersonator attacks.

We present next in more details the three components of the matching scheme.

4.3.2 The FILTER

The FILTER weeds out obvious non-matching accounts and returns a small candidate set of possible matching account.

Input: A target account $a \in SN_1$ and all the accounts in SN_2.

Output: A candidate set $C(a)$ of accounts in SN_2 similar with a.

Objective: Minimize the size of $C(a)$ while keeping a high probability that the matching account $\hat{a} \in C(a)$.

Conception: Several techniques are possible to build the FILTER:

Indexing: We can opportunistically use the Facebook search API to find accounts with the same or similar names to the target account. Indexing techniques are often used for such tasks [34]. These techniques are computationally light and scalable however they are limited to using names to find similar accounts.

Supervised approach: Use the LINKER to build the candidate set. The drawback of such approach is that it needs to compute the similarity between the target account and *all* the accounts on SN_2, which is inefficient. However, this approach is able to use multiple attributes to find similar accounts.

Unsupervised approach: We could map all the accounts in an unsupervised way so that we can cluster similar accounts without the need of computing the similarities among all the accounts. Such approach is not straightforward and we leave it as future work. Furthermore, this approach is more suitable when we have access to the whole social network data and has limited benefits if we have to use the Facebook API to gather portions of data.

Realization: We opt for a combination between the first and the second approach because this is what we can practically do. Since we do not have access to all the accounts on Facebook we use the Facebook search API to generate a candidate set, $C_{Fb}(a)$ (as described in §4.2), and then we apply the LINKER (with a threshold corresponding to 1% false positive rate) on $C_{Fb}(a)$ to weed out accounts that are not similar enough to build $C(a)$. Recall from §3.5 that for 70% of Twitter accounts $\hat{a} \notin C_{Fb}(a)$.

4.3.3 The DISAMBIGUATOR

The goal of the DISAMBIGUATOR is to find the matching account in the candidate set of accounts returned by the FILTER.

Input: A target account $a \in SN_1$ and $C(a)$.

Output: The account in $C(a)$ most similar to a. We call this account the TOPMATCH.

Objective: Maximize the probability $\hat{a}$ is the most similar account in $C(a)$ to a.

Conception: Similar to building the LINKER, we train another probabilistic classifier to determine whether an account in the candidate set is the matching account or not.

Realization: To build a training dataset, we take 1,000 Twitter accounts from DATASET 1 (different from the ones used for testing the matching schemes) and for each Twitter account, a, we generate the candidate set, $C_{Fb}(a)$, using the Facebook search API, we then filter $C_{Fb}(a)$ using the FILTER to obtain $C(a)$. Since we know the Facebook matching accounts for all the Twitter accounts we can train the classifier with matching accounts $(a, \hat{a})$ where $\hat{a} \in C(a)$ and non-matching accounts (a, b) where $b \in C(a)$ and $b \neq \hat{a}$.

Similar to the LINKER, given two accounts (a, b), the DISAMBIGUATOR outputs the probability, p, that they are matching accounts. Note that the DISAMBIGUATOR estimates would be different from the LINKER as the training sets and the discriminability of the public attributes within the training sets are very different. For example, real name and screen name attributes have high discriminating power in the training sets used for LINKER (see Figure 4.1a, 4.1b), while they have limited discriminating power in the training set here (see Figure 4.5a, 4.5b and compare with Figure 4.1a, 4.1b). Contrary to the LINKER the DISAMBIGUATOR gives high weights to the similarity between other attributes beside the real names.

We could proceed by searching a threshold for p to detect matching accounts by investigating the tradeoff between the true and false positive rates. Alternatively, to improve the reliability of the DISAMBIGUATOR, for each Twitter account a, we choose to return instead only the Facebook account with the highest probability of being a matching account from $C(a)$. The intuition is that while some Facebook accounts might cross the probability threshold, the Facebook matching account would match the Twitter account better than

other accounts. This can increase the precision because it forces the DISAMBIGUATOR to only output one matching account. Secondly, even if the Facebook matching account does not cross the probability threshold, it might still be the most similar account in $C(a)$. This can potentially increase the recall.

4.3.4 The GUARD

The GUARD ensures a reliable matching by distinguishing between the scenarios where the DISAMBIGUATOR outputs a true match versus a false match and abstain from producing an output in the latter case. The GUARD exploits the fact that there can only be one matching account in a social network.

Input: The two most similar accounts in $C(a)$, i.e. the accounts with the highest probabilities to be matching accounts as computed by the DISAMBIGUATOR.

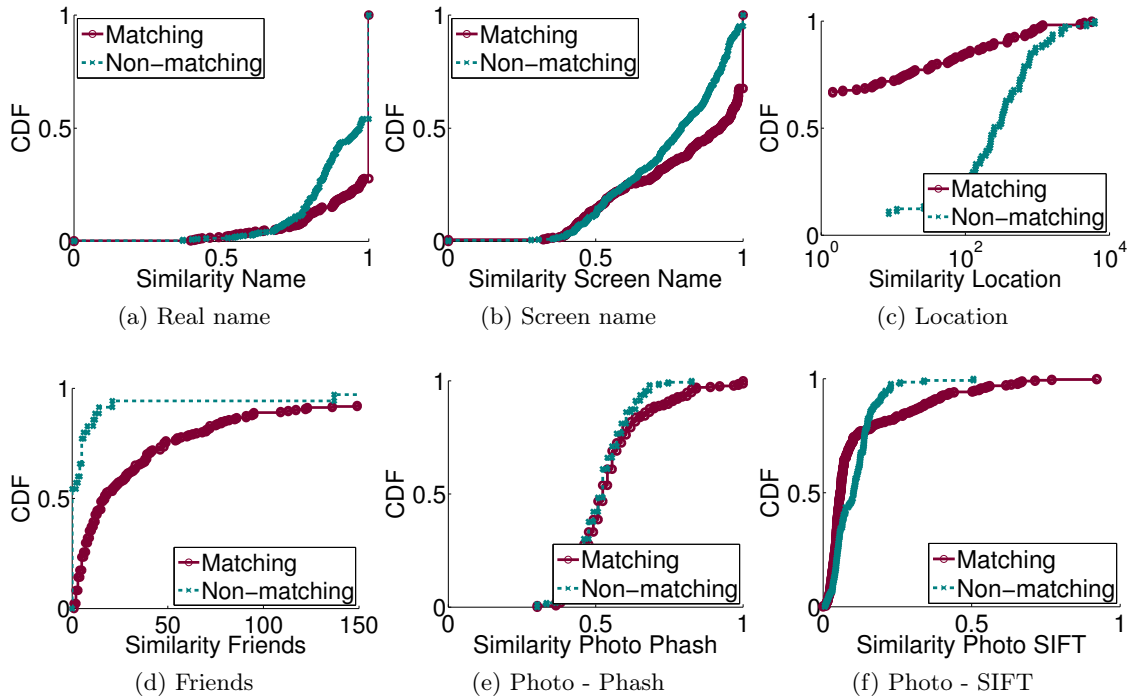
Output: The matching account on SN_2 or nothing.

Objective: Minimize the number of false matches while maximizing the number of true matches returned by the matching scheme.

Conception: There are three scenarios where the DISAMBIGUATOR might output a false match:

1. If an attacker creates a cloned account on SN_2 that is more similar than the true-matching account. It might be possible to distinguish these cases as there will be two accounts (the real and the impersonator) that both have high similarities with the target Twitter account, and the difference between their similarity is small.
2. When the true matching account in Facebook does not exist in the candidate set, forcing the non-matching account that is most similar to the given Twitter account to be chosen as output. Intuitively, we might be able to distinguish this case, as the attributes of the TOPMATCH might be less similar to those of the given Twitter account compared to the scenarios where the true matching account is provided as output.
3. When the true matching account exists in the candidate set, but a falsely correlated Facebook account is chosen as output because it is more similar to the given Twitter account than the true matching account (due to the lack of attribute availability and/or consistency). Intuitively, it might be possible to distinguish this case, as there would be multiple Facebook accounts with relatively high similarities to the given Twitter account.

To investigate evidence that supports these intuitions, we analyze cases when the TOPMATCH is a true match and when it is a false match. We investigate a dataset containing 500 TOPMATCH accounts that are true matches and 500 that are false matches. Figure 4.5



(a) Real name (b) Screen name (c) Location

(d) Friends (e) Photo - Phash (f) Photo - SIFT

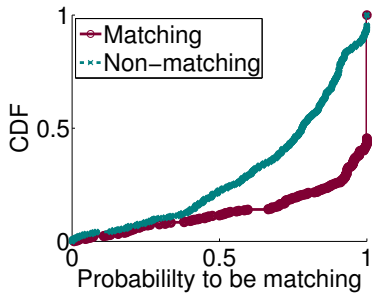
Figure 4.5: CDF of similarity scores for each attribute, when TOPMATCH is the matching account and when it is not.

confirms our intuition, it presents the cumulative distributions of attribute similarity scores between the given Twitter account and the TOPMATCH. We show two distributions per plot, one when the TOPMATCH is the matching account and another it is not. Matching account are in general more similar to the Twitter account across the different attributes than non-matching account. Thus, we can leverage the differences to distinguish the cases when the TOPMATCH is a true match vs. a false match.

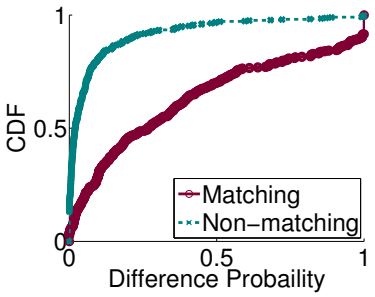
Figure 4.6a shows the cumulative distributions of DISAMBIGUATOR probabilities for TOPMATCH to be a matching account. We plot two separate distributions, one when the TOPMATCH is the matching account and another when it is not. The graphs show that DISAMBIGUATOR matching probabilities are higher when the TOPMATCH is the matching account than when it is not. For example, the median probability of a matching account is 1, whereas this value is 0.78 for non-matching account.

Figure 4.6b shows the cumulative distributions of the difference in DISAMBIGUATOR's probabilities between the TOPMATCH and the second most similar account. When the TOPMATCH is the matching account, the median difference to the second most similar account is 12 times more than when the TOPMATCH is not the matching account.

Realization: Based on these observations we can build a binary classifier, that takes as input the DISAMBIGUATOR estimates of the probabilities that the top two accounts are matching accounts and determines whether the TOPMATCH is a true matching account. The DISAMBIGUATOR assumes that all the accounts in $C(a)$ are independent. The GUARD



(a) Probability to be correlated computed by the DISAMBIGUATOR.



(b) Difference between the similarity of the TOPMATCH and the second most similar account.

Figure 4.6: CDFs when the TOPMATCH is the matching account and when it is not.

takes into account that the accounts are actually dependent on a and exploits the structure in $C(a)$.

For training the classifier, we again select 1,000 Twitter accounts from DATASET 1 (that are not used for testing the matching schemes or for training the FILTER and DISAMBIGUATOR) and we apply, as described, the FILTER and DISAMBIGUATOR to find the TOPMATCH. We train the classifier with the probabilities of being matching accounts of the two most similar accounts in $C(a)$ when TOPMATCH is the matching account and when it is not. The GUARD is able to detect 42% of the cases when the TOPMATCH is the matching account while it only misclassifies 1% of the cases when the TOPMATCH is non-matching account. It can also operate at a 60% true positive rate for a 5% false positive rate.

4.3.5 Evaluation at scale

We investigate the accuracy of the three-step matching scheme at each step.

FILTER (PRECISION = 1%, RECALL = 69%) Similar with the evaluation of the LINKER at scale in §4.2 the FILTER is able to attain a recall of 69% for a precision of 1%.

DISAMBIGUATOR (PRECISION = 50%, RECALL = 50%) We sort the Facebook accounts in $C(a)$ according to p. We define the *rank* as the position of the matching account in the sorted list of candidate accounts. The best case is when the matching account has a rank of one. Figure 4.7 shows the CDF of the rank of the matching account. The X-axis is in log scale to focus on small ranks. When the matching account is not in $C(a)$ we put the rank to be 10,000.

The DISAMBIGUATOR yields the Facebook matching account for 50% of the Twitter accounts. For the remaining Twitter accounts, the TOPMATCH is *not* the matching account. While 50% might sound low, note that for 30% of the Twitter accounts, the candidate set does not contain the matching account and thus, it cannot yield a true match. Excluding

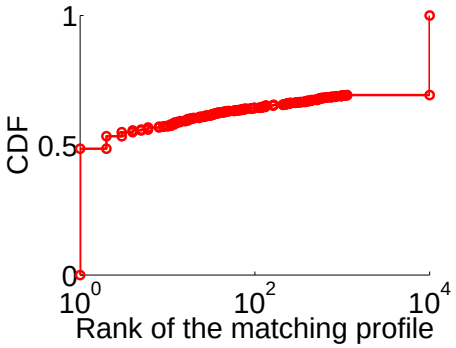


Figure 4.7: CDF of the rank of the matching account for matching Twitter to Facebook accounts using the FILTER and the DISAMBIGUATOR.

such cases, the percentage of Twitter accounts for which DISAMBIGUATOR yields a true match is as high as 75%. While the recall of the DISAMBIGUATOR might be acceptable at 50% (or higher), the precision of the DISAMBIGUATOR is still too low at 50%. Any reliable matching scheme would require the precision to be substantially higher a challenge that it will be solved by the GUARD.

If instead of selecting the TOPMATCH, we used a threshold on the probability p that a is the matching account, as we did for the LINKER, the FILTER and the DISAMBIGUATOR would achieve a 30% recall for a 23% precision. The precision is better than the LINKER but it is still unsatisfactory. Perhaps the most important asset of the DISAMBIGUATOR, however, is in increasing the gap between the true and false matches to make the GUARD more accurate. Indeed, the DISAMBIGUATOR separates very well the matching account (giving a high probability) from other accounts in $C(a)$ (giving a low probability). On the contrary, the LINKER gives very close probabilities to be matching to all accounts in the candidate set (i.e., all accounts with similar real names).

GUARD (PRECISION = 98%, RECALL = 21%) The DISAMBIGUATOR returns the same number of true and false matches (on average 50% of the TOPMATCH accounts are true matches and 50% are not). When we test the GUARD we obtain a 21% recall for a 98% precision or a 30% recall for a 92% precision.

Thus, the three-step matching scheme can match 21% of the Twitter accounts to their corresponding Facebook accounts with a very high precision. Whenever the GUARD returns an account 98% of times it will be the true match and only 2% of times will be a false match. Although 21% true positive rate may seem low, the next section shows that it is almost as good as what humans can do.

Takeaway: The three-step matching scheme can match 21% of Twitter accounts to their corresponding Facebook accounts with 98% precision, or match 30% of Twitter accounts with a 92% precision. To increase the precision the scheme has to trade off recall.

4.3.6 Evaluation against human workers

We compare the performance of the three-step matching scheme with that of humans in an AMT experiment. We randomly select 200 Twitter accounts from DATASET 1 (that are not used for training the matching scheme). In each assignment, we give AMT workers a link to a Twitter account as well as links to the 10 most similar Facebook accounts (we shuffle their position) and we ask AMT workers to choose the matching account. We also allowed AMT workers to say that they are unable to identify the matching account. For each assignment we ask the opinion of three different AMT workers. We design two versions of the experiment: in the first if the matching account is not in $C_{Fb}(a)$ (the candidate set returned by the Facebook search API), the matching account will not be in the list of 10 Facebook accounts; and a second version, where whenever the matching account is not in $C_{Fb}(a)$ we replace one of the 10 Facebook accounts with the matching account. The first version of the experiment gives AMT workers similar conditions with the matching scheme while the second version gives them perfect conditions, i.e., there is no limitation introduced by the Facebook search API.

In the first version of the experiment, AMT workers are able to match 25% of the Twitter accounts to their Facebook matching accounts with full agreement and 40% with majority agreement; 2% of Twitter accounts are matched to the wrong Facebook accounts with full agreement, and 4% with majority agreement. This means AMT workers achieve a 25% recall for a 98% precision, which is only slightly better than three-step scheme, or a 40% recall for a 96% precision. Thus, under similar conditions, the accuracy of AMT workers is not much higher than the three-step matching scheme. Under perfect conditions (second version), AMT workers can correctly match 27% of Twitter accounts with full agreement and 58% with majority agreement.

To understand the limitations of the three-step matching scheme we compare the accounts AMT workers detect vs. the accounts the matching scheme detects. About half of the matching accounts detected by AMT workers with full agreement and 40% with majority agreement, the matching scheme also detects. To understand why the matching scheme does not detect the rest, we manually checked the descriptions AMT workers gave in the assignments. About 58% of the missed accounts AMT workers detect are because they have the same person in the photos, 15% have some matching bio information and the rest have some sort of content, interests that match. These results hint at ways we can improve the matching.

The matching scheme is able to detect 5% of matching accounts that AMT workers do not detect. These accounts have friends in common which AMT workers do not check. These results suggest that if we combine AMT workers with the matching scheme we can potentially attain a 30% recall (25%+5%) for a 98% precision or 45% recall (40%+5%) for a 96% precision.

Takeaway: Under similar conditions, the AMT workers do not detect much more accounts than the three-step matching scheme, AMT workers can achieve a 25% recall for a 98% precision.

4.4 Testing the reliability of the three-step matching scheme

To ensure that the three-step matching scheme is indeed reliable, we test how well it performs when there is no matching account on the second social network, and when there is an impersonator.

4.4.1 Reliability in the absence of a matching account

In practice, a Twitter user may not have a Facebook account. Ideally the matching scheme should return no account in this case. To test the reliability of the matching scheme in this scenario we take the dataset used for the evaluation at scale and for all Twitter users we remove the Facebook matching account from the candidate set. The matching scheme only returns a false matching account for 1% of the Twitter accounts. We manually investigate the 1% cases when the matching scheme returns an account. Half of these cases correspond to a false match; the other half, however, correspond to accounts of the same person. These accounts correspond to either impersonators or people that maintain duplicate accounts on Facebook. Thus, the matching scheme is reliable when there is no matching account on the second social network.

4.4.2 Reliability to impersonation

We now study the vulnerability of the matching scheme to impersonation attacks. We distinguish two different scenarios when impersonation attacks occur: (1) when a user does not have a matching account on the second social network and (2) when a user has a matching account on the second social network.

In the case where there is no matching account on the second social network the matching scheme is vulnerable to impersonation attacks. The matching scheme is based only on public attributes. So an attacker can easily gather the attributes of the public account of the victim in one social network and create a cloned account on the second social network. If the impersonator creates an account that has the same real name, similar screen name and has the same account photo or location, she would be able to fool the matching scheme into marking the impersonated account as the matching account.

In the case where the user has an account on the second social network, the task of impersonation becomes much harder. Recall from §4.3 that GUARD is designed to be resilient to impersonation attacks. If the attacker creates a cloned account on the second social network then there will be two accounts (the real and the impersonated one) that have high similarities and there is a good chance that the difference between their high similarity scores will be small. In this case, the GUARD will not return a matching account, avoiding a false match. The only scenario when the attacker can trick the matching scheme is when the matching account does not have a good similarity with the target account.

In fact, we can use the robustness of the matching scheme to impersonation to study if spammers and malicious attackers are impersonating accounts today. To search for impersonators, we check Twitter accounts in DATASET 1 that have two Facebook accounts

with probabilities to be matching (as returned by the DISAMBIGUATOR) higher than 0.95 that are rejected by the GUARD. The intuition is that one of the accounts might be an impersonator. We collected 16 such accounts. For each of the 16 Twitter accounts, we take the account which is not the matching account and we ask AMT workers whether the Twitter and this Facebook account belong to the same person. If someone is trying to impersonate, then the AMT workers should think the accounts belong to the same person. AMT workers said that four accounts out of 16 belong to the same person. This shows that the design of the matching scheme makes it resilient to impersonation attacks in cases where the victim has created an account on both sites and that it can be used as a tool to find impersonators. We will explain this property in Chapter 6.

Takeaway: The three-step matching scheme is reliable both when there is an impersonator on the second social networks and when there is no matching account.

4.5 Matching in the wild

We implement a prototype of the three-step matching scheme to match Twitter to Facebook accounts on demand. We deployed the prototype at: `http://matchingaccounts.app-ns.mpi-sws.org/`. We evaluate the matching scheme on a set of active Twitter accounts. We do not know whether these accounts have a corresponding Facebook account or, when they do, which one is the corresponding Facebook account. Our goal is to evaluate for which fraction of these Twitter accounts can we identify a matching account.

We collect a set of Twitter users by tapping the Twitter streaming API form March to June 2013. We extract all the users that tweeted something during this period. From this set of users we randomly sample 100 popular users (i.e., with more than 1,000 followers), 100 medium popular users (i.e, with between 100 and 1000 followers) and 100 unpopular users (i.e, with less than 100 followers).

We measure the fraction of Twitter accounts for which the matching scheme returns an account. The matching scheme returns an account for 16% of the unpopular, 17% of the medium popular, and 7% of popular Twitter accounts. These fractions are lower than what we found when matching Twitter and Facebook accounts in DATASET 1 because they include cases when users do not have a matching account. For popular users the fraction is even lower. This result is likely due to fact that popular Twitter accounts are either celebrities or organizations such as newspapers that have Facebook pages, not accounts. This will be taken into account in future improvements of the online service.

Takeaway: We provide an online service that can match a significant portion of Twitter accounts to their corresponding Facebook accounts in the wild and on demand.

4.6 Summary

We conducted a systematic and detailed investigation of how to reliably match user accounts across large-scale, real-world online social networking sites like Twitter and Facebook. Our analysis yielded a number of key insights which we summarize below.

We analyzed the ACID properties of public attributes of profiles such as name, location, and photos. We found that they satisfy these properties to different extents, though none satisfies all properties. Nevertheless, a large majority of users provide consistent values for public attributes across the different social networking sites we study, which enables cross-site account matching using only public attributes. Only 7% of users maintain different personas on different sites. Finally, the profile attributes for the users in DATASET 2 (a biased set of users who willingly link their accounts) are more available and consistent than for the users in DATASET 1 (a more representative set of users). As the availability and consistency of features directly impacts the accuracy of the matching schemes, this has implications on the representativeness of results showed by previous works (most of the previous works evaluated their schemes over datasets similar to DATASET 2).

Humans are good at recognizing persons, so we decided to set up an Amazon Mechanical Turk experiment to see what fraction of accounts can be detected by humans as belonging to the same user out of a list of accounts we know, a priori, match. Surprisingly, the results show that humans can only detect 58% of a random sample of Twitter and Facebook accounts that belong to the same users. This result has two sides. It demonstrates that it is indeed possible to match a large fraction of accounts across social networks only using the public information present in user profiles, but it also demonstrates that there is another important fraction of accounts that even humans cannot match. Consequently, this shows that is unrealistic to expect that a matching scheme will be able to detect 100% of a random sample of matching accounts.

It is easy to achieve high recall (83%) with high precision (89%) when attempting to match accounts between datasets containing only a few thousand accounts. Nevertheless, maintaining a high recall without compromising precision is a challenge at the scale of today's popular social networks, which contain hundreds of millions of users. The three-step scheme is able to match 21% of accounts between Twitter and Facebook with 98% precision. Although we cannot claim that 21% is a high recall, humans, under similar conditions, can only detect 25% of matching accounts. Note that Twitter and Facebook have the lowest attribute availability and consistency, thus the accuracy of matching other social networks should be higher than what we obtain for Twitter and Facebook. Most of the accounts that we cannot match are of users who barely use Twitter, so one can argue that it is not as interesting to match these accounts. We achieve this performance thanks to a combination of three learning algorithms with separate training, which better leverage the power of the different attributes. We believe that this approach can lead to improvements in other applications using learning. Furthermore, the three-way matching scheme is robust when a user does not have an account on a second social network as well as to impersonation attacks. Finally, we can match a significant portion of Twitter accounts to their corresponding Facebook accounts in the wild and on demand.

Overall, our findings reflect the potential as well as the limits of reliably matching accounts

at scale. We implemented a prototype of the three-step matching scheme to match Twitter to Facebook accounts on demand. We deployed the prototype at http://matchingaccounts.app-ns.mpi-sws.org/. The online service can be used by privacy conscious users to check how well one can detect their accounts in different social networks.

Chapter 5

Matching Accounts using Public Innocuous Information

In this chapter we study how potential attackers can identify accounts on different social networks that all belong to the same user (even when profile attributes do not match) by exploiting only innocuous activity that inherently comes with posted content. This study has significant privacy implication as it presents a novel class of attacks that exploit users' tendency to assume that, if they maintain different personas with different names, the accounts cannot be matched together.

In contrast to the previous chapter, we focus on exploiting implicit features derived from a user's *activity*, rather than leveraging information explicitly provided—and hence more easily controlled—such as name or profile photo. Specifically, we explore matching accounts based on *where*, *when* and *what* a user is posting. As it turns out, combining these three types of features provides attackers with a powerful tool to match accounts.

Our focus in this chapter is on demonstrating the feasibility of matching accounts based on innocuous data rather than the scalability and reliability of the matching scheme. We devise a possible set of attack heuristics, yet we emphasize that our choices are far from exhaustive. We also emphasize that, contrary to the previous chapter, it is unrealistic to expect such attacks to work reliably in a fully automated fashion. Given the large number of accounts online, even small false positive rates would quickly render any fully automated approach infeasible. In that setting, identifying a small candidate set of accounts on other networks is sufficient to allow for manually sifting through for the correct match.

We account users with three implicit features of their activity: the geo-location attached to a user's posts; the timestamps of a user's posts; and the user's writing style modeled with a probabilistic approach. We first evaluate the potential of each of these three features individually to match user accounts across social networks (in §5.2, §5.3, and §5.4, respectively). Then, we evaluate the improvements in accuracy that result from combining all three features (§5.5). Our results show that, when available, location and timing are powerful for matching accounts across social networks while a user's language model is not as effective. We find that the combination can identify almost as many matching accounts between Flickr and Twitter as when we exploit screen names (a much more obvious feature to key on). Moreover, the three features together can identify 37% more matching accounts between Yelp and Twitter than screen names.

Our work demonstrates a novel *class* of attacks by showing that innocuous features of a user's posts can help match accounts across social networks. Indeed, it remains the very fact that users want to post content that makes them vulnerable.

In this chapter we analyze matching attacks with data collected from three social networks: Flickr, Twitter, and Yelp. We choose these social networks because of their popularity and because they represent different types of social networks: photo sharing, micro-blogging, and service reviewing. We note that many Flickr, Twitter, and Yelp users may *not* necessarily consider account matching across these networks as a compromise of their privacy (in fact, 40% of the Flickr users in our dataset have an identical screen name on Twitter). We use them to demonstrate a technique that would also apply to users with different screen names as well as to more sensitive sites, for which the users may care if they were aware of the threat (e.g., dating or medical sites).

5.1 Features of innocuous activity

Our overall goal is to understand how user activity on one social network can implicitly reveal their identity on other social network. To this end, we have to extract *features* that we derive from user activity to build activity profiles. We choose three types of features for building activity fingerprints that are present on many social networks: location, timing, and language characteristics.

Location Many social networks provide location information directly in the form of geotags attached to user content, potentially with high accuracy if generated by GPS-enabled devices like mobile phones. However, even without geotags, one can often derive locations implicitly from posted content (e.g., when users review a place on Yelp, that gives us an address). Furthermore, a number of online services map images and textual descriptions to locations or geographic regions (e.g., by identifying landmarks) [2, 5, 33, 39, 91]. For our study, we use the *location fingerprint* of a user, i.e., the list of all locations associated with her posts on a specific social network. The intuition behind that choice is that the combination of locations a user posts from may sufficiently fingerprint an individual across social networks.

Timing Many mobile services and applications such as `Gowalla`, `Foursquare`, and `Instagram` allow users to automatically send content to multiple social networks simultaneously. The resulting posts then have almost identical timestamps, which we can exploit to match the corresponding accounts.

Language The natural language community has demonstrated that users tend to have characteristic writing styles that identify them with high confidence [135]. While these methods typically work best with longer texts, such as blog posts or articles, it is unknown how they perform for short texts such as tweets and how they can contribute to matching attacks.

5.2 Location fingerprint

In this section, our goal is to understand the degree to which locations attached to user content are sufficiently unique to identify an individual. The availability of location data differs between social networks. On Twitter, 5% of users have at least one post with geotags, on Flick, 4% of users have at least one photo that has geotags, while on Yelp, all the users have posts with location metadata. Our focus here is not necessarily on the scalability of the matching scheme but rather on showing that is possible to match accounts using only the location information that comes with their posts even if there are not many users that have such location data. The location metadata is one of the hardest features to impersonate as the attacker must travel and upload content from the same places from where the victim uploads. Measuring the consistency and discriminability of location involves two parts, which we discuss in turn: *(i)* representing a user's location in the form of a fingerprint suitable for comparison; and *(ii)* defining a similarity measure between two such accounts. For evaluation, we focus on matching accounts from the Yelp and Flickr GT^{area} sets to the Twitter C^{area} sets. Based on the results, we also investigate what properties enable matching users successfully by their location fingerprints.

5.2.1 Building the fingerprint

To motivate the use of locations, we start by examining the degree to which location fingerprints represented as zip code sets uniquely identify a user. Out of all Twitter accounts from the combined sets C^{area} from all the five areas, 91% exhibit *unique* zip code combinations (i.e., no other user posts from the same set of zip codes). Of the remaining 9%, almost all post from only a very small number of zip codes: 74% only post from one, 21% from two, 5% from more than two, and 5 accounts post from more than ten locations. Manually inspecting the latter, we find that three of them appear to belong to a single person maintaining separate personas on Twitter—which, incidentally, means we have just matched related accounts by their location information. For Flickr, 96% have unique zip code sets; out of the remaining 4%, 97% post from only one zip code and 3% from two. For Yelp, 77% have unique zip code sets; out of the 23% non-unique ones, 89% post from one zip code, 8% from two, and 3% from more than two zip codes. These results encourage us to use locations to fingerprint users.

We define a user's *location fingerprint* as a histogram that records how often we observe each location in her posts. The histogram's bins represent "location units", such as zip code, city, coordinates of a longitude/latitude cluster or region[1]. To eliminate the bias of users posting more often on one social network than another, we normalize each histogram by the total number of location units in the histogram such that they represent probability distributions.

As location units, we test three different types of choices:

[1]We also experimented with other fingerprint representations, such as a binary vector just indicating whether a location is present and non-histogram approaches such as matching directly on geo-coordinates, but the histogram approach provided the best results.

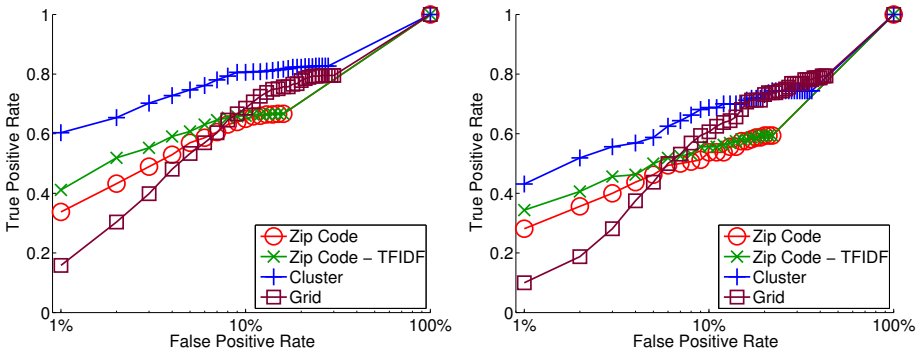
Grids: We map each latitude/longitude geo-coordinate to the cell within a spatial grid that has its center closest to the coordinate. Considering cell sizes ranging from 1x1 km^2 to 12x12 km^2, 10x10 km^2 proves most effective in our experiments.

Administrative regions: We map each latitude/longitude geo-coordinate to an address using the Bing Maps API [1]. Trying alternative address granularities (streets, zip codes, cities, counties, states), we find zip codes yield the best results.

One problem with representing location fingerprints as normalized histograms of zip codes is that all zip codes contribute the same to the similarity between two accounts. That however is undesirable as some zip codes are much more popular than the others (especially on Yelp, where people go out). Profiles containing those zip codes are therefore likely to have a high similarity even if they do not correspond to matching accounts. To adjust for that effect, we borrow the *term frequency - inverse document frequency (TF-IDF) [86]* weighting scheme from the information retrieval field to weight zip codes proportionally to their popularity. We apply TF-IDF as follows: for each zip code in an account's location fingerprint, TF represents the frequency of the zip code in the location fingerprint, and IDF represents the number of times the zip code appears in other location fingerprints in C^{area}. Then the weight of the zip code is $TF/log(IDF)$. With TF-IDF, zip codes that are less common across all accounts but more representative of specific location fingerprints have higher weights.

Clusters: We use a clustering approach as a more dynamic scheme to group geo-coordinates into regions. Using the k-means algorithm with an Euclidean distance, we group latitude/longitude geo-coordinates from all users in each C^{area} into corresponding longitude/latitude clusters. A small cluster represents a popular small area (e.g., blocks of downtown San Francisco), while larger clusters represent bigger, less populous regions (e.g., a park or forest). Our experiments show that using 10,000 total clusters per area produces the best results. We then associate each geo-coordinate with its N closest clusters. We assign weights to each of the N clusters based on a Gaussian distribution, with mean equal to the location, and variance set to 400 (the optimal value according to our experiments). In this approach, the cluster with the centroid closest to the location is assigned the largest weight of 1. The remaining clusters are assigned decreasing weights equal to values along the tail of the Gaussian distribution, according to the distances of their centroids to the geo-coordinate. Using $N > 1$ is better because associating more clusters to each location represents a 'soft' assignment. This soft assignment is advantageous in cases where locations of a user's posts in one social network is close to, but not exactly the same as, the locations of the same user on a different site. In our experiments N=20 produces the best results. We obtain the final cluster-based location fingerprint histogram for an account by first adding the weights of all clusters associated with all locations of the account, and then normalizing the weight of each location by this total sum of the weights.

Figures 5.1a and 5.1b compare the accuracy of using histograms at grid level, zip code level, zip code level weighted with TF-IDF, and cluster level at their best configurations. We use the Cosine distance to measure the similarity between histograms (in the next section,



(a) Flickr to Twitter (b) Yelp to Twitter

Figure 5.1: ROC curves for different location representations for matching Flickr and Yelp users (GT^{SF}) to Twitter users (C^{SF}).

we explore alternative choices). Figure 5.1a shows the ROC curve for matching Flickr to Twitter for users in San Francisco (the conclusions were similar for other cities), and Figure 5.1b for Yelp to Twitter. We obtain each ROC curve by varying the similarity score threshold from highest to lowest similarity score values, and computing the true positive rate (TPR) and false positive rate (FPR) when only considering as a match accounts with similarity score above the threshold. These plots take into account all pairs $s(a_i, b_j)$, where $a_i \in GT^{SF}$ and $b_j \in C^{SF}$. The best case would be a vertical line at 0% FPR followed by a horizontal line at 100% TPR; a random classifier would be a diagonal line from 0% TPR and FPR, to 100% TPR and FPR. Note that the plots are in log scale to focus on low false positive rates.

Grids have the lowest TPR in both cases. For a FPR of 1%, grids never achieve TPR higher than 20%. Users from densely populated areas have a greater chance of being confused with one another, when using grids, because the places from where they post tend to be closer to each other, which makes users post from different grids less often. In addition, in less populated areas (e.g national parks), grids split places that should be considered the same in different locations, which makes matching accounts that post from different grids in the same place look less similar.

Zip codes achieve higher TPR than grids, in particular when combined with TF-IDF, because zip codes take into account population density. Clusters achieve the highest TPR for all values of FPR. Their accuracy is significantly better for small FPR, which is the operational point we are interested in. For example, when the FPR is 1%, the TPR for identifying Flickr users in Twitter is 60%. Clusters have higher accuracy because they capture population densities. Furthermore, the soft cluster assignment finds similarities in cases when a user posts from two close by zip codes in her Flickr and her Twitter account. We analyze all ground truth users for which the location fingerprint of their account in Flickr and Twitter had no zip code in common (i.e., they had similarity score equal to zero when using zip codes). Half of these users were indeed posting from neighboring zip codes, and hence had higher similarity scores when using clusters.

While the above considers the complete data sets, we also examine building location finger-

prints individually per time interval: one month, one year, two years, three years, and all available data. Our results show that by aggregating at smaller time intervals, we end up removing too many data points from the accounts, making them less precise. While doing so helps to better identify a few prolific users, it impacts most users negatively.

The clusters made of complete datasets of posts achieve higher TPR than grids and zip codes, in particular for low FPR, thus we use them for the rest of this chapter.

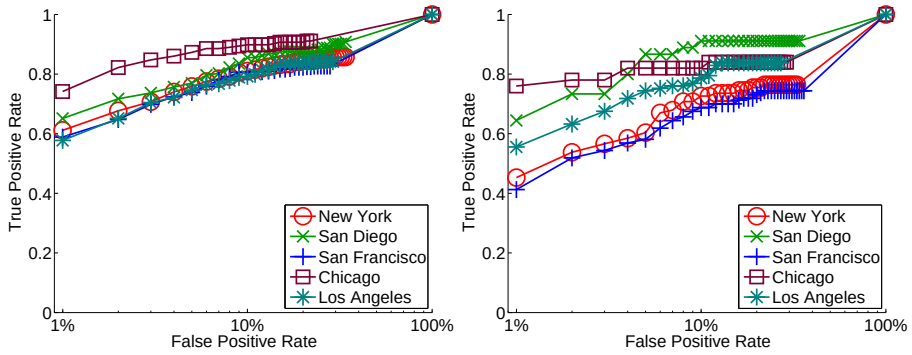
5.2.2 Similarity metrics

So far we have used the Cosine distance to compare histogram-based location fingerprints. Another possibility is to train a classifier and obtain a data-driven function to perform this match; however, the feature space for the classifier is too large and sparse, as we have more than 300,000 features (i.e., clusters). Furthermore it has been shown that if you train a neural network to match two discrete probability distributions using the squared error criterion, it learns to approximate the cosine distance [149]. We now proceed and examine other distance functions to compare the histograms. The statistics literature offers a variety of metrics for measuring similarity between two probability density functions P and Q [28]. We test a series of candidates, including Cosine and Jaccard from the Inner Product family; Euclidean and Manhattan from the Minkowski family; Hellinger from the Squared-chord family and Kullback-Leibler (KL) divergence from the Shannon Entropy family. We skip the details here for brevity but our analysis finds that except for the Euclidean distance others show comparable accuracy (which agrees with the previous mentioned result [149]). The Euclidean distance yields a much lower accuracy because it is sensitive to the absolute difference between two bins, in particular if it is large. In contrast, similarity metrics such as Cosine are sensitive to bins with non-zero values in both accounts, which better suit the matching of location fingerprints. Since the Cosine, Jaccard, and Hellinger distances have similar TPR in our experiments, we use Cosine for the remainder of our discussion.

5.2.3 Evaluation at scale

In this chapter, we evaluate the matching scheme only on the large-scale dataset described in §3.5. An evaluation over a small scale dataset does not make sense as most people will likely be from different parts of the world, hence matching accounts will be very easy to detect. Furthermore, the evaluation against human workers does not make sense because we are using features that are not visible to them. Given the large number of users that post in a particular region, even a small false positive rate could render an attack infeasible by returning a large number of false matched accounts. Hence, we typically tune the threshold so that it reports false positive rates of 1%. Furthermore, we do not have high expectations as in the previous chapter to fully automate the matching scheme; instead, we allow the matching scheme to return a *small* set of accounts that an attacker can check manually instead of only returning one account.

The previous sections show that representing the location of posts with clusters and identifying similar location fingerprints with the Cosine distance achieve the best tradeoff between TPR and FPR for identifying matching accounts in Flickr-Twitter and Yelp-Twitter.



(a) Flickr to Twitter (b) Yelp to Twitter

Figure 5.2: ROC curves for different urban areas for matching Flickr and Yelp users to Twitter users using clusters.

In this section, we discuss the overall accuracy of using location to identify matching accounts across social networks.

Figure 5.2a presents the accuracy of matching Flickr to Twitter accounts for each of the five regions we study, whereas Figure 5.2b presents the same results for Yelp to Twitter (the San Francisco results are the same as the Cluster curves in Figures 5.1a and 5.1b). For San Francisco, at 1% FPR, we have 60% TPR to match Flickr and Twitter accounts and 42% TPR for Yelp to Twitter. As a toy example, consider how these numbers apply to a small company of 10 employees, where all of them have Flickr accounts. Assume an attacker aiming to find their respective accounts on Twitter starting from a pre-filtered list of 100 candidate accounts. Among the total of 1,000 (Flickr, Twitter) account pairs, 10 are true matches and 990 are not. With 60% TPR and 1% FPR, our location-based attack will return a set of about 16 (Flickr, Twitter) account pairs that are possible match: 6 true matches (60% of the 10 users) and 10 false matches (1% of the 990 pairs). An attacker will need to sift through these 16 account pairs manually to identify the 6 true matches. Consider now the scenario used in our experiments, where an attacker wishes to identify the Twitter account of one given Flickr user in the San Francisco area using only location information. In this scenario, she has 60% chance of finding the Twitter account associated with the Flickr account by manually investigating the 750 most similar Twitter accounts, instead of searching all of the 75,474 San Francisco Twitter accounts. The precision for such operational point in less than 0.1%. This is acceptable here, however, because our focus is to see if we can return a small list of accounts that can be manually checked rather than returning only one account.

Figures 5.2a and 5.2b show that, although the shape of the ROC curves are similar across areas, the accuracy of the attack based on location is even higher for other areas than San Francisco. Our analysis of matching accounts in each area shows that most differences come from the fact that some areas such as San Francisco and New York have more users whose posts in Flickr or Yelp have no location in common with posts in Twitter. This observation is especially true when matching accounts from Yelp to Twitter. In San Francisco and New York, many people work and live away from the neighborhoods in the city center, where

people often go out. If a given user mainly tweets during her daily activities and not when she is out in restaurants, the location of her tweets will have little overlap with the places of restaurants she reviews on Yelp.

The comparison of Figures 5.2a and 5.2b shows that the accuracy of matching accounts from Flickr to Twitter is higher than from Yelp to Twitter. This difference comes, most likely, from the nature of these social networks. Users of Flickr and Twitter have more unique location fingerprints, because they can post or take a picture from anywhere, whereas Yelp reviews come from a large, but fixed set of locations (which correspond to the address of the reviewed restaurants). Indeed, §5.2.1 showed that only 77% of Yelp profiles are unique as opposed to 96% unique accounts for Flickr. Moreover, Flickr users tend to post from more locations. Finally, Flickr posts have more common locations with the corresponding Twitter account than Yelp posts do.

5.2.4 Implications

We now study the implications of these results for users. In particular, we investigate which properties of a user's location fingerprints can help prevent the attacker from successfully matching her accounts. Although the location fingerprint is a powerful feature for matching account of a single user across social networks, the results in Figures 5.2a and 5.2b show that we cannot identify all of our ground truth users with low FPR.

We define a new metric to split users in three groups according to the difficulty for an attacker to identify the correct account from the set of candidate accounts. Our metric determines the number of accounts in C^{area} with similarity scores higher than or equal to $s(a, \hat{a})$ (the similarity score of the true matching pair), which we term a user's *rank* for a given attack:

$$rank(a, \hat{a}) := \#\{b_i \in C^{\text{area}} : s(a, b_i) \geq s(a, \hat{a})\}.$$

Having $rank(a, \hat{a}) = 1$ means taht the matching is perfect and the attacker will pick the right account directly. Since a perfect matching is hard to obtain, we typically check if $rank(a, \hat{a}) \leq m$, i.e., the correct user is amongst the top m most similar accounts. For small m, an attacker can inspect that set manually.

We define as *vulnerable* the set of users with rank smaller 750 (this is equivalent of 1% FPR for users in San Francisco); *medium vulnerable* users with rank between 750 and 7,500; and *protected* as users with rank higher than 7,500. We check how many locations users in each of these groups post from, and how the number of common locations between the location fingerprints of the two matching accounts affects the rank. To investigate this we use all true account matches from Flickr to Twitter and Yelp to Twitter in the San Francisco area.

We find that protected users generally post from fewer locations, only 36% of protected users post from more than five locations, whereas 70% of vulnerable users post from more than five locations. Moreover, 95% of protected users have no common locations between the location fingerprints of their accounts across social networks; whereas all vulnerable and all medium vulnerable users have at least one location in common. These results suggest that one approach to protect against this attack is to minimize the number of

common locations across social networks. In fact, there is an 80% probability of the matching Twitter account to have a rank lower than 750 given a Flickr account when the two accounts have posts in three common locations (this probability is 69% to have rank lower than 375 and 47% lower than 50). If the user in the two accounts posted from more than six locations in common, then these probabilities increase to 85%, 76%, and 58% respectively.

Thus, the number of common locations across social networks is the most important property that makes users vulnerable to the account matching and even posting from a few common locations can already be enough to identify a small set of candidate matching accounts.

5.3 Timing fingerprint

Many third-party applications, in particular on mobile devices, allow users to automatically send updates to different social networks simultaneously. For example, when Instagram uploads to Flickr, it can automatically tweet a pointer to the photo. We exploit this behavior to match accounts based on the timestamps of such automated posts.

In any social networks the posts have upload dates attached to them, thus the availability of timing data is 100% for any social networks. However, the granularity of the upload date might not be sufficient in some cases. The timing data is very hard to impersonate as the attacker has to post at the same time as the user. In this section, we focus on Flickr and Twitter datasets because Yelp only gives the date, and not the exact time of each post. Generally, we aim to find accounts where one or more timestamps of Flickr photos match the timestamps of tweets. However, even for simultaneous posts, timestamps may differ slightly due to processing delays and desynchronized clocks. Hence, we consider a small time window around each timestamp to declare that the timestamp of the photo and that of the tweet match. The question is what an appropriate window size is. If the window is too small, we might miss true post matches, whereas a larger window may report many false matches.

To answer that question, we investigate the timestamp differences we see in our ground truth set, considering all the GT Twitter - Flickr pairs. For each account pair (a, b), $a \in Twitter$, $b \in Flickr$, where $user(a) = user(b)$, if the list of timestamps of posts in a is $tstmps(a) = \{t_1, t_2, t_3\}$ and in b is $tstmps(b) = \{T_1, T_2, T_3\}$ and $t_1 < t_2 < T_1 < t_3 < T_2 < T_3$, then we define the set of timestamp differences as the set of differences between timestamps of two consecutive posts on different social networks $td(a, b) = \{T_1 - t_2, t_3 - T_1, T_2 - t_3\}$. This set contains all the timestamp differences between posts on the two social networks potentially corresponding to the same content (e.g., a photo on Flickr and its link on Twitter). Note that in this example, if T_2 represents a Flickr image post, and t_3 the automated Tweet for the image post, then $T_2 - t_3$ represents the delay resulting from desynchronized clocks between Flickr and Twitter.

We investigate what is an appropriate threshold for this delay between posts across social networks so that we detect automated posts with low false positives. We manually investigate the content of posts with timestamp differences smaller than 30 s, as we consider 30 s

a safe upper bound for the maximum delay between automated posts. We can differentiate automated posts from the others as they have similar texts, and the metadata attached to tweets contains the name of the application that generated it. We find that most posts with timestamp differences larger than 5 s are not automated. We thus investigate TPR of applying thresholds ranging between 1 s to 5 s to match accounts.

We define the timestamp similarity score $s(a, b)$ between accounts a and b as the number of timestamp differences in $td(a, b)$ that are lower than a given delay threshold. We experimented with normalizing this value by the size of $td(a, b)$, but it did not improve the matching quality. We set FPR to 1% and measure TPR for thresholds ranging from 1 s (which includes all timestamp differences between 0 s and 1 s) to 5 s for accounts in Flickr GT^{SF} to accounts in Twitter C^{SF}. The 1 s threshold has the highest TPR (13%) while 5 s has the lowest TPR (12%). Hence, we use a 1 s threshold to match accounts based on timing.

The reason why the TPR is never higher than 13% is because only few users in our datasets use automated posts. When users do use automated posts, however, we often find a perfect match because timestamp is highly discriminable and consistent. In our dataset, all the users with more than four timestamp matches have a rank of one. This means that even if users only use automated posting for a brief period or just to test them, an attacker can match their accounts with a very high precision. As applications such as Instagram and Foursquare become more popular, we expect the timing information to allow matching even more pairs of accounts.

5.4 Language fingerprint

The final type of feature we consider for matching accounts is textual data. Textual data is always available when users are active in a social network, but there are users that never posted anything. On Twitter, 57% of users have at least one post, on Flickr, only 35% of users posted at least one photo, and on Yelp 52% of users posted at least one review. Textual data is also hard to impersonate as the attacker has to invest a significant amount of time to be able to adopt the writing style of the victim.

This approach builds on existing work that demonstrate that free-form text can exhibit characteristics sufficiently unique to identify an author [90]. To explore this potential, we examine matching Yelp reviews and Flickr photo descriptions with Twitter posts. We do not explore exact text matches because these are usually automated, and we capture these cases with timing matching.

For each Yelp account we consider the joint set of all the reviews; for each Flickr account we consider all the descriptions, tags and titles attached to a photo; and for each Twitter account we consider all the tweets with the exception of re-tweets and tweets that share links (as the text represents the title of the article in the link and not something that the user wrote). In the GT^{SF} and C^{SF} datasets, we find an average of 546 distinct words per Twitter account, 730 per Yelp account, and 516 per Flickr account. Note that these words may contain punctuation, and are case-sensitive. If we remove punctuation and disregard case, we have 394 distinct words per Twitter account, 218 per Yelp account and 480 per

Flickr account.

There are tens of millions of distinct words found in the posts of the three social network accounts, and many do not appear across all three accounts (only 200,000 of the roughly 40 million case-sensitive words in Twitter, along with punctuation, appear in Yelp and Flickr). Hence, it is important to first apply a pre-filter to reduce the number of words for several reasons: *(i)* to reduce the total number of words to a computationally-manageable size, *(ii)* to remove words that do not appear across multiple accounts, which would not significantly affect user account matching, but could de-emphasize the words that do significantly affect the matching, *(iii)* to remove common words (i.e. “and", “the") that may not be user-discriminative, and *(iv)* to account for case-sensitivity and punctuation. We recognize, however, that certain users may prefer certain combinations of case and punctuation in their writing style, potentially making case and punctuation user-discriminative features. After removing words that do not appear in both Yelp and Twitter, or Flickr and Twitter, we conduct two investigations based on the aforementioned points. First, we investigate the effects of punctuation and case-sensitivity of words. Second, we investigate the effect of removing the most frequent words between Yelp and Twitter, and Flickr and Twitter. The pre-filtering approach of removing punctuation and case-sensitivity, along with the top 1,000 most frequent words, gives the optimal results.

We build probabilistic language models for each Twitter user by constructing histograms of word unigrams, and normalizing them by the total word count per user such that each histogram represents a unigram probability distribution. We choose word unigrams as the unit for our models because our experiments show no further improvements when broadening to higher n-grams (i.e., multi-words). The reason why higher n-grams and other stylometry methods are less effective is because *(i)* the pre-filtering already removes what often links words together, and *(ii)* tweets consist mostly of keywords with fewer stylistic expressions. To measure the similarity between the Yelp and Twitter or Flickr and Twitter accounts, we accumulate the probabilities of each word in the Yelp or Flickr text from the language model of the Twitter account. This approach is a general version of the approach implemented by Stolcke [170].

In general, the language-based results are significantly worse compared to the location-based results, and achieve only a 6% TPR at 1% FPR for matching Yelp to Twitter accounts, and 10% for matching Flickr to Twitter accounts. The small TPR from Yelp to Twitter likely comes from the fact that the same user may adopt drastically different kinds of textual structure when writing Yelp reviews (typically complete paragraphs using words mostly found in the English lexicon) versus when tweeting (typically short sentences with fewer standard words). Correlating accounts from Flickr to Twitter is better than from Yelp to Twitter possibly because the short description of the photos may be more similar in style and topic to tweets than reviews.

5.5 Combining features

The previous sections discuss matching accounts across social networks with one *individual* feature at a time (location, timing, or language). We now use all three features *simultaneously*. The premise here is that combining the individual metrics should *(i)* achieve

stronger matching by leveraging their respective strengths, while *(ii)* making it harder for users to defend against such attacks. We then compare the results obtained by combining the three features with existing attacks that exploit screen names to match accounts.

5.5.1 Method

To assess the performance of combining multiple features to identify accounts that belong to one user across social networks, we use a binary logistic regression classifier [180]. For a pair of accounts in different social networks, the classifier takes as input the similarity scores of each feature (using the best settings for each feature as discussed in §5.2, §5.3, and §5.4) and predicts whether the pair of accounts is a match (i.e., belong to the same user) or not, as well as the probability of a match. We build classifiers for different combinations of features. For matching Yelp to Twitter, we build three classifiers using location and language (one classifier using location alone, another using language alone, and a third combing these two features). For Flickr to Twitter, we build six classifiers with different combinations of location, language, and timing.

We build our training and test sets from a dataset with all pairs of accounts in GT^{area} and C^{area}. As a result, we obtain an imbalanced training and test sets with fewer cases of account pairs that are true matches (only $|GT^{\text{area}}|$) and significantly more account pairs that are not matches ($|\widetilde{SN}^{\text{area}}| \times |GT^{\text{area}}| - |GT^{\text{area}}|$). This imbalance is representative of real-world datasets (where we expect the number of true matches to be orders of magnitude smaller than the total possible account pairs between two social networks). To account for the data imbalance and to correctly train the classifier we set the cost of mis-classifying the matching accounts inversely proportion to the proportion of matching accounts in the training dataset. We then evaluate the accuracy of each classifier using 10-fold cross validation.

5.5.2 Evaluation at scale

We compare the accuracy of classifiers using different combinations of features. We only present results for users in the San Francisco area, but the conclusions are similar for other areas. Table 5.1 presents the classification accuracy of each classifier for matching accounts from Flickr to Twitter and from Yelp to Twitter. This table also includes results for screen names for discussion in §5.5.3. The table presents the average TPR corresponding to 1% FPR across the ten runs of cross validation as well as the 95% confidence interval computed with vertical averaging [153].

TPR for classifiers based on individual features–location, timing, and language–are practically the same as the results in §5.2, §5.3, and §5.4, respectively. The small differences come from the fact that here we present results from the 10-fold cross validation, whereas earlier sections simply computed TPR for the entire dataset. The comparison between Loc and (Loc, Lang) when matching Flickr and Yelp with Twitter shows that language doesn't improve TPR when combined with location (in fact, it seems to reduce TPR slightly when matching Yelp to Twitter accounts). Hence, at low FPR, language doesn't help to identify more matching accounts than the ones location already identifies. We note that when we

Table 5.1: Comparison of the TPR for different classifiers at 1% FPR for matching Flickr and Yelp accounts to Twitter.

	TPR at 1%FPR	
Feature	**Flickr-Twitter**	**Yelp-Twitter**
Timing (T)	13±3%	-
Language (Lang)	10±3%	6±3%
Location (Loc)	60±6%	44±6%
Screen name (S)	77±3%	7±4%
Loc, Lang	60±6%	42±6%
Loc, T	70±3%	-
Loc, Lang, T	63±5%	-
Loc, S	86±2%	44±6%
Loc, Lang, S	86±2%	44±7%
Loc, T, Lang, S	88±2%	-

consider a higher than 10% FPR, adding language to location can increase the TPR by 10% for Flickr to Twitter matching. Timing, however, is more powerful than language. When we combine timing with location TPR improves by 7% over location alone. This increase shows that, when present, timing can very precisely identify true matches which helps improve the TPR especially for low FPR. The combination of location, language, and timing increases the TPR over the entire range of FPR. Timing improves TPR when FPR is low, whereas language helps when FPR is high. At 1% FPR, the highest TPR we achieve for matching Flickr accounts to Twitter is 70% when we combine location and timing. The highest TPR for matching Yelp accounts to Twitter is 44% when using location alone. With the best combination, for the Flickr to Twitter matchings, 17% of the ground truth users can be identified in the top 10, 27% in the top 50 and 33% in the top 100, while for the Yelp to Twitter matchings, 1% can be identified in the top 10, 4% in the top 50 and 7% in the top 100.

5.5.3 Comparison with screen name matching

This section compares the accuracy of our classifiers, which only use features extracted from innocuous user activities, with the state-of-the-art technique to match accounts across social networks: matching based on the screen name. We compute the similarity between two screen names using the Jaro distance [36], which is the state-of-the-art distance in record linkage to measure the similarity between two names. Perito et. al [148] showed that the Jaro distance performs well to match screen names across different social networks as well.

Table 5.1 also shows the average TPR at 1% FPR for matching accounts from Flickr to Twitter and from Yelp to Twitter based on screen names. We first note that screen names alone achieve 77% TPR for matching accounts from Flickr to Twitter. When matching Yelp accounts to Twitter, however, screen names only reach 6% TPR, which is lower than any of the other features we consider. Screen names achieve high accuracy to match accounts

in Flickr to Twitter, because many users use the same or similar screen names on these two social networks. On the contrary, Yelp users often select as screen names just their first name and the initial of their last name or some alias, reflecting their desire to maintain their reviews pseudo-anonymous.

When we compare matching based on screen names with the combination of location, timing, and language for matching Flickr to Twitter accounts we observe that the TPR of screen names is higher than that of the combination of the three other features together. If we combine screen names with the other three features, we obtain even better results (TPR increases to 88%). Screen name is clearly a powerful feature to match Flickr and Twitter accounts today, as we showed in Chapter 4. We should not forget, however, that it is easy for users who want to hide to obfuscate their identity by simply selecting different screen names. So, the accuracy of screen names to match accounts across social networks can decrease drastically as soon as users realize that correlating information across social networks represents a real threat to their privacy.

We further check whether we can match, by exploiting the user activity, users that maintain different personas on Flickr and Twitter that we cannot match only using screen names. Our results show that we can match, by exploiting user activities, almost 50% of the users that cannot be matched using only screen names.

Given that users in Yelp select screen names that do not reveal their identity, when matching Yelp to Twitter accounts, location alone achieves a much higher TPR than screen names (44% vs. 7% TPR for 1% FPR). Screen name does not even help increase TPR when combined with location (see Table 5.1). In fact, out of all detected matches between Yelp and Twitter 78% are only identified by location. Our approach of using features based on innocuous user activity should always work better than screen names for social networks like Yelp, where users do not use their true identity.

5.6 Discussion

Our results in Sections 5.5.2 and 5.5.3 demonstrate the power of our matching scheme, which provides a high match quality even when tuned for the low false positive rates that such needle-in-a-haystack challenges require. We now discuss our results further in terms of realistic attack models, availability of the features we exploit, and potential defenses users may take.

Attack model Given our matching accuracy we see two attack models as particularly relevant. First, our matching scheme allows an attacker to find further accounts that belong to a specific target individual by quickly winnowing down from a large initial starting set to a much smaller number of candidate accounts suitable for manual inspection. While she may still need to invest non-trivial effort into the final verification step, the automatic prescreening nevertheless enables an attack that would not be feasible at all otherwise.

Second, it is possible to attack a group of people rather than a specific individual. An example here is finding employees of a large company that might be vulnerable to bribery (maybe because of gambling habit that indicates money problems) or extortion (maybe because

of a medical condition, or an affair). In such a model, the attacker would start with the set of company employees, e.g., on LinkedIn; match them with other social networks, and potentially further public records, to collect more personal information; and eventually match all that to relevant target sites such medical forums, addiction advise networks, or dating sites.

Feature availability Most social networks provide the features our attacks exploit. For example, Facebook posts carry timestamp information, and Facebook check-ins come with location information. Likewise, both Google+ posts and Youtube videos make the upload time available, and either can include location in its metadata. However, even if an attacker does not have direct access to some of the features on a particular network, often she might still infer it from the posted content itself. For example, with LinkedIn we could get a suitable location fingerprint from the places somebody has previously worked. More interestingly, the multimedia community is developing a range of approaches to accurately determine location information from content, such a photos, videos, and meta-tags [2, 5, 33, 39, 53, 91]. Currently, only 1% of all the tweets have geotags and only 5% of the active Twitter users have at least one post geotagged. Since our results show that we only need coarse-grain location information to match users' accounts, we believe that these techniques can be reliably used in our attack to infer the location of posts when geotags are not available.

One can also collect the necessary features outside of social networks. A particular privacy threat concerns mobile applications with access to a user's current location. If that information is provided back to the application developers (as is typical, for instance, for map and search services), they can identify users by associating corresponding location fingerprints with social identities. As we have seen, even coarse locations, like zip codes, convey sufficient information, and hence simple privacy-conscious schemes, such as blurring the resolution, will not protect from such attacks.

While we discuss just three specific features for account matching, there are others that an attacker can exploit in a similar way. In particular, content may indirectly reveal further personal information that can help guide the matching process, such as "Happy Birthday" greetings from friends that reveal a person's birthday, even if she does note make the date itself publicly available. Another possibility concerns matching based on interests as inferred from the context and the content one "likes".

Defenses As we indicated earlier, it remains hard to defend against de-anonymization attacks that exploit information so intrinsically, and ubiquitously, linked with content. However, there are some countermeasures that can make such attacks less likely to succeed, in particular, from the perspective of an individual who is part of a larger group of potential victims[2].

As a possible defense against our timing matching, applications could slightly delay automated posts, introducing random jitter that makes it harder to find suitable thresholds separating them from manually issued content. Our analysis suggest that a variation interval in the order of 10 s of seconds would prove more than sufficient. We suggest two

[2]"I don't need to outrun the bear; I just need to outrun *you*."

strategies to avoid becoming vulnerable to location matching. As the more obvious one, it clearly helps not to post to separate social networks from the same location because that's what the attack keys on (remember from §5.2.4 that 95% of protected users do not have any common location between their accounts). A more interesting, and less drastic, countermeasure exploits the fact that one can *correct past mistakes* (i.e., already sharing many locations between accounts) by adding further unrelated locations to the mix. Doing so effectively blurs the link to other networks by adding noise. For example, for a vulnerable user (with a rank less than 750, see §5.2.4), that has 5 common locations between his accounts, to become medium vulnerable or protected he needs to add respectively around two or seven unrelated locations on one social network.

Finally, we note that defending against account matching generally gets more difficult as the attacker combines further features, hence making the analysis more robust against inconsystencies in any individual feature. There is a fundamental tradeoff here in that *any* useful information that a user publishes will potentially increase the chance of a successful matching attack.

5.7 Summary

This chapter presents a powerful set of techniques for matching user accounts across social networks, based on otherwise innocuous information like location and timing patterns. Our approaches work independent of standard privacy measures, such as disabling tracking cookies or using anonymizing proxies. For our study, we collected data from the three social networks Twitter, Flickr and Yelp, including extensive ground truth of 13,629 users with accounts on both Twitter and Flickr and 1,889 users with accounts on both Twitter and Yelp. Our results go beyond prior work by not relying on more obvious, user-chosen information (e.g., screen names [148]) and by evaluating the power of the matching in real-world scenarios. We show for example that, using the location information, we can match 60% of Flickr accounts with their corresponding Twitter accounts, while only introducing a small percentage of falsely matching accounts. Moreover, our results show that we only need coarse-grained location information to match a relevant number of accounts. Combining all features together gives comparable results with matching on screen names for Flickr to Twitter matching, and can identify 37% more matching accounts for Yelp to Twitter matching.

In the process of matching accounts based on locations, we propose a way to aggregate/represent the location profile of an individual such that it is specific enough to make him unique and general enough to approximate areas of interests rather then meaningless locations.

The privacy implications of our results are two-fold. First, we point out that it is the *aggregate* set of a user's complete online footprint that needs protection, not just content on individual social networks. Second, we find that it is hard to defend against such attacks as the information that enables them often comes intrinsically with the very activity one *wants* to publish.

While our work examines a specific set of websites and matching techniques as case studies,

it demonstrates the broader potential, and risk, of cross-site matching. Our approaches remain conceptually simple yet we expect that, soon, more sophisticated variants will emerge for exploiting the increasing volume of innocuous user information that websites now offer via convenient APIs. In particular, we anticipate that automated content analysis technology—such as face recognizers and natural language processing—will enable matchings more powerful than what we demonstrate here. As such, we see our contribution less in the specific performance numbers that our experiments yield—which will always vary between users, features, and social networks—but primarily in pointing out that identifying users by their posting activity indeed poses a real threat. From a research perspective, we encourage our community to devise novel privacy protections that take such threats into account and, where hard to prevent, at least support users in understanding their vulnerability.

Chapter 6

Detection and Characterization of Impersonators

In this chapter we use the tools developed in the previous chapters to study impersonation attacks in social networks. We first propose a technique to detect impersonating accounts and then we characterize impersonating accounts on Twitter. Our technique to detect impersonating accounts works in two steps. We first identify all the accounts in a social network that portray the same person; by building a rule-based matching scheme that exploits findings from previous chapters and emulates human judgement in deciding whether two accounts portray the same person. We then build a classifier that takes pairs of accounts portraying the same person and outputs whether one account is impersonating the other or both accounts are real (i.e., managed by the same person).

Traditional methods to detect fake accounts perform poorly for detecting impersonating accounts. Many techniques detect fake accounts relying on ground truth from humans to build classifiers that do the detection automatically [185]. We show that AMT workers are easily tricked into thinking the impersonating accounts are real. Thus, if the ground truth is flawed, the classifiers will do a poor job at detecting impersonating accounts. Furthermore, traditional methods to detect fake accounts exploit features that characterize single accounts. We show that, to detect impersonating accounts, we have to build methods that exploit features of pairs of accounts instead of single accounts.

Our analysis of 5,693 impersonating accounts on Twitter shows that the attackers target not only celebrities, but also ordinary Twitter users. Furthermore, attackers create accounts that impersonate users mostly to evade the Twitter fake account detection systems rather than to build social engineering attacks. Even if the victim is not harmed directly, she can still suffer indirectly because the attacker alters her online image. For example, we found in our datasets a technology company that was impersonated and the impersonating account retweeted celebrities and posted tweets such 'I think I was a stripper in a past life'. This is clearly not the image the company wants to promote. Our findings reveal a new type of impersonation attacks that can impact negatively the online image of any user.

6.1 A framework to detect impersonating accounts

In this section we set the general framework for detecting impersonating accounts. We first define the problem and we give the intuition behind our approach, we then describe the features and the dataset we use.

6.1.1 Problem definition and approach

Given a person p, we want to find the accounts that impersonate p in the set of accounts in a social network SN. *Impersonating accounts* are the accounts that pretend to be p but that are not managed and do not have the legal authorization to act on behalf of p. We denote an impersonating account as $\hat{a} = impersonator(p)$. Some users maintain multiple accounts in a social network – one account can be professional and the other personal – thus there might be multiple real/legal accounts that portray p. We refer to these accounts as the *avatars* of p. We denote an avatar account as $\bar{a} = avatar(p)$.

We approach the problem of detecting impersonating accounts by splitting it in two distinct parts:

1. Detect all the accounts that portray the same person p inside a social network, $CI(p)$.
2. Split $CI(p)$ in avatar and impersonating accounts. $CI(p)$ can contain multiple avatar and multiple impersonating accounts which makes the problem more challenging.

For the first step, we leverage the techniques that we developed for matching the accounts of the same person across social networks, while for the second step we take a supervised approach to distinguish between impersonating and avatar accounts.

6.1.2 Dataset

We base our analysis on Twitter. Some social networks such as Facebook and LinkedIn clearly stipulate in their Terms of Use that users should only have one account in a social network, while other social networks such as Twitter allow users to have multiple accounts. We chose to detect impersonating accounts on Twitter because it is more challenging since there can be more than one real account that portray a person. The techniques we develop apply also to other social networks such as Facebook and LinkedIn.

Our impersonator detection technique requires a dataset of accounts that can potentially portray the same person in a social network. To generate the dataset, we first choose a set of accounts in Twitter, $\mathbf{S}$, and then, for each account $a \in \mathbf{S}$, we collect a candidate set of accounts $C(a)$ that can potentially represent the same person portrayed by a and can potentially be impersonating accounts. To generate $C(a)$ we use the Twitter API to collect all the accounts that have the same or similar names as a.

Since the number of impersonating accounts on Twitter should be small compared with the number of good accounts, we generate the initial list of Twitter accounts, $\mathbf{S}$, in an opportunistic way. Starting with a seed impersonating account, we do a breath first search crawl on the followers of the seed account. Our intuition is that we might find other

impersonating accounts in the close network of an impersonating account. The seed account was an impersonator of Nick Feamster which we incidentally stumbled upon. We collected 100,000 accounts with the breath first search crawl on the followers of the seed account, which we call Initial Accounts. To cover a larger part of the social graph the breath first search crawler does not crawl users with more than 10,000 followers.

For each account a in Initial Accounts, we gather a set of 40 accounts, $C(a)$, with similar names with a, using the Twitter API. We call the resulting 4 million pairs of accounts (a, b) where $b \in C(a)$ and $a \in$ Initial Accounts the Initial Pairs. Table 6.1 summarizes the datasets we use for this section. We will explain later the rest of datasets in the table.

Table 6.1: Datasets for studying and analyzing impersonating accounts.

Initial Accounts	100,000
Impersonation Accounts	5,693
Avatar Accounts	4,423
Initial Pairs	4,000,000
Same Person Pairs	7,967
Mixed Pairs	4,423
Avatar Pairs	1,332
Impersonation Pairs	635

For each account, we use the Twitter API to collect profile information, the timeline, the list of followers and friends, as well as the list of tweets the user added to this favorites[1] and the Twitter users he mentions[2] in his tweets.

6.1.3 Features

To distinguish between impersonating accounts and avatar accounts we use gather features that characterize a single account and features that characterize pairs of accounts.

Features to characterize an account: We gather two categories of features to characterize an account: features that describe the activity of a user and features that describe the popularity of a user.

From the profile information of accounts we extract the following features that describe the activity of a user: *creation date of the account, timestamp of the first tweet, timestamp of the last tweet, number of followers, number of friends, number of tweets, number of retweets,* and *number of tweets favorited.* Users can mention other users in their tweets or bios using conventions such as @oanagoga. We define the *number of mentions* as the number of unique users mentioned by an account. We consider these features because they are markers of 'good' behavior and spikes in any of the features could rise suspicion (e.g., having more retweets than normal users could indicate that an account is selling retweets).

[1] On Twitter users can either re-tweet or favorite a tweet they like.
[2] On Twitter users can mention other users through convention such as @oanagoga.

The more influential a user is, the more likely it is that he is a real and not an impersonating account. Twitter allows users to create lists to split followers in different categories. For example, users can create a list for news tweets where they can put @nytimes and @cnn. The *number of lists where the user appears* counts the number of times a given users appears in the lists of other users. The *klout score* [92] is a wide used score that measures the social influence of an account. The klout score has values between 1 and 100, 100 meaning that the account is very influential and 1 that the accounts is not influential.

Features to characterize pairs of accounts To characterize pairs of accounts, we use three types of features: features that measure the similarity between two accounts, features that characterize the interactions between two accounts, and features that characterize the time overlap between two accounts.

To measure the similarity between *names, screen names, locations, profile photos, friends*, and *followers*, we use the metrics from §4.1.2. To measure the *bio* similarity we simply count the number of common words between the bios of the two accounts. This is a common technique in entity matching to measure the similarity between records and has good results in practice [34]. We leave more sophisticated techniques based on the popularity of words as future work. To achieve better accuracy, we pre-process the bios and remove the stop words, i.e., the most frequently used words such as the, is, at, and which. We obtained a list of stop words in 19 languages from Ranks NL project.[3]

Besides the similarity between profile attributes we also measure the similarity between the *interests* of two accounts. We use the algorithm proposed by Parantapa et al. [21] to infer the interests of a user, by exploiting who the user follows. The algorithm returns a list of interests that characterizes an account. Some interests are more common than others. To give more weight to the least common interests we borrow the *term frequency - inverse document frequency (TF-IDF) [86]* weighting scheme from the information retrieval field to weight interests proportionally to their popularity. We apply TF-IDF as follows: for each interest in an account's interests lists, TF represents the frequency of the interest in the interests lists, and IDF represents the number of times the interest appears in other interests lists of accounts in Initial Accounts. Then the weight of the interest is $TF/log(IDF)$. We finally use the cosine distance to measure the similarity between the two weighted lists of interests.

We add a number of features that represent the interactions between the two accounts portraying the same person, (a, b). The intuition behind these features is that avatars managed by the same person will likely interact, while impersonators will never interact with their victims as they do not want to be discovered and reported. Users can mention other users through conventions such as @oanagoga. For example a user might have in his bio the following text: "Hello, this is my personal profile. For professional content please go to @oanagoga_lip6". To catch such behavior, we define a *mention match* when a mentions b or b mentions a in his tweets or bio. Similarly, we define a *retweets match* when a retweets a tweet of b or vice versa, a *friends match* when a is friend with b and a *followers match* when a follows b or vice versa.

[3] http://www.ranks.nl/stopwords/

We also add features related to the time overlap between two accounts: *time difference between the creation dates*, *time difference between the last tweets*, *time difference between the first tweets* and whether one account stopped being active after the creation of the second account, we denote this feature as *outdated account*. We finally consider the difference between the klout scores of the two accounts, the *klout score difference*. A small klout score difference could be indicative of avatars of the same person while a large klout score difference could be indicative that one account is impersonating the other.

6.2 Detection of accounts that portray the same person

In this section we discuss matching schemes to detect accounts that portray the same person inside a social network. We start by describing the problems with directly applying the three-step matching scheme from Chapter 4 to our current scenario. We then propose a better suited matching scheme and we evaluate it using human workers.

To detect the accounts that portray the same person, we exploit the following public attributes of accounts: real name, screen name, profile photo, bio, and location. We do not use other features such as the number of common friends or location and language fingerprints because we want to detect all the accounts that *portray* the same person and not all the accounts *managed* by the same person. Consequently, we need attributes that capture the appearance of an account and not the behavior of a user.

6.2.1 Naive approach

We first investigate whether we can use the three-step matching scheme proposed in Chapter 4 to detect all the accounts that portray the same person in a social network. The three-step matching scheme achieves a high precision to output a matching account because it assumes there is only one matching account in a social network (recall that, the Guard abstains instead of returning a matching account whenever it stumbles on cases where there are multiple possible matching accounts but there is no single account that stands out). This assumption, however, does not hold in our scenario as we focus on detecting cases where there are multiple accounts that portray the same person inside a social network. The first two steps of the matching scheme, the Filter and the Disambiguator, however, do not make this assumption. Thus, we could potentially remove the Guard and only use the first two steps of the matching scheme to identify all accounts portraying a user. This approach, however, is sub-optimal because we tuned the parameters (e.g., the weights given to each attribute and the thresholds for the probability of two accounts to match) of the Filter and Disambiguator to detect the matching account of a user across different social networks.

6.2.2 Single-site matching scheme

Since we do not have ground truth of accounts that portray the same person inside Twitter, we cannot take a supervised approach to build the matching scheme. We can build,

however, a new matching scheme based on the findings of Chapter 4. More precisely, we exploit the following two observations:

- When we asked AMT workers why they think two accounts correspond to the same person they usually said that their names are similar and some other piece of information is similar such as their photo, their location or their bio, see §4.2.5.
- We know the similarity thresholds after which humans consider two values of an attribute to be consistent, see §4.1.2 (we obtained these thresholds from an AMT experiment that we did to measure how consistent users are in providing the same value for an attribute across social networks). The thresholds are chosen such as 90% of the attributes identified as consistent by humans have a similarity above the threshold and less than 10% of attributes identified as not being consistent have a similarity lower than the threshold.

We combine these two observations to build two new rule-based matching scheme that emulates human judgment. In entity recognition, it is a well known and widely used technique, to define rule-based matching schemes built by domain experts [34].

Loose bounds matching scheme: A pair of accounts is in the loose bounds whenever the real names or screen names are consistent (the similarity of names is higher than the consistency threshold identified by humans) and some other feature such as the locations, bios, or profile photos are also consistent:

$$(s_{name} > thr_{name} \mid s_{screen-name} > thr_{screen-name}) \;\&\; (s_{location} > thr_{location} \mid s_{bio} > thr_{bio} \mid s_{photo} > thr_{photo}) \quad (6.1)$$

Our intuition is that all the accounts that portray the same person should fall in the loose bounds, i.e., it is very unlikely that two accounts that portray the same person do not have any consistent attribute. This is especially true for an impersonator whose goal is to be as similar as possible to the victim account. Thus, while it is likely that this matching scheme catches all the accounts that portray the same person, i.e., it has a very high true positive rate, it also likely catches accounts that do not portray the same person, i.e. it has high false positive rate. For example, there can be people that have the same name and live in the same city.

Tight bounds matching scheme: A pair of accounts is in the tight bounds whenever the names or screen names are consistent and the bios, or profile photos are also consistent (we leave out the location for this bounds).

$$(s_{name} > thr_{name} \mid s_{screen-name} > thr_{screen-name}) \;\&\; (s_{bio} > thr_{bio} \mid s_{photo} > thr_{photo}) \quad (6.2)$$

The tight bounds is more restrictive than the loose bounds matching scheme and likely misses some accounts that portray the same person, i.e., it has a lower true positive rate. However, our intuition is that all the accounts that fall in the tight bounds portray the same person and there are no accounts that portray different persons that fall in the tight bounds, i.e., it has a low false positive rate. Put differently, having a consistent name and bio or a consistent name and profile photo is so unique that there cannot be another different person that has the same values.

We evaluate the loose bounds and the tight bounds matching scheme on all the pairs of accounts in Initial Pairs. For all the accounts a in Initial Accounts, we compute the similarity between real names, screen names, profile photos, bios, and locations for all pairs of accounts (a, b) where $b \in C(a)$. We then test how many pairs of accounts (a, b) fall in the loose and in the tight bounds. Out of the 4 million possible pairs of accounts, there are 27,582 pairs that fall in the loose bounds and 7,967 pairs that fall in the tight bounds. We denote the pairs of accounts that fall in the tight bounds the Same Person Pairs. There are accounts a for which there are multiple pairs of account (a, b) where $b \in C(a)$ that fall in the loose or tight bounds. There are 13,007 accounts in Initial Accounts that have at least one pair in the loose bounds and 5,897 accounts that have at least one pair in the tight bounds. We evaluate the accuracy of the matching schemes next.

6.2.3 Evaluation using AMT workers

We want to evaluate how many pairs of accounts from the loose and tight bounds matching schemes actually portray the same person. Since we do not have ground truth, we setup an AMT experiment to estimate the true positive rate and the false positive rate of the matching schemes. We assume the true positive rate of the loose bounds matching scheme is close to 100% and we only measure the precision. We do not to evaluate the true positive rate of the loose bounds matching scheme because it needs a very expensive AMT experiment.

In Chapter 4, we saw that humans are good at identifying whenever two accounts portray the same person. We randomly select 220 pairs of accounts from the loose bounds out of which 62 fall in the tight bounds. In each assignment, we give AMT workers two links corresponding to the two Twitter accounts and we ask them to choose between three options: *'the accounts belong to the same person'*, *'the accounts do not belong to the same person'*, or *'cannot say'*. For each assignment we ask the opinion of three different AMT workers. We say that we have a *full agreement* when all the AMT workers chose the same answer, a *majority agreement* when at least two AMT workers chose the same answer.

Table 6.2: Number of accounts detected by AMT workers as portraying the same person in the tight bounds and in the loose bounds.

	Total	Detected as same person
Loose bounds	220	94
Tight bounds	62	61

Table 6.2 shows the number of pairs of accounts AMT workers identified as belonging to the same person with majority agreement. The results show that 43% of the accounts that

fall in the loose bounds belong to the same person, while 98% of accounts that fall in the tight bounds belong to the same person. This validates our intuition that all the accounts that fall in the tight bounds portray the same person. Furthermore, 65% of accounts that belong to the same person fall in the tight bounds.

Consequently, the tight bounds matching scheme has a 65% true positive rate (assuming the loose bounds matching scheme has a 100% true positive rate) for a 2% false positive rate. The true positive rate could be slightly overestimated as there might be accounts that portray the same person but do not fall in the loose bounds. Although we cannot estimate how many accounts fall in this category, our intuition is that there should not be many. We picked the similarity threshold for consistent values of an attribute such that 90% of the values identified by AMT workers as consistent to fall above the threshold. Thus, there can be 10% of consistent values we miss. Thus, in the worst possible case we miss 20% of accounts that portray the same person with the loose bounds. For the rest of the chapter we use the tight bounds matching scheme to identify accounts that portray the same person. We discuss next how we can identify impersonators in the list of accounts returned by the tight bounds matching scheme.

6.3 Detection of impersonating accounts

In this section, we show methods to detect impersonating accounts out of the accounts that portray the same person. If the social network restricts the number of accounts that each person can have to one, then the problem of detecting the impersonating accounts out of a list of accounts that portray the same person is easier. We can, for example, detect the impersonating account as the one who was created more recently or the account that has the lowest social influence (i.e., klout score). In some social networks, however, there can be multiple accounts that are avatars of the same person, hence the impersonation account detection is more challenging. We approach the problem in a supervised way and we study two classification techniques to detect impersonation accounts, one based on features of accounts alone and one based on features of pairs of accounts.

6.3.1 Ground truth

Obtaining ground truth data about impersonators is a challenging problem. Since we are the first to do an impersonator study, there is no dataset with ground truth about impersonating accounts. We managed, however, to gather a satisfying set of ground truth from two sources: twitter suspended accounts and accounts that interact with each other.

Accounts suspended by Twitter: Three months after we first crawled the 7,967 pairs of accounts in the tight bounds (the Same Person Pairs) we re-crawled them and 64% of the pairs had at least one account suspended. Hence, the suspended accounts were likely impersonating accounts. We denote the accounts suspended by Twitter the Impersonation Accounts and the corresponding accounts that are not suspended the Avatar Accounts. We use these datasets as ground truth for detecting impersonators. We also denote the pairs of accounts with one account suspended and one not the Mixed Pairs

and the accounts with both accounts suspended the Impersonation Pairs. Table 6.1 shows the number of pairs in each category.

We suspect that Twitter suspends these accounts because they are involved in illegal activities such as fake followers or fake retweets rather than being detected as impersonating accounts. Currently, there is no framework to automatically detect impersonating accounts and the only solution is for the victims to manually report the accounts who are impersonating them [75]. We validate this assumption in §6.5.

Accounts that interact: We previously discussed that is very likely that if two accounts interact/mention each other they are avatars managed by the same person and not impersonating accounts. We found 1,332 pairs of accounts in the tight bounds that either have a mention match, retweets match, friends match or followers match. We denote the pairs that mention each other the Avatar Pairs. To validate that these accounts are actually representing avatars of the same user and not impersonators we randomly picked 100 pairs of accounts and we manually investigated them [4]. In only three pairs out of 100 there was an impersonating account. These pairs corresponded to cases where one user complained that there is someone else impersonating him. Since the majority of pairs do not contain impersonating accounts we consider them as ground truth for pairs of avatar accounts.

Our ground truth has limitations: the accounts suspended by Twitter might miss certain types of impersonating accounts (that do not engage in fake retweets or fake followers scams) and our ground truth for accounts that interact might also miss some types of avatars that do not mention each other. An alternative way to get ground truth is to ask AMT workers whether they think two accounts are avatars of the same person or one account is impersonating another. We will see, however, in §6.4 that AMT workers are not reliable enough to be used as ground truth. We leave as future work to study how we can combine different ways to gather ground truth to obtain more representative datasets. We envisage in the future to try to directly ask Twitter users whether an account is impersonating them or if it is an avatar. Our current ground truth, however, is still useful to detect and characterize impersonators.

6.3.2 Methods to detect impersonating accounts

We approach the problem of detecting impersonating accounts in two ways. We first present a traditional approach that builds a classifier based on features of single accounts. We then present a new approach that builds a classifier based on features of pairs of accounts instead of accounts alone.

Account-based approach: The traditional approach to detect fake accounts in a social network is to analyze the features of a single account with the help of human workers or matching learning algorithms and decide whether the account is real or fake. The drawback

[4] We are trained at identifying impersonating accounts after long hours looking at the data.

of such approach is that it can potentially miss impersonating accounts because they look real when looking at them in isolation.

IMPERSONATION ACCOUNTS and AVATAR ACCOUNTS contain ground truth of impersonating and avatar accounts. We build a binary classifiers, which we call ACCOUNTS CLASSIFIER, that distinguishes between impersonating and avatar accounts. We select 1,000 random accounts from IMPERSONATION ACCOUNTS and 1,000 random accounts from AVATAR ACCOUNTS to build the training set. We use a Naive Bayes classifier and we train it with all the features characterizing an account that we presented in §6.1.3. We use 10-fold cross validation to train and evaluate the classifier.

To evaluate the classifier, we focus on small false positive rates as it is important when detecting impersonators not to label an avatar account as an impersonating account. The ACCOUNTS CLASSIFIER only achieves a 6% true positive rate for a 1% false positive rate for detecting impersonating accounts and a 93% true positive rate for a 1% false positive rate for detecting avatar accounts. Thus, the account-based approach is not able to accurately identify impersonating accounts. The feature that are the most indicative of an impersonating account is a recent account creation date and the feature that is most indicative of an avatar is a high klout score.

Pair-based approach: A more suited approach to detect impersonating accounts is to analyze pairs of accounts instead of analyzing single accounts in isolation. Given two accounts a and b that portray the same person there are multiple possibilities: (1) both accounts are avatars, $(avatar(p), avatar(p))$; (2) both accounts are impersonators, $(impersonator(p), impersonator(p))$ – attackers might create multiple accounts using the same victim account information; and (3) one account is an avatar and one account is an impersonator, $(avatar(p), impersonator(p))$. The pair-based approach first detects whether a pair of accounts corresponds to one of the three categories by analyzing features that characterize pairs of accounts. Whenever we have a pair of accounts $(avatar(p), impersonator(p))$, we can further check the creation date and the klout score of the accounts to detect which account is the impersonator and which account is the avatar. The avatar account usually has a klout score higher than the impersonating account and the creation date of the impersonating account is after the creation date of the real account. In the MIXED PAIRS dataset, all but one suspended account have the creation date after the creation date of the unsuspeded accounts. The only unsuspended account that has the creation date after the suspended account corresponds to a case where a spammer created multiple fake accounts that have the same name, profile photo and bio.

MIXED PAIRS and AVATAR PAIRS contain ground truth on pairs of accounts that are avatars and pairs of accounts that contain an impersonating account. We use these datasets to train a binary classifier to distinguish between $(avatar(p), avatar(p))$ pairs of accounts and $(avatar(p), impersonator(p))$ pairs of accounts. We call this classifier the PAIRS CLASSIFIER. We build a training set with 1,000 random samples of $(avatar(p), avatar(p))$ pairs and 1,000 random samples of $(avatar(p), impersonator(p))$ pairs. We use all the features that characterize pairs of accounts described in §6.1.3 besides mention, retweets, friends, and followers match because we already used these features to build the ground truth. We use 10-fold cross validation to train and evaluate a Naive Bayes classifier.

The Pairs Classifier achieves a 75% true positive rate for a 1% false positive rate to identify $(avatar(p), impersonator(p))$ pairs of accounts and a 85% true positive rate for a 1% false positive rate to identify $(avatar(p), avatar(p))$ pairs of accounts. Thus, the accuracy of the pair-based approach is much higher than the accuracy of the account-based approach.

The features that are the most indicative of pairs of accounts $(avatar(p), avatar(p))$ are the similarity between the accounts' interest and the number of common friends and followers. The most indicative features of pairs of accounts $(avatar(p), impersonator(p))$ are the similarities between profile attributes, i.e. the more two accounts look alike the more likely is that one is impersonating the other. Also the time difference between the creation date of the two accounts is a powerful indicator that one account is impersonating the other.

6.3.3 Evaluation over unlabeled pairs of accounts

We test the Pairs Classifier over 1,578 pairs of accounts in the Same Person Pairs that are not in the Mixed Pairs, Avatar Pairs or Impersonation Pairs dataset. Pairs Classifier identifies 1,131 pairs of accounts $(avatar(p), avatar(p))$ and 356 pairs of accounts $(avatar(p), impersonator(p))$. Thus, the Pairs Classifier can be used to detect more avatars and impersonating accounts with a high accuracy.

6.4 Detecting impersonating accounts using humans

To understand how well humans detect impersonating and avatar accounts, we setup two AMT experiments. More specifically, we want to evaluate the true positive rate and the false positive rate of humans to detect impersonating and avatar accounts. In all the experiments we ask the opinion of three AMT workers and we report the results for majority agreement. We detail next the experiments we did and the results.

Single account: To evaluate the true positive rate and false positive rate of humans we select 50 impersonating accounts and 50 avatar accounts from Same Person Pairs which we manually validate. In each assignment, we give AMT workers a link to a Twitter account and we ask them to choose between three options: *'the account is real'*, *'the account is fake'* and *'cannot say'*. Table 6.3 shows the results. AMT workers correctly detected only 9 out of 50 impersonating accounts, however they detected 48 out of 50 avatar accounts. Thus, in this scenario, AMT workers have only a 18% true positive rate for a 82% false positive rate to detect impersonation accounts and a 96% true positive rate for a 4% false positive rate to detect avatar accounts.

Pairs of accounts: Hoping to achieve a better accuracy, instead of showing AMT workers one account, we show them two accounts that portray the same person: either the impersonating and the corresponding avatar account, or two avatar accounts. We picked the same 50 impersonating accounts (and their corresponding avatars) and the same 50 avatar accounts (and their corresponding other avatars) as in the previous experiment. In

Table 6.3: Number of impersonating and avatar accounts detected by AMT workers, based on single account features, out of the true impersonating and avatar accounts.

	True impersonating	**True avatar**
Detected impersonating	9	2
Detected avatar	41	48

Table 6.4: Number of $(avatar(p), avatar(p))$ and $(avatar(p), impers(p))$ pairs of accounts detected by AMT workers out of the true $(avatar(p), avatar(p))$ and $(avatar(p), impers(p))$ pairs of accounts.

	True $(avatar(p), impers(p))$	**True** $(avatar(p), avatar(p))$
Detected $(avatar(p), impers(p))$	18	34
Detected $(avatar(p), avatar(p))$	32	16

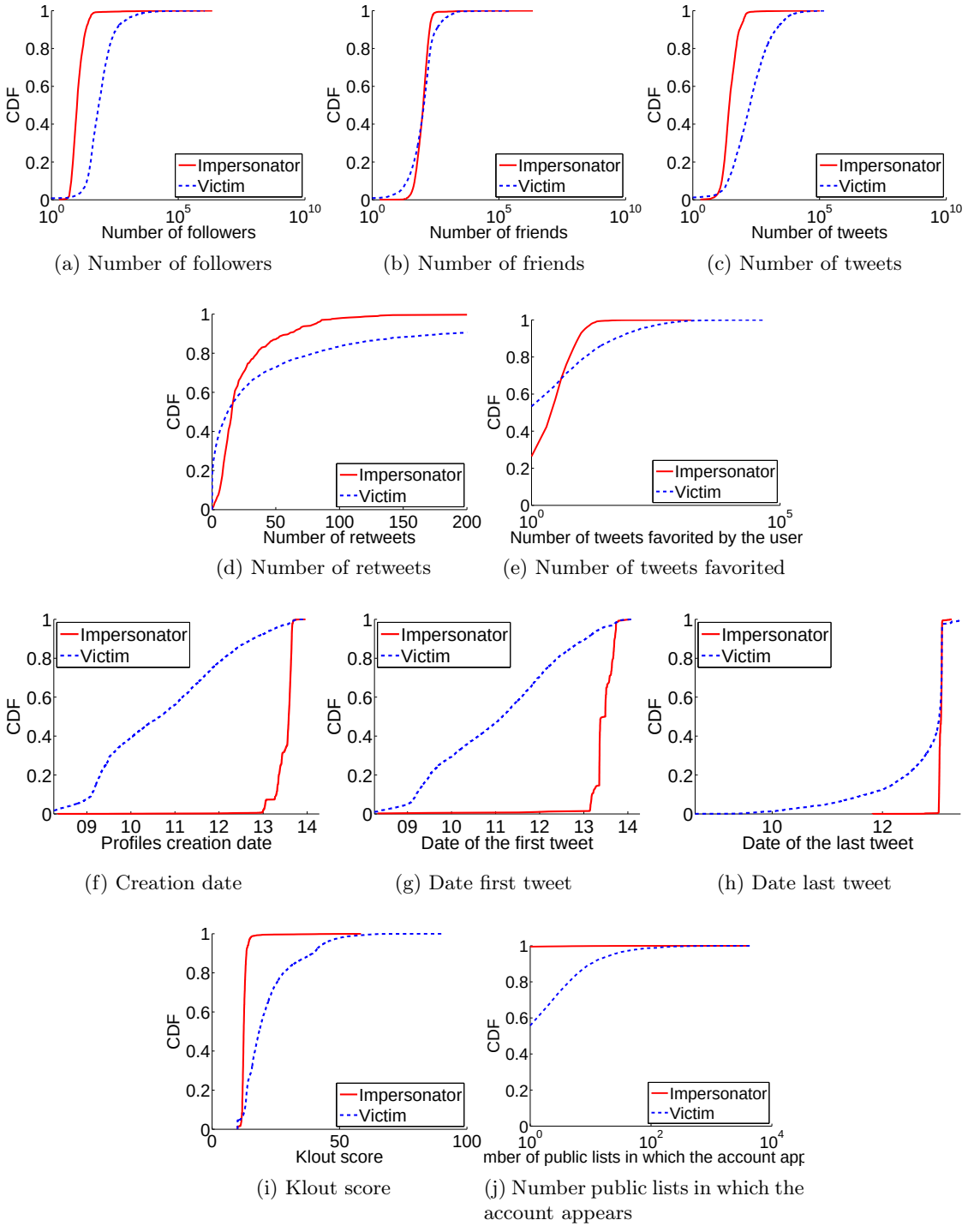
each assignment, we give AMT workers two links corresponding to the two Twitter accounts and we ask them to choose between four options: *'both accounts are real'*, *'both accounts are fake'*, *'one account is impersonating the other account'*, and *'cannot say'*.

Table 6.4 shows the results. AMT workers are able to correctly detect more impersonating accounts (18 out of 50 pairs of accounts), but, they correctly detected less avatar accounts (34 out of 50 pairs of accounts). Thus, in this scenario, AMT workers have a 36% true positive rate for a 68% false positive rate to detect $(avatar(p), impersonator(p))$ pairs of accounts and a 68% true positive rate for a 32% false positive rate to detect $(avatar(p), avatar(p))$ pairs of accounts. This result shows that humans are generally bad at identifying impersonators, in particular compared to an automated classification Pairs Classifier.

6.5 Characterization of impersonation attacks

In this section we want to understand who are the victims of impersonation attacks, what is the goal of impersonators and what is the behavior of impersonating accounts. We use the Impersonation Accounts and the Avatar Accounts datasets to analyze these questions. Figure 6.1 shows the CDFs of different properties of impersonating and victim accounts.

Characterization of victims: Figure 6.1a shows the CDF of the number of followers of victim accounts. The median number of followers is only 73. Since celebrities usually have from many thousands to millions of followers, this shows that attackers do not only target famous people but also ordinary users. Figure 6.1c shows the CDF of the number of tweets per victim account. The median number of tweets is 181. This shows that the victims are fairly active Twitter users. Furthermore, Figure 6.1h shows that 75% of victim accounts posted at lest one tweet in 2013. Figures 6.1j and 6.1i show the CDF of



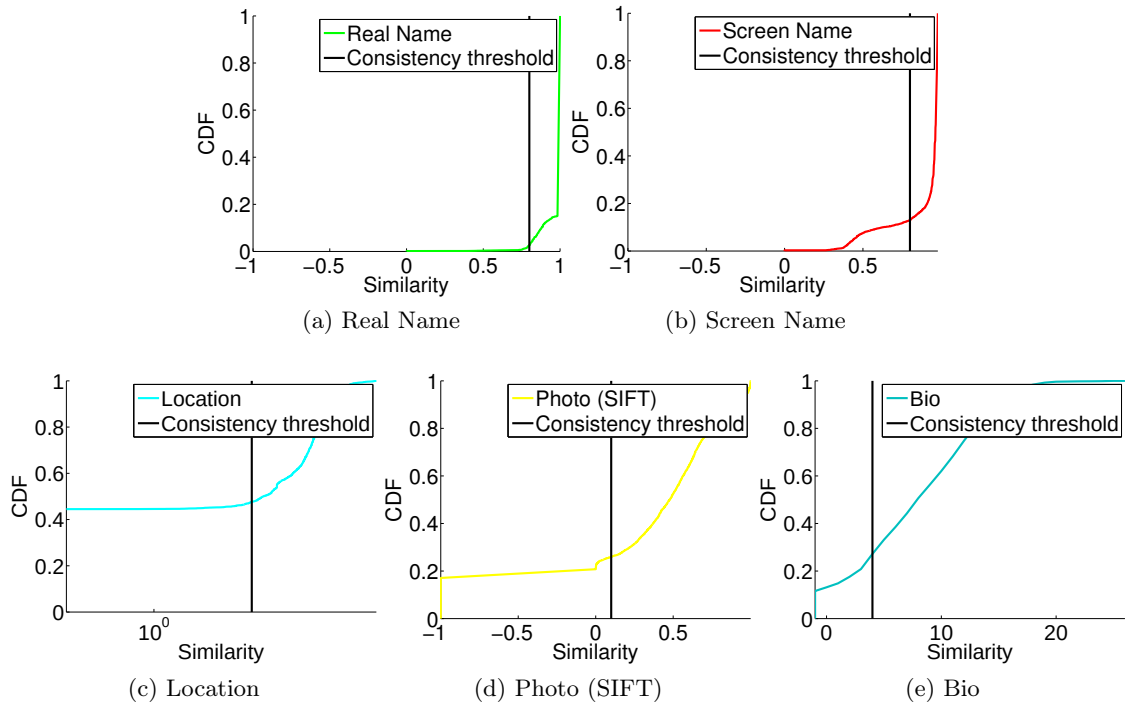
(a) Number of followers (b) Number of friends (c) Number of tweets

(d) Number of retweets (e) Number of tweets favorited

(f) Creation date (g) Date first tweet (h) Date last tweet

(i) Klout score (j) Number public lists in which the account appears

Figure 6.1: CDFs of different properties of impersonating and victim accounts.



(a) Real Name (b) Screen Name

(c) Location (d) Photo (SIFT) (e) Bio

Figure 6.2: CDFs of the similarity between different profile attributes between impersonating and victim accounts.

the number of lists where a victim account appears and the klout score. 40% of victim accounts appear in at least one list and 30% of users have klout scores higher than 25 (e.g. Dina Papagiannaki – https://twitter.com/dpapagia – has a klout score of 26 and Jon Crowcroft – http://twitter.com/tforcworc – has a klout score of 45). This shows that a significant fraction of victim accounts correspond to influential users. Consequently, the victim accounts are users that are active on Twitter, that are not necessarily celebrities, and some of them are quite influential on Twitter.

Characterization of impersonators: Figure 6.1d and Figure 6.1e show the CDFs of the number of retweets and the number of tweets favorited by impersonation accounts. This are the only two properties for which the impersonating accounts have higher counts than the victim accounts. Our intuition was that attackers create these accounts to sell fake retweets and fake favorites. Indeed, we checked who the impersonators are retweeting or favoring and we found that most of the impersonating accounts retweet tweets from a specific set of accounts. Consequently, we believe that most of the impersonating accounts we detected are used for retweet fraud and they do not actually try to directly harm the victims. The median number of followers, friends and tweets of impersonating accounts are 11, 104, and 28 respectively. Such low numbers do not denote suspicious activity. We believe that impersonators aim to keep their account properties in normal ranges so that they are not detected by fake account detection systems. The median number of impersonating accounts per victim accounts is one, but for 10% of the accounts, there are more than one impersonating accounts.

Figure 6.1f shows that impersonation attacks are a recent threat, as all the impersonating accounts are created starting 2013. Since these accounts operate under the radar, Twitter took in average 287 days to suspend these accounts. This means that victims were impersonated, and their online image was potentially harmed, for several months.

Figures 6.2a to 6.2e show the CDFs of similarity scores between different features between impersonating and victim accounts. These plots show how much information impersonators copy from victim accounts. The consistency threshold is the threshold identified by AMT workers after which they consider two values of an attribute to be consistent. If the similarity between two attributes cannot be computed because of missing values, we put a -1. These plots show a number of interesting things. First, most impersonators copy most of the profile attributes of their victims. Second, the location is not as consistent as the other attributes, there are more than 40% of accounts that do not have the same location as their victim accounts. This means that attackers either alter sometime the profile attributes copied from their victims, or do not update the impersonating profile.

6.6 Summary

We conducted the first study to detect and characterize impersonating accounts in current social networks. We showed that we can detect impersonating accounts in social networks by firstly identifying accounts that portray the same person inside the social network and then using a classifier to detect which are the impersonating accounts in the returned list.

We proposed a rule-based matching scheme that emulates human judgement to detect accounts that portray the same person inside a social network. The scheme infers that two accounts portray the same person whenever two accounts have consistent names and consistent bios or photos. The matching scheme is an easy and intuitive way to detect accounts that portray the same person and achieves a 65% recall for a 98% precision.

We proposed a new way of detecting fake accounts in a social network. Contrarily to previous work, our technique exploits features of pairs of accounts instead of features of single accounts. We showed that this technique is better suited to detect impersonating accounts. In fact, our AMT experiments showed that our technique is better than humans at detecting impersonating accounts. Studies of impersonators should avoid relying on humans as ground truth.

Our analysis of impersonating and victim accounts on Twitter revealed that attackers target a wide range of users and anyone that has a Twitter account can be victim of such attacks. We also discovered a new type of impersonation attacks where attackers create impersonating accounts to look real and to evade fake account detection schemes rather than to engage in social engineering attacks.

Our findings reveal a new type of privacy threat against the online image of users. Many surveys [38, 48] suggest that U.S. firms started to do background checks for job applicants that involve mining data from their online profiles. The most concerning fact is that attackers impersonate a wide range of users and use the accounts to promote different types of products and post random tweets. This can potentially have a tremendous negative

impact on the online image of users especially when someone searches for information about them and stumbles across the impersonating account. As we saw humans can be very easily tricked into thinking an impersonating account is a real account, thus we call for applications that can assist users when searching for people online to help them decide whether the account is an impersonator or an avatar.

Chapter 7

Conclusions and Perspectives

The proliferation of social data is providing many opportunities for developing exciting new applications. At the same time, however, it is also raising many new questions about the privacy and security of people online. An important premise of many new services and research is that it is possible to match the different accounts of a user. The main contribution of my thesis is the development and analysis of scalable and reliable matching schemes to match the accounts that correspond to the same individual in todays' social networks. Matching accounts across social networks allows applications to work on more complete user profiles. It also rises, however, more serious privacy concerns, in particular when we can match the accounts of users that deliberately change the information in their profiles to maintain separate personas. Finally, matching accounts within a site is a powerful tool to detect impersonators.

7.1 Summary of contributions

The thesis makes the following contributions:

Representative ground truth of matching accounts across social networks: We gathered ground truth data of matching accounts by exploiting the Friend Finder mechanism in social networks with a list of 10 million random email addresses. With this method, we can find all the accounts on different social networks that were created using a specific email address. This dataset better represents users in general than the ground truth used by most previous work that captured matching accounts of people that willingly provide links to their accounts in different social networks.

Characterization of user behavior across social networks: We show what kind of information users provide in different social networks and how consistent they are between different social networks. We found that only 7% of Twitter users have no attribute in common (not even names) with their Facebook account. The rest of the users have at least consistent names. We also show that the consistency and availably of attributes across accounts is higher for users who willingly provide links to their accounts in different social

networks that for random users. This result implies that this kind of ground truth will have higher matching rate than on random users.

Framework to evaluate the quality of features used for matching accounts: We proposed a set of four properties: Availability, Consistency, non-Impersonability and Discriminability (ACID) to evaluate the quality of different features to match accounts. We believe that these properties are necessary and sufficient for an ideal feature to have in order to guarantee a scalable and reliable matching. None of the attributes extracted from users profiles, however, have all four properties. Thus, we proposed ways to combine them in order to build a scalable and reliable matching scheme. Besides being useful in deciding whether a new feature has the potential to improve matching accuracy, these properties should be useful to anonymize datasets collected from social networks and measure the potential risks of de-anonymization.

Evaluation of the performance of human workers to match accounts: We evaluated the performance of human workers to match accounts using Amazon Mechanical Turk (AMT) experiments. In an experiment where we showed AMT workers two pairs of accounts and asked them whether they correspond to the same person or not, humans were able to correctly detect 60% of the matching accounts with no false matches. For the remaining 40% of accounts, AMT workers said that they cannot decide due to lack of information. This study showed that: (1) humans do a good job at identifying when two accounts correspond to the same person and when they do not; (2) humans need more attributes than just names matching in order to be confident that two accounts correspond to the same person.

A multifaceted approach to evaluate matching schemes: We proposed to evaluate matching schemes in three scenarios: on a small dataset, on a large-scale dataset and against human workers. Our evaluation shows that a matching scheme with a high accuracy on a small dataset may have low accuracy on a large-scale dataset. Furthermore, a more realistic target for matching schemes based on the public profiles of users is to achieve the accuracy of humans rather than detect 100% of matching accounts.

A scalable and reliable matching scheme based on public attributes: We developed a scheme in three steps that exploits public attributes provided by users to achieve a reliable matching even when applied on large social networks. Our problem has some unique constraints that prevent us from using general matching schemes developed for entity matching. First, we have to deal with very large datasets. For example, Facebook has over 1 billion users, hence the data contains one billion matching accounts and one quintillion non-matching accounts. Second, the number of features we can use to match accounts is relatively small because there are not many profile attributes available across multiple social networks. The most common profile attributes are names, location, friends, and profile photos. The small number of available attributes hampers the accuracy of traditional matching methods so that the matching is no longer reliable. To overcome this problem, instead of using one classifier to detect matching and non-matching accounts, we

use two classifiers sequentially that achieve better accuracy. Furthermore, we take advantage of the fact that there can only be one matching account on a second social network to increase the reliability of the matching. At the end, we are able to achieve a 21% recall for a 98% precision which is close to what humans can achieve, 25% recall for a 98% precision. Our scheme is robust to impersonators.

An online service to match accounts: We developed an online service that takes as input an account in Twitter and searches in real-time the matching account on Facebook. The service can be found at http://matchingaccounts.app-ns.mpi-sws.org/. Besides providing a potentially useful tool to the public, this service allowed us to test our matching schemes in the wild and to identify the challenges faced when using the matching schemes in the real world.

A matching scheme using only innocuous information about user activities: We showed that by only exploiting the location, timing and writing style of user's posts we can match his accounts across social networks. This result has important privacy implications as users tend to think that, if they maintain different pseudonyms on different social networks, their accounts are not linkable. We show, however, that we can leverage the user activity to match accounts. We show for example that, using the location information, we can match 60% of Flickr accounts with their corresponding Twitter accounts, while only introducing a small percentage of falsely matching accounts. Moreover, our results show that we only need coarse-grained location information to match a relevant number of accounts. Combining all features together gives comparable results with matching on screen names for Flickr to Twitter matching, and can identify 37% more matching accounts for Yelp to Twitter matching.

A method to detect impersonators inside a social network: We developed a method to detect impersonators in two steps. Our method, first, exploits the same similarity metrics used for matching accounts across social networks to detect accounts that portray to the same person inside a social network. We then use a classifier built on features that characterize pairs of accounts to detect whether a pair of accounts corresponds to different avatars of the same person or one account is impersonating the other. Traditional methods to detect fake accounts perform poorly for detecting impersonators. Our study shows that detecting impersonators requires to build methods that exploit features that characterize pairs of accounts rather than features that characterize single accounts as done so far.

Demonstrate that humans are not very good at detecting impersonators: Many social networks use humans to check if an account corresponds to a real person. Our experiments on AMT show that humans are not very good at detecting impersonators. Humans can achieve a 18% true positive rate for a 82% false positive rate to detect impersonating accounts and a 96% true positive rate for a 4% false positive rate to detect avatar accounts. Hence, their accuracy is not good enough to be used as ground truth.

A characterization study of impersonators: We performed the first characterization of impersonators on Twitter. Our analysis of 5,693 impersonating accounts shows that not only celebrities are victims of such attacks but anyone with a Twitter account. We also show the main purpose of these impersonating accounts is to avoid traditional fake account detection schemes rather than launch social engineering attacks on their victims.

Together, these contributions advance the state of the art in measuring and protecting the online footprint of users. We hope that our work will lead to better tools to manage and control the online image of individuals as we discuss next.

7.2 Future work

Our ideas for future work are both on improving the accuracy of matching schemes and on potential applications. We also have some more long term research agendas regarding the privacy of users online.

7.2.1 Improving matching schemes

Mobile phones are able to connect to a wide variety of sensors that can log almost all user activities. Applications on mobile phones use these sensors to provide different monitoring services: we have applications to monitor sleep, heart rate or the number of steps we do each day. Most of these applications offer the possibility to share these logs on Facebook or other social networks. Sensors are still in their early days and we can only expect that more and more sensors will surround us. With the proliferation of all these sensors, mobile applications and social networks integration, it is becoming easier and easier to share any aspect of a user's life at a very fine grain and provide high quality context data to every post. There is already a lot of data about users online and we can only expect this data to become even more precise and complete in the future. Thus, it is important to keep developing better matching schemes.

Improve the account matching using the graph structure of the social network: In this thesis, we only exploited the friends of a user to match their accounts without exploiting the full graph structure of the social network. There are many studies that propose techniques to de-anonymize graphs. The main idea of such approaches is to start with a few nodes (seeds) for which we know the corresponding accounts on the two graphs, and then propagate the matching by measuring the similarity between the two graphs. We plan to combine such approaches with the method we currently have to match accounts.

Study the uniqueness of named friendship relations: In this thesis we used the friends of a user as a feature to match accounts. We feel however, that we did not explore fully this features. We want to investigate what is the uniqueness of friendship links: i.e. if a user, *user1*, has the name *name1* and one of his friends, *user2*, has the name *name2*, we want to check if there exist other users, *user3* and *user4* with the same *name1* and *name2* that are friends inside the same social network. If a friendship relation is unique,

whenever we see two people that are friends with *name1* and *name2*, we will know for sure that they correspond to *user1* and *user2*. If true, this would be a powerful and easy method to match accounts across social networks.

Measure the spread of single sign-on services: Besides improving our matching schemes, we can also study the implicit matching of accounts provided by single sign-on services. We want to measure what percentage of users connect with their social networks accounts on third-party sites. We would like to understand who are these users, to which types of sites they are connecting and how often they share the activities they do on third-party sites on social networks. By measuring the integration between social networks and third-party sites we will be able to understand how much information on social networks is generated with the help of third-party sites and what third-party sites learn about their users from social networks. Furthermore, a better understanding of implicit matching can potentially lead to applications that build on top of it. In the next section we will give an example of such application.

7.2.2 Applications of matching

The amount of data generated by users that is publicly available has kindled a lot of fundamental research on online communities such as user influence estimation [188], user expertise estimation [117, 162], community structure and link analysis [139] and opinion mining [144]. Most prior research, however, has focused on single social networks and did not analyze social networks in aggregate. Knowing the matching accounts of users would allow to compare and combine findings across social networks and have a better global view. Finally, it also opens new research directions like understanding the migration patters of users between sites [108] or detect impersonators.

Following are a few particular research studies that we would to pursue:

Sybil detection using cross-site information: A lot of recent research has focused on the problem of verifying user identities (i.e., assessing the trustworthiness of user accounts) in social networks like Twitter, Facebook, LinkedIn, or Google+. Existing proposals to detect fake / untrustworthy identities rely on analyzing information about the identities (e.g., their characteristics or activities) that is available within a single site or domain. We want to investigate the feasibility of leveraging information about identities that is aggregated across multiple social networking sites to reason about their trustworthiness. The key insight is that, while honest (especially popular) users naturally maintain their *presence* on multiple sites, attackers might be discouraged by the additional effort and costs to do the same.

Combine multiple sources of information to infer the expertise of users and transfer expertise from one social network to another: Researchers showed it is possible to infer the expertise of users on social networks [117, 162]. For example, Sharma et al. [162] showed that we can easily infer the expertise of users on Twitter by exploiting Twitter *lists* (thematic lists where users can group the people who they are following).

Finding experts on Twitter leads to easy ways to find people to follow and find relevant information regarding a subject. While Twitter and other social networks have metadata that helps the organization and inference of such information, blogs, on the other hand, have no such infrastructure. We want to provide better tools for finding relevant content on blogs by matching blogs to their corresponding accounts on Twitter. This approach will permit to structure the unstructured world of blogs using the structured world of Twitter. Furthermore, we can use the content on both sites to further refine the expertise of users.

Understand what makes a person popular across different social networks: People are often interested in strategies to make their accounts more popular on different social networks. Knowing that we can match accounts across social networks, we want to investigate a number of questions: do the same strategies have the same impact on different social networks? does the popularity on one social network influence the popularity on another social network? is the audience the same on different social networks? Answering these questions will lead to a better understanding of what makes an account popular and what are the best strategies to adopt by companies and people who want to increase their popularity.

Understand how information spreads from one social network to another: We would like to understand whether information spreads from one social network to another and, if so, how. We would like to investigate whether there exist *relying nodes* that always transfer the information or whether there are only random users that transfer random pieces of information. If there are indeed relying nodes, this information could be valuable for advertisers and marketers.

Recommenders with heterogeneous information: So far, recommender system have focused on recommending content inside a system using the information gathered inside the particular system. Knowing the users accounts on different social networks will allow us to see if it is possible to use the data learned in one site to recommend things in another site. For example, we would like to check the information learned about users on Quora can help do better recommendations on LinkedIn. Or, the profile learned about a user on Twitter can help make better recommendations on a shopping site. This study can be useful, for example, to better estimate the value of personal data, to bootstrap recommender systems or to better understand users and how their preferences relate on different subjects.

Investigate whether single sign-on services help reduce spam: Even if this study does not build directly on our matching scheme, it is based on the implicit matching of accounts provided by single sign-on services. Detecting spam is a continuous fight. Big social networks such as LinkedIn, Facebook or Twitter use sophisticated techniques to detect spam and have entire teams that work full time on this [169]. Smaller sites or social networks, however, do not have access to the same resources. We want to investigate whether third-parity sites that allow users to connect with their Facebook or Twitter accounts have less spam. Our intuition is that, first, it is harder to create a fake account on

Facebook or Twitter because of their fake account detection system, and second, Facebook and Twitter might be quicker at detecting accounts that post spam and suspending them. We plan to do a measurement study on Quora. This study has several implications, if it is indeed true that connecting with a Facebook or Twitter account reduces spam on third-party sites, than Facebook or Twitter could sell this as a service. At the same time third-party sites will have even better incentive to integrate with Facebook or Twitter. Furthermore, this can be the initial step towards proposing collaborative schemes to detect spam where both Facebook or Twitter and third-party sites participate with information about the trustworthiness of accounts.

7.2.3 Protecting user privacy

The privacy of users online will remain an important topic. Although we do not have short term research agendas, we are maturing at two ideas regarding online privacy.

Technical and legal solutions for protecting and controlling the digital footprint of individuals: Given the current data enthusiasm where everybody is logging everything we are at a point where we leave digital traces everywhere we go with or without our knowledge and will. Ideal solutions for protecting the privacy of individuals should protect their whole digital footprint both online and offline. The online digital footprint contains anything that is publicly visible on the Internet about a given person, while the offline digital footprint includes all the logs that are collected but not shared online. These logs can range from medical records to bank statements and tracking cookies. The problem of protecting users' privacy is complex and legal approaches should be combined with technical approaches to obtain optimal solutions. For example, one combined approach could consist of: (1) laws requiring companies to share with the user any kind of data they collect about them; (2) technical solutions to store and collect such data, visualize it, and have algorithms to show what else can be inferred.

Privacy specific to every person: The notion of privacy is different from person to person and it is hard to build tools or laws that satisfy everyone's perspective. For example, teenagers mostly do not want their parents to see their posts. For young adults, privacy could mean that a future employer cannot see the photos of their undergrad parties. For other adults privacy might mean that an organization they want to be part of will not deny their membership because of their political convictions. Therefore, we need research in sociology to better understand privacy.

Résumé en français

Aujourd'hui, plus de 2,4 milliards d'utilisateurs ont accès à l'Internet et une grande partie d'entre eux ont un compte actif sur un réseau social. Il y a plus de 1,2 milliards utilisateurs actifs sur Facebook seul (728 millions de ces utilisateurs se connecte quotidiennement), 540 millions sur Google+, 259 millions sur LinkedIn, et 232 millions sur Twitter [171]. En 2009, une étude d'Anderson Analytics [178] a montré que 91% des utilisateurs de Twitter et 82% des utilisateurs de LinkedIn ont également un compte Facebook. Nous nous attendons à ce que ce chevauchement soit encore plus élevé en 2014. Les utilisateurs partagent toutes sortes d'informations sur les réseaux sociaux à un rythme effarant. Par exemple les utilisateurs de Facebook partagent 4,75 milliards de pièces de contenu quotidiennement. Toutes les 60 secondes, les utilisateurs postent 510 commentaires, 293 000 mises à jour de statuts, et 136 000 téléchargements de photos sur Facebook [141]. Les utilisateurs se livrent souvent à des activités différentes et révèlent des informations sur différents aspects de leur vie sur différents réseaux sociaux. Sur Facebook, les utilisateurs communiquent avec leurs familles et leurs amis et partagent certains aspects de leur vie personnelle. Sur LinkedIn, les utilisateurs donnent des détails sur leur évolution professionnelle et à leurs aspirations. Sur Twitter, les utilisateurs ont tendance à poster des choses qui les passionnent.

Il y a un intérêt croissant dans l'identification des multiples comptes qui correspondent à un seul individu. Premièrement, les organisations sont intéressées par la corrélation des activités d'un utilisateur pour regrouper toutes les informations sur plusieurs réseaux sociaux et élaborer un profil plus complet d'un utilisateur individuel que le profil fourni par un seul réseau social. Deuxièmement, les réseaux sociaux s'intéressent à la découverte de tous les comptes correspondant à un seul individu au sein d'un seul réseau social. Les utilisateurs sont censés ouvrir un seul compte dans un réseau social (comme stipulé dans les conditions de service), cependant certains utilisateurs créent plusieurs comptes. En outre, les utilisateurs malveillants usurpent souvent l'identité des utilisateurs honnêtes. Dans les deux cas, nous avons besoin de techniques permettant de trouver les comptes d'un seul individu.

Nous voyons déjà des modèles légitimes d'entreprises basés sur ces techniques de corrélation. De nombreuses compagnies émergentes essayent d'analyser automatiquement les profils d'utilisateurs sur différents réseaux sociaux pour aider les recruteurs dans leurs décisions [166]. Des portails de vente combinent les informations téléphoniques avec les informations sur différents réseaux sociaux pour présenter un profil plus complet d'un client aux représentants des ventes. Cela permet d'aider les appelants des hotlines plus efficacement [159].

Finalement, certaines sociétés comme PeekYou [146] et Spokeo [4] proposent des *"moteurs de recherche de personnes"* : a partir d'informations de base telles que noms d'utilisateur ou noms et prénoms réels, ces services retournent les profils d'utilisateur collectées dans différents réseaux sociaux.

La corrélation des comptes d'utilisateurs sur différents réseaux sociaux a aussi des applications pour des nombreux problèmes de recherche. Il

y a beaucoup de recherche fondamentale sur les communautés sociales telles que l'estimation d'influence d'utilisateurs [188], l'estimation de l'expertise d'utilisateurs [117], la structure des communautés en ligne [139] et l'analyse d'opinion [144]. A ce jour, les chercheurs ont analysé cette problématique dans un seul réseau social et ils n'ont jamais analysé les problèmes à travers plusieurs réseaux sociaux. Par exemple, les chercheurs ont étudié le comportement des utilisateurs sur Facebook ou LinkedIn séparément [10, 18, 182]. Ceci donne uniquement une vue partielle d'un utilisateur. Les interactions dans Facebook peuvent probablement seulement caractériser les interactions avec des amis, et les interactions de LinkedIn peuvent probablement seulement caractériser les interactions avec les collègues de travail. Connaître la correspondance des comptes sur plusieurs réseaux sociaux fournit l'occasion de construire un meilleur portrait d'un utilisateur. Avoir une meilleure compréhension d'un utilisateur peut ensuite mener à de meilleurs services personnalisés ou une meilleure estimation d'expertise d'utilisateurs. Un bloc de construction important pour toute recherche à travers différents réseaux sociaux est d'avoir des techniques fiables pour corréler les comptes sur les réseaux sociaux.

Alors que la création de tels profils complets des utilisateurs a de nombreuses applications dans l'industrie et la recherche, elle apporte aussi des légitimes et sérieuses préoccupations sur la protection de la vie privée des utilisateurs en ligne. Sur chaque site, un utilisateur peut juger approprié ce qu'il poste sur son compte Facebook, Twitter, ou LinkedIn ; mais il peut révéler beaucoup plus que ce dont il ne se rend compte lorsqu'on les examine de façon *agrégée*. Par exemple, une attaque d'ingénierie sociale pourrait d'abord identifier les employés d'une organisation victime sur LinkedIn, puis examiner leurs comptes Facebook pour trouver des

antécédents personnels a exploiter et regarder leurs tweets pour comprendre leurs modes de déplacement.

Dans cette thèse, nous développons des méthodes pour identifier les comptes sur différents réseaux sociaux d'un utilisateur donné. Nous avons d'abord étudié comment nous pouvons exploiter le profils public (p. ex., les nom d'utilisateurs, les endroits où ils vivent, la bio et photo de profil) que les utilisateurs maintiennent dans différents réseaux sociaux pour corréler à leurs comptes. Nous avons identifié quatre propriétés importantes – la disponibilité, la cohérence, la non-impersonabilité et discriminabilité (ACID) – pour évaluer la qualité des différents attributs de profil pour corréler les comptes. Exploiter les profils a un bon potentiel pour corréler les comptes, car un grand nombre d'utilisateurs ont les mêmes noms et d'autres informations personnelles sur différents réseaux sociaux. Pourtant, il demeure difficile d'atteindre une précision utile en pratique en raison de l'ampleur réelle des réseaux sociaux. Afin de démontrer que cette corrélation des comptes est faisable sur des vrais réseaux sociaux et les erreurs sont suffisamment fiables pour qu'elle soit utilisée dans la pratique, nous avons mis l'accent sur la conception des méthodes qui permettent d'atteindre un faible taux d'erreur même lorsqu'elles sont appliquées à des réseaux à grande échelle avec des centaines de millions d'utilisateurs. Puis, nous montrons que nous pouvons encore corréler les comptes sur des différents réseaux sociaux même si nous exploitons seulement ce que les utilisateurs postent, c'est-à-dire leur activité sur un réseaux social. Ceci démontre que, même si les utilisateurs sont conscients et cherchent à maintenir des profils différents sur différents réseaux sociaux, nous pouvons encore potentiellement corréler à leurs comptes. Enfin, nous montrons que, par l'identification des comptes qui correspondent à la même personne au sein d'un réseau

social, nous pouvons détecter les usurpateurs d'identités.

Données personnelles partagées dans les réseaux sociaux

Une grande partie des données partagées par les utilisateurs est public parce que le but d'avoir un profil dans de nombreux réseaux sociaux est de rendre un individu plus visible. Le but d'un profil LinkedIn est d'avoir une meilleure visibilité pour les recruteurs potentiels, alors que le but d'un profil Twitter est d'atteindre une large audience pour promouvoir des idées et des intérêts. Plus simplement, la raison pour rendre le contenu public est soit d'atteindre soit d'être atteint plus facilement par d'autres utilisateurs. De ce fait, cependant, n'importe qui peut facilement accéder à ce contenu et l'analyser de façons qui sont hors du contrôle de l'utilisateur qui l'a généré.

En général, nous pouvons trouver trois types d'informations sur les utilisateurs d'un réseau social. Nous avons d'abord le *profil utilisateur* qui est l'information quel les utilisateurs fournissent au sujet d'eux-mêmes. Les informations de profil peuvent inclure des attributs comme le nom de l'utilisateur, la ville où il vit actuellement, où il est allé à l'école, son employeur actuel, le jour de son anniversaire, ou la photo du profil. En plus du profil utilisateur nous avons également des informations sur les *activités de l'utilisateur* dans le réseau social, comme ce que l'utilisateur poste ou quels livres, films, ou équipes sportives il aime. Les posts des utilisateurs viennent souvent avec des métadonnées. Par exemple, dans Facebook et Twitter, les posts peuvent être marqués avec l'exacte position d'où le message a été envoyé. Sur Yelp, les utilisateurs notes les restaurants, de sorte que nous disposons d'informations sur l'emplacement et le type de restaurant. D'autres types de métadonnées

incluent les hashtags, la langue du poste ou quelle application a été utilisée pour poster le post. Enfin, nous disposons d'informations sur le *graphe social*: les amis des utilisateurs.

Tous les réseaux sociaux permettent aux utilisateurs de marquer des éléments de leur profiles comme contenu privé. Même avec les paramètres les plus restrictifs, cependant, il y a toujours des éléments d'information qui doivent être rendus publiques. La quantité minimale d'informations qui reste toujours public dans tout réseau social est le nom réel, le nom d'écran, et la photo du profil d'un utilisateur.

Techniques actuelles pour corréler les comptes d'utilisateurs

Aujourd'hui, un certain nombre d'organisations regroupent déjà des informations d'un utilisateur unique sur plusieurs sites. Il existe quatre approches actuelles pour corréler des comptes: (i) des sites qui permettent aux utilisateurs de montrer explicitement les liens vers leurs profils sur différents réseaux sociaux; (ii) les sites qui prennent en charge les services single sign-on qui corrèle implicitement les comptes; (iii) les moteurs de recherche qui utilisent des algorithmes qui exploitent les noms d'utilisateur; et (iv) les algorithmes proposés par les chercheurs qui exploitent différentes sortes de données dans les profils des utilisateurs. Nous discutons maintenant ces approches de manière plus détaillée.

Des réseaux sociaux comme Google+ permettent aux utilisateurs de lister leurs comptes sur d'autres réseaux sociaux. Les gens, les marques, et les compagnies qui veulent avoir une forte présence sur le web se voient souvent conseillées de maintenir des comptes actifs sur différents réseaux sociaux et de poster des liens d'un compte à un autre. Avoir plusieurs

comptes reliés ensemble accroît leur visibilité et peut aussi accroître leur position dans Google search [161]. Outre Google+, il y a aussi des sites dédiés tels que `itsmyurls.com`, `environ.me`, et `hi.im` qui permettent aux utilisateurs de créer une page où ils publient des liens vers tous leurs comptes. Ces sites permettent aux utilisateurs de mieux gérer leur empreinte en ligne. Outre certains utilisateurs qui veulent maintenir une forte présence sur le web, il n'y a pas beaucoup d'autres utilisateurs qui utilisent ces services pour lister leurs comptes.

Single sign-on est un type d'authentification de l'utilisateur qui permet à un utilisateur de saisir un nom et un mot de passe pour se connecter à plusieurs sites ou applications. OpenID [155] était la technique traditionnelle sur le web. Cependant, cette technique ne réussit pas à obtenir une grande adoption des sites et utilisateurs, principalement à cause du manque de mesures incitatives [172]. Au cours des dernières années, Facebook, Google+, Twitter, LinkedIn ont commencé à autoriser les utilisateurs à se connecter à d'autres sites avec leurs comptes de réseaux sociaux. Cela fait partie d'un effort visant à autoriser des sites et des applications tierces à s'intégrer aux réseaux sociaux. Nous pouvons voir l'intégration entre Facebook, Google+, Twitter et LinkedIn avec des sites tiers comme une *corrélation implicite* des comptes entre les deux sites. L'intégration de Facebook est déjà très populaire, plus de 24,3% du top 10 000 des sites web ont une certaine forme d'intégration [150]. Cette solution a reçu une plus grande adoption parce que l'intégration est bénéfique pour les propriétaires des sites et des réseaux sociaux. Lorsque les utilisateurs se connectent avec leur compte Facebook sur un site tiers, Facebook partage certaines informations sur l'utilisateur telles que leur âge, ville actuelle ou ce qu'il aime. Dans le même temps, Facebook en tire des avantages : cela rend très facile pour les utilisateurs de partager

sur Facebook toute activité qu'ils font n'importe où sur Internet. Par conséquent, quelques-uns des principaux réseaux sociaux pourrait devenir à la fois les agrégateurs et les répartiteurs d'une grandes parties des données générées par les utilisateurs en ligne. Toutefois, il n'y a pas une corrélation implicite entre les différents réseaux sociaux.

Les moteurs actuels de recherche tels que `peekyou.com`, `spokeo.com`, `wink.com`, `pipl.com`, et `zabasearch.com` agrègent d'informations globales sur les utilisateurs sans leur consentement explicite ou connaissance. La plupart des moteurs de recherche utilisent des données agrégées des registres publics, des enquêtes et des réseaux sociaux. Généralement, lorsqu'ils sont interrogés, ils ont simplement à retourner tous les comptes de personnes partageant le même nom. Étant donné que, dans la plupart des cas, les noms des personnes ne sont pas uniques, les résultats peuvent conduire à de nombreuses fausses corrélations. Spokeo stipule explicitement dans leurs conditions d'utilisation qu'ils ne garantissent pas l'exactitude de leurs données. Certains moteurs de recherche utilisent plusieurs algorithmes sophistiqués. PeekYou a déposé une demande de brevet [74] pour corréler les noms de personnes à leurs comptes dans les blogs, les réseaux sociaux et des forums. L'algorithme consiste principalement, à l'aide des informations recueillies sur des sites différents, à attribuer empiriquement différents poids aux différents éléments d'information sur l'identités. Ces heuristiques ne sont pas fiables, comme signalé par Perito et al. [148]. Pour construire des services, il est crucial de comprendre les limites et les capacités de ces techniques d'appariement.

Enfin, il y a de nombreux efforts récents de recherche dirigés vers la corrélation des comptes à travers différents réseaux sociaux [6, 16, 76, 122, 128, 134, 137, 145, 148, 195]. Ces efforts mobilisent un variété de don-

nées de comptes allant d'attributs public de profil utilisateur au contenu généré par l'utilisateur et aux données privées de comptes. Ce travail fait allusion au potentiel de corrélation à grande échelle des comptes ; toutefois, la plupart de ces études n'ont pas évalué leurs méthodes à l'échelle [76, 128, 134, 148, 195] et le peu qui l'ont fait ont constaté que leurs méthodes ont tendance à être très peu fiable [6, 122, 145], c'est-à-dire qu'ils ont un grand nombre de fausses corrélations. Ainsi, le problème des corrélations fiables reste un défi ouvert.

Les menaces à la vie privée et à la sécurité dans les réseaux sociaux

Les gens sont de plus en plus intéressés par la confidentialité en ligne avec la médiatisation des risques possibles de partage des contenus personnels. Nous pouvons voir des réactions quand il y a une violation de données et de nombreuses poursuites ont été intentées contre Google, Yahoo ou Target [176]. En conséquence, de nombreux gouvernements et organisations proposent des moyens de réglementer la vie privée en ligne [175].

Le problème de la protection de la vie privée des gens en ligne est difficile. Chaque jour, nous découvrons de nouvelles attaques de la confidentialité en ligne contre des gens et nous n'avons pas encore une vue claire de toutes les attaques possibles. Ainsi, il est difficile de savoir quelles sont les mesures à prendre et à quel point elles sont globales. Généralement, une grande partie de l'attention des gouvernements et des organisations se concentre sur comment les grands sites tels que Facebook ou Google et les annonceurs comme DoubleClick voient nos schémas de navigation à

travers les cookies. Les défenseurs de la vie privée surveillent également les changements dans les propriétés des sites individuels, tels que les paramètres de partage sur Facebook et le nouveau Google les termes de service.

Ce qui a été négligé jusqu'à présent est une plus large menace d'attaquants corrélant informations personnelles *au delà des limites des sites*. En effet, nous pouvons en apprendre beaucoup plus sur un utilisateur lorsque nous connaissons ses comptes sur plusieurs réseaux sociaux. Sur Facebook, nous pouvons apprendre des détails sur un individu, par exemple, le jour de son anniversaire, ses films préférés, et où il est allé à l'école. Nous pouvons également déduire certains renseignements. Par exemple, sur Twitter nous pouvons déduire les intérêts d'une personne en analysant le texte de ses tweets ou qui il suit [127]. Sur LinkedIn, nous avons tous les contacts professionnels présents et passés d'un utilisateur, nous pouvons apprendre les sociétés pour qui il a travaillé et quelles sont ses compétences. Sur Yelp, nous pouvons apprendre quel genre de cuisine l'utilisateur veut et où il veut aller. Par conséquent, il est encore plus difficile de protéger la confidentialité des utilisateurs en ligne lorsque les attaquants peuvent mettre en corrélation des informations de différents sites et inférer de nouvelles informations sur les utilisateurs. La première étape pour atténuer cette menace est d'être capable de mesurer l'empreinte en ligne d'un utilisateur. Comprendre comment nous pouvons facilement faire correspondre les comptes d'un individu sur différents réseaux sociaux peut fournir aux utilisateurs des outils permettant de mesurer leur empreinte en ligne et de mieux comprendre les risques d'entrave à la vie privée.

Notre discussion a porté jusqu'à maintenant sur la protection de la vie privée menace de regrouper des utilisateurs des informations à travers

les réseaux sociaux. Une autre classe de menaces pour l'utilisateur en ligne image provient d'imitateurs. Il y a de plus en plus anecdotiques témoigne des célébrités et personnes importantes être usurpée [126, 160] mais d'autres personnes sont potentiel d'usurpation d'identité objectifs trop. Depuis un fraudeur peut sérieusement affecter l'image en ligne de l'utilisateur, il est très important de détecter de telles attaques. Jusqu'ici, cependant, la plupart de l'attention des chercheurs et de réseau social les administrateurs a mis l'accent sur la détection faux comptes ou spam [17, 25, 196, 200], il n'existe aucun cadre pour détecter automatiquement imitateurs et la seule solution est que les victimes à signaler manuellement les comptes qui usurpent leur [75].

Contributions

Dans cette thèse, nous développons et analysons des méthodes pour identifier les comptes qui correspondent à la même personne sur différents réseaux sociaux et à l'intérieur d'un même réseau social. Nous exploitons les techniques pour corréler des comptes correspondant à la même personne à l'intérieur d'un réseau social pour détecter les usurpateurs d'identité en ligne. Plus précisément, les contributions de la thèse sont les suivantes.

Corrélation des comptes sur des réseaux sociaux fiable à grande échelle

Il y a beaucoup d'intérêt et de préoccupation, tant dans le domaine de la recherche que dans l'industrie, pour d'éventuelles corrélations des

comptes d'un utilisateur sur plusieurs réseaux sociaux. Nous mettons l'accent sur le défi de la conception de méthodes pour corrélation de comptes exploitant des donnes publiquement visibles pour atteindre une haute *fiabilité*, c'est-à-dire, un faible taux d'erreur même lorsqu'il est appliqué à *grande échelle* dans des réseaux avec des centaines de millions d'utilisateurs. Le principal défi dans la réalisation d'une corrélation fiable provient du bruit inhérent des données publiquement accessibles. Nous avons identifié quatre propriétés importantes – la disponibilité, la cohérence, la non-impersonabilité et discriminabilité (ACID) – afin d'évaluer la qualité des différents attributs dans la comparaison de comptes. Les attributs publics comme le nom, l'emplacement, photo de profil, et amis satisfont les propriétés ACID à des degrés divers, ce qui rend la détection simple des comptes corrélés utilisant directement des techniques simples de classifications inexactes lorsqu'elle est appliqué à grande échelle.

Nous montrons qu'il est possible de tirer parti de plusieurs attributs pour créer un schéma fiable et évolutif correspondant à une classification en trois étapes : d'abord filtrer les comptes qui sont clairement différents, puis désambiguer le vrai compte correspondant d'un ensemble de comptes similaires, et enfin assurer la fiabilité en mesurant à quel point le compte le plus corrélé est distinct des autres comptes similaires. Les techniques précédentes de matching d'entités ne sont pas efficaces pour la comparaison de comptes en raison des contraintes et spécificités de notre scénario : (1) nous avons à traiter de très grands ensembles de données; (2) nous ne disposont que d'un nombre restreint d'attributs pour discriminer le compte correspondant parmi plus d'un milliard d'autres comptes ; et (3) nous sommes sûrs qu'il ne peut y avoir qu'un seul vrai compte correspondant dans un réseau social. Nous éval-

uons les performances de matching de comptes sur Facebook et Twitter, deux des plus grands le monde réel les réseaux sociaux. Nos résultats montrent que la corrélation de comptes à grande échelle est difficile. Malgré tout, nos techniques peuvent matcher 30% de Twitter avec leur comptes Facebook correspondants, avec 92% de précision. Nos constatations reflètent le potentiel ainsi que les limites de la comparaison fiable de comptes à l'échelle en utilisant uniquement les attributs des comptes d'utilisateur. Outre les contributions analytiques, nous avons également développé un service en ligne qui prend en entrée un compte Twitter et recherche en temps réel le compte correspondant sur Facebook, qui peut être trouvé à http://matchingaccounts.app-ns.mpi-sws.org/.

Dans la poursuite de l'édification d'un schéma de matching fiable et évolutif, nous avons également fait certaines contributions méthodologiques. Premièrement, nous avons développé une méthode objective pour rassembler les données "ground truth" de la correspondance de comptes. Nous pensons que cette méthode donne un échantillon représentatif d'utilisateurs en général. Deuxièmement, nous proposons une évaluation systématique des schémas de matching qui nous montre que la précision d'un schémas de matching à petite échelle n'est pas indicative de la précision du système à grande échelle. Enfin, nous avons évalué comment les humains peuvent matcher des comptes. La précision des humains est un standard plus réaliste de la précision que nous pouvons espérer atteindre par un schéma de matching automatique.

Corrélation de comptes en exploitant les activités d'utilisateurs

Nous étudions comment les pirates potentiels peuvent identifier les comptes sur différents réseaux sociaux qui appartiennent au même utilisateur,

en exploitant seulement l'activité des utilisateurs. Nous examinons trois caractéristiques spécifiques sur Yelp, Flickr, et Twitter : la géolocalisation jointe aux posts d'utilisateurs, le timestamp des posts, et le style d'écriture capturé par modèles de langage. Nous montrons que, parmi ces trois caractéristiques l'emplacement des posts est l'attribut le plus puissant pour identifier les comptes qui appartiennent au même utilisateur dans différents sites. Lorsque nous combinons les trois attributs, la précision de l'identification des comptes Twitter qui appartiennent à un ensemble d'utilisateurs Flickr est comparable à celui des attaques existantes qui exploitent les noms d'écran. Notre attaque peut identifier 37% de comptes en plus par rapport aux schémas basés sur le nom d'écran lorsque nous cherchons plutôt les correspondance entre Yelp et Twitter. Nos résultats ont d'importantes conséquences pour la protection de la vie privée : ils présentent une nouvelle classe d'attaques qui exploitent le fait que les utilisateurs ont tendance à supposer que, si ils maintiennent des personnages différents avec des noms différents sur différents réseaux sociaux, les comptes ne peuvent pas être matchés ensemble; tandis que nous démontrons que les posts eux-mêmes peuvent fournir suffisamment d'informations pour faire la correspondance.

Détection et caractérisation des usurpateurs d'identité

Les gens sont conscients que les attaquants imitent des comptes de réseaux sociaux. En dehors de quelques témoignages anecdotiques, toutefois, il n'y a pas eu de qualification approfondie de l'usurpation d'identité aujourd'hui dans les réseaux sociaux. Nous proposons une technique en deux étapes qui détecte les comptes usurpateurs. La première étape retourne les comptes qui dépeignent la même personne dans un réseau

social. La deuxième étape détecte quel compte est un imposteur. Les méthodes traditionnelles pour détecter les faux comptes marchent mal pour détecter les usurpateurs d'identité. Nous montrons que pour la détection d'usurpateurs, nous devons à bâtir des méthodes qui exploitent des attributs qui caractérisent des paires de comptes plutôt que des comptes uniques comme cela a été fait jusqu'à présent. Nous faisons une étude de caractérisation d'environ 5 693 cas d'attaques d'usurpation d'identité que nous avons détecté sur Twitter. Nous avons constaté que les attaques d'usurpation d'identité ne ciblent pas uniquement les célébrités mais aussi des utilisateurs moins populaires sur Twitter. En outre, leur principal objectif est d'échapper à la détection de faux comptes de Twitter plutôt que d'utiliser les comptes pour des attaques d'ingénierie sociale. Nos résultats révèlent un nouveau type d'attaques d'usurpation d'identité que peut avoir un impact négatif sur l'image en ligne de n'importe quel utilisateur, et pas seulement celles des célébrités.

References

Bibliography

[1] Bing Maps API. http://www.microsoft.com/maps/developers/web.aspx.

[2] geonames.org. http://geonames.org.

[3] Phash. http://www.phash.org.

[4] Spokeo. http://www.spokeo.com/.

[5] Yahoo! placemaker. http://developer.yahoo.com/geo/placemaker/.

[6] Alessandro Acquisti, Ralph Gross, and Fred Stutzman. Faces of facebook: Privacy in the age of augmented reality. In *BlackHat*, 2011.

[7] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 835–844, New York, NY, USA, 2007. ACM.

[8] Luca Maria Aiello and Giancarlo Ruffo. Lotusnet: Tunable privacy for distributed online social network services. *Computer Communications*, 35(1):75–88, 2012.

[9] Get better results with less effort with Mechanical Turk Masters – The Mechanical Turk blog. http://bit.ly/112GmQI.

[10] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Robin Dunbar. Dynamics of personal social relationships in online social networks: A study on twitter. In *Proceedings of the First ACM Conference on Online Social Networks*, COSN '13, pages 15–26, New York, NY, USA, 2013. ACM.

[11] Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 3rd Annual ACM Web Science Conference*, WebSci '12, pages 33–42, New York, NY, USA, 2012. ACM.

[12] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[13] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: Improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 61–70, New York, NY, USA, 2010. ACM.

[14] Randy Baden, Adam Bender, Neil Spring, Bobby Bhattacharjee, and Daniel Starin. Persona: An online social network with user-defined privacy. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, pages 135–146, New York, NY, USA, 2009. ACM.

[15] Marco Balduzzi, Christian Platzer, Thorsten Holz, Engin Kirda, Davide Balzarotti, and Christopher Kruegel. Abusing social networks for automated user profiling. In *RAID*, 2010.

[16] Sergey Bartunov, Anton Korshunov, Seung-Taek Park, Wonho Ryu, and Hyungdong Lee. Joint link-attribute user identity resolution in online social networks. In *SNA-KDD Workshop*, 2012.

[17] Fabricio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgilio Almeida. Detecting spammers on Twitter. In *Proceedings of the Seventh Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, July 2010.

[18] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '09, pages 49–62, New York, NY, USA, 2009. ACM.

[19] Claudio Bettini, X. Sean Wang, and Sushil Jajodia. Protecting privacy against location-based personal identification. In *Proceedings of the Second VDLB International Conference on Secure Data Management*, SDM'05, pages 185–199, Berlin, Heidelberg, 2005. Springer-Verlag.

[20] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data*, 1(1), March 2007.

[21] Parantapa Bhattacharya, Saptarshi Ghosh, Juhi Kulshrestha, Mainack Mondal, Muhammad Bilal Zafar, Niloy Ganguly, and Krishna P. Gummadi. Deep twitter diving: Exploring topical groups in microblogs at scale. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, pages 197–210, New York, NY, USA, 2014. ACM.

[22] Leyla Bilge, Thorsten Strufe, Davide Balzarotti, and Engin Kirda. All your contacts are belong to us: Automated identity theft attacks on social networks. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 551–560, New York, NY, USA, 2009. ACM.

[23] Sonja Buchegger, Doris Schiöberg, Le Hung Vu, and Anwitaman Datta. PeerSoN: P2P social networking - early experiences and insights. In *Proceedings of the Second ACM Workshop on Social Network Systems Social Network Systems 2009, co-located with Eurosys 2009*, pages 46–52, Nürnberg, Germany, March 31, 2009.

[24] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. "you might also like: " privacy risks of collaborative filtering. In *IEEE Symposium on Security and Privacy*, pages 231–246. IEEE Computer Society, 2011.

[25] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 15–15, Berkeley, CA, USA, 2012. USENIX Association.

[26] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI*, 2012.

[27] Claude Castelluccia, Chaabane Abdelberi, Markus Dürmuth, and Daniele Perito. When privacy meets security: Leveraging personal information for password cracking. *CoRR*, abs/1304.6584, 2013.

[28] Sung-hyuk Cha. Comprehensive survey on distance / similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007.

[29] Abdelberi Chaabane, Gergely Acs, and Mohamed A. Kaafar. You are what you like! information leakage through users' interests. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium*, NDSS '12, 2012.

[30] Rhonda Chaytor. Privacy advisors for personal information management. In *SIGIR Workshop on Personal Information Management*, SIGIR Workshop on Personal Information Management, 2006.

[31] Terence Chen, Mohamed Ali Kaafar, Arik Friedman, and Roksana Boreli. Is more always merrier?: a deep dive into online social footprints. In *WOSN*, 2012.

[32] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 759–768, New York, NY, USA, 2010. ACM.

[33] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, 2010.

[34] Peter Christen. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Data-centric systems and applications. Springer, 2012.

[35] Peter Christen. A survey of indexing techniques for scalable record linkage and deduplication. *Knowledge and Data Engineering, IEEE Transactions on*, 24(9):1537–1555, Sept 2012.

[36] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *IIWeb*, 2003.

[37] Sunny Consolvo, Jaeyeon Jung, Ben Greenstein, Pauline Powledge, Gabriel Maganis, and Daniel Avrahami. The wi-fi privacy ticker: Improving awareness & control of personal information exposure on wi-fi. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, Ubicomp '10, pages 321–330, New York, NY, USA, 2010. ACM.

[38] Microsoft Survey: Online 'Reputation' Counts, 2010. http://www.internetnews.com/webcontent/article.php/3861241/Microsoft+Survey+Online+Reputation+Counts.htm.

[39] David J. Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. Mapping the world's photos. In *Proceedings of the 18th international conference on World Wide Web*, WWW '09, pages 761–770, 2009.

[40] Emiliano De Cristofaro, Claudio Soriente, Gene Tsudik, and Andrew Williams. Hummingbird: Privacy at the time of twitter. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 285–299, Washington, DC, USA, 2012. IEEE Computer Society.

[41] Mathieu Cunche, Dali Kaafar, and Roksana Boreli. I know who you will meet this evening! linking wireless devices using wi-fi probe requests. In *13th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, San Francisco, California, June 2012.

[42] L.A. Cutillo, R. Molva, and T. Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. *Communications Magazine, IEEE*, 47(12):94–101, Dec 2009.

[43] Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. Identification via location-profiling in gsm networks. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*, WPES '08, pages 23–32, New York, NY, USA, 2008. ACM.

[44] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, January 2007.

[45] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

[46] Csilla Farkas and Sushil Jajodia. The inference problem: A survey. *SIGKDD Explor. Newsl.*, 4(2):6–11, December 2002.

[47] Stephen E. Fienberg. Comment on gates: Toward a reconceptualization of confidentiality protection in the context of linkages with administrative records. *Journal of Privacy and Confidentiality*, 3:65, 2011.

[48] Employers Using Social Networks for Screening Applicants, 2008. http://wikibin.org/articles/employers-using-social-networks-for-screening-applicants.html.

[49] M. Fredrikson and B. Livshits. Repriv: Re-imagining content personalization and in-browser privacy. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 131–146, May 2011.

[50] Julien Freudiger, Raoul Neu, and Jean pierre Hubaux. Private sharing of user location over online social networks, 2010.

[51] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. Evaluating the privacy risk of location-based services. In *Proceedings of the 15th International Conference on Financial Cryptography and Data Security*, FC'11, pages 31–46, Berlin, Heidelberg, 2012. Springer-Verlag.

[52] Gerald Friedland, Gregor Maier, Robin Sommer, and Nicholas Weaver. Sherlock holmes' evil twin: On the impact of global inference for online privacy. In *Proceedings of the 2011 Workshop on New Security Paradigms Workshop*, NSPW '11, pages 105–114, New York, NY, USA, 2011. ACM.

[53] Gerald Friedland, Gregor Maier, Robin Sommer, and Nicholas Weaver. Sherlock Holmes' evil twin: on the impact of global inference for online privacy. In *Proceedings of the 2011 Workshop on New Security Paradigms Workshop*, pages 105–114, 2011.

[54] Gerald Friedland and Robin Sommer. Cybercasing the joint: On the privacy implications of geo-tagging. In *Proceedings of the 5th USENIX Conference on Hot Topics in Security*, HotSec'10, pages 1–8, Berkeley, CA, USA, 2010. USENIX Association.

[55] Gerald W. Gates. How uncertainty about privacy and confidentiality is hampering efforts to more effectively use administrative records in producing u.s. national statistics. *Journal of Privacy and Confidentiality*, 2011.

[56] Daniel Gayo Avello. All liaisons are dangerous when all your friends are known to us. In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, HT '11, pages 171–180, 2011.

[57] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. Private queries in location based services: Anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 121–132, New York, NY, USA, 2008. ACM.

[58] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, pages 77–80, New York, NY, USA, 2006. ACM.

[59] Philippe Golle. A study on the re-identifiability of dutch citizens. In *3rd Hot Topics in Privacy Enhancing Technologies*, HotPETs 2010, 2010.

[60] Philippe Golle and Kurt Partridge. On the anonymity of home/work location pairs. In *Proceedings of the 7th International Conference on Pervasive Computing*, Pervasive '09, pages 390–397, Berlin, Heidelberg, 2009. Springer-Verlag.

[61] Neil Zhenqiang Gong, Eui Chul, Richard Shin, Ameet Talwalkar, Emil Stefanov, Lester Mackey, Elaine (runting Shi, Ling Huang, and Dawn Song. Jointly predicting links and inferring attributes using a social-attribute network (san. In *In SNA-KDD*, 2-12.

[62] Neil Zhenqiang Gong, Wenchang Xu, Ling Huang, Prateek Mittal, Emil Stefanov, Vyas Sekar, and Dawn Song. Evolution of social-attribute networks: Measurements, modeling, and implications using google+. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 131–144, New York, NY, USA, 2012. ACM.

[63] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, MobiSys '08, pages 40–53, New York, NY, USA, 2008. ACM.

[64] Virgil Griffith and Markus Jakobsson. Messin' with texas deriving mother's maiden names using public records. In *Proceedings of the Third International Conference on Applied Cryptography and Network Security*, ACNS'05, pages 91–103, Berlin, Heidelberg, 2005. Springer-Verlag.

[65] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, WPES '05, pages 71–80, New York, NY, USA, 2005. ACM.

[66] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, pages 31–42, New York, NY, USA, 2003. ACM.

[67] Saikat Guha, Bin Cheng, and Paul Francis. Challenges in measuring online advertising systems. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 81–87, New York, NY, USA, 2010. ACM.

[68] Saikat Guha, Bin Cheng, and Paul Francis. Privad: Practical privacy in online advertising. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pages 13–13, Berkeley, CA, USA, 2011. USENIX Association.

[69] Seungyeop Han, Jaeyeon Jung, and David Wetherall. A study of third-party tracking by mobile apps in the wild.

[70] Bing-Zhe He, Chien-Ming Chen, Yi-Ping Su, and Hung-Min Sun. A defence scheme against identity theft attack based on multiple social networks. *Expert Syst. Appl.*, 41(5):2345–2352, April 2014.

[71] Jianming He, Wesley W. Chu, and Zhenyu (Victor) Liu. Inferring privacy information from social networks. In *Proceedings of the 4th IEEE International Conference on Intelligence and Security Informatics*, ISI'06, pages 154–165, Berlin, Heidelberg, 2006. Springer-Verlag.

[72] Brent Hecht, Lichan Hong, Bongwon Suh, and Ed H. Chi. Tweets from justin bieber's heart: the dynamics of the location field in user profiles. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 237–246, 2011.

[73] Lee Humphreys, Phillipa Gill, and Balachander Krishnamurthy. *How much is too much? Privacy issues on Twitter*, pages 1–29. ACM Press, 2010.

[74] M.P. Hussey, P.A. Baranov, T.E. McArdle, T.M. Boesenberg, and B. Duggal. Distributed personal information aggregator, January 14 2010. US Patent App. 12/384,098.

[75] Twitter Reporting impersonation accounts, 2014. https://support.twitter.com/articles/20170142-reporting-impersonation-accounts.

[76] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying users across social tagging systems. In *ICWSM*, 2011.

[77] Danesh Irani, Steve Webb, Kang Li, and Calton Pu. Large online social footprints–an emerging threat. In *SocialCom*, 2009.

[78] Danesh Irani, Steve Webb, Kang Li, and Calton Pu. Modeling unintended personal-information leakage from multiple online social networks. *IEEE Internet Computing*, 15(3):13–19, May 2011.

[79] Collin Jackson, Andrew Bortz, Dan Boneh, and John C. Mitchell. Protecting browser state from web privacy attacks. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 737–744, New York, NY, USA, 2006. ACM.

[80] T.N. Jagatic, N.A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *Communications of the ACM*, 50(10), 2007.

[81] Markus Jakobsson and Steven Myers. *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. Wiley-Interscience, 2006.

[82] ChristianS. Jensen, Hua Lu, and ManLung Yiu. Location privacy techniques in client-server architectures. In Claudio Bettini, Sushil Jajodia, Pierangela Samarati, and X.Sean Wang, editors, *Privacy in Location-Based Applications*, volume 5599 of *Lecture Notes in Computer Science*, pages 31–58. Springer Berlin Heidelberg, 2009.

[83] Carter Jernigan and Behram F. T. Mistree. Gaydar: Facebook friendships expose sexual orientation. *First Monday*, 14(10), 2009.

[84] Lei Jin, Hassan Takabi, and James B.D. Joshi. Towards active detection of identity clone attacks on online social networks. In *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, CODASPY '11, pages 27–38, New York, NY, USA, 2011. ACM.

[85] Jeff Jonas and Jim Harper. *Effective Counterterrorism and the Limited Role of Predictive Data Mining*. 2006.

[86] Karen SpŁrck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[87] Jaeyeon Jung, Anmol Sheth, Ben Greenstein, David Wetherall, Gabriel Maganis, and Tadayoshi Kohno. Privacy oracle: A system for finding application leaks with black box differential testing. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, pages 279–288, New York, NY, USA, 2008. ACM.

[88] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preventing location-based identity inference in anonymous spatial queries. *Knowledge and Data Engineering, IEEE Transactions on*, 19(12):1719–1733, Dec 2007.

[89] Anderson Keith B., Erik Durbin, and Michael A. Salinger. Identity theft. *Journal of Economic Perspectives*, 2008.

[90] Vlado Keselj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. In *Pacific Association for Computational Linguistics*, 2003.

[91] Sheila Kinsella, Vanessa Murdock, and Neil O'Hare. "i'm eating a sandwich in glasgow": modeling locations with tweets. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, SMUC '11, pages 61–68, 2011.

[92] Klout, 2014. http://klout.com/.

[93] G. Kontaxis, I. Polakis, S. Ioannidis, and E.P. Markatos. Detecting social network profile cloning. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 295–300, March 2011.

[94] Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data Knowl. Eng.*, 69(2):197–210, February 2010.

[95] Nitish Korula and Silvio Lattanzi. An efficient reconciliation algorithm for social networks. *PVLDB*, 7(5):377–388, 2014.

[96] G Kossinets and D Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[97] David Kotz, Tristan Henderson, and Ilya Abyzov. CRAWDAD data set dartmouth/campus (v. 2004-12-18). Downloaded from http://crawdad.org/dartmouth/campus, December 2004.

[98] Balachander Krishnamurthy, Phillipa Gill, and Martin Arlitt. A few chirps about twitter. In *Proceedings of the First Workshop on Online Social Networks*, WOSN '08, pages 19–24, New York, NY, USA, 2008. ACM.

[99] Balachander Krishnamurthy, Delfina Malandrino, and Craig E. Wills. Measuring privacy loss and the impact of privacy protection in web browsing. In *Proceedings of the 3rd Symposium on Usable Privacy and Security*, SOUPS '07, pages 52–63, New York, NY, USA, 2007. ACM.

[100] Balachander Krishnamurthy and Craig Wills. Privacy diffusion on the web: A longitudinal perspective. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 541–550, New York, NY, USA, 2009. ACM.

[101] Balachander Krishnamurthy and Craig E. Wills. Generating a privacy footprint on the internet. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 65–70, New York, NY, USA, 2006. ACM.

[102] Balachander Krishnamurthy and Craig E. Wills. Characterizing privacy in online social networks. In *Proceedings of the First Workshop on Online Social Networks*, WOSN '08, pages 37–42, New York, NY, USA, 2008. ACM.

[103] Balachander Krishnamurthy and Craig E. Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2Nd ACM Workshop on Online Social Networks*, WOSN '09, pages 7–12, New York, NY, USA, 2009. ACM.

[104] Balachander Krishnamurthy and Craig E. Wills. Privacy leakage in mobile online social networks. In *Proceedings of the 3rd Conference on Online Social Networks*, WOSN'10, pages 4–4, Berkeley, CA, USA, 2010. USENIX Association.

[105] John Krumm. Inference attacks on location tracks. In *Proceedings of the 5th International Conference on Pervasive Computing*, PERVASIVE'07, pages 127–143, Berlin, Heidelberg, 2007. Springer-Verlag.

[106] John Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, August 2009.

[107] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 611–617, New York, NY, USA, 2006. ACM.

[108] Shamanth Kumar, Reza Zafarani, and Huan Liu. Understanding user migration patterns in social media. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011.

[109] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[110] Sebastian Labitzke, Irina Taranu, and Hannes Hartenstein. What's in a name?: An unsupervised approach to link users across communities. In *SNA-KDD*, SNA-KDD '11, 2011.

[111] Cliff A.C. Lampe, Nicole Ellison, and Charles Steinfield. A familiar face(book): Profile elements as signals in an online social network. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 435–444, New York, NY, USA, 2007. ACM.

[112] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 462–470, New York, NY, USA, 2008. ACM.

[113] Fengjun Li, Jake Chen, Xukai Zou, and Peng Liu. New privacy threats in healthcare informatics: When medical records join the web. In *BIOKDD workshop*, BIOKDD workshop 2010, 2010.

[114] Michael D. Lieberman and Jimmy Lin. You are where you edit: Locating wikipedia contributors through edit histories. In Eytan Adar, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng, editors, *ICWSM*. The AAAI Press, 2009.

[115] Jack Lindamood, Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Inferring private information using social network data. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 1145–1146, New York, NY, USA, 2009. ACM.

[116] Jack Lindamood and Murat Kantarcioglu. Inferring private information using social network data. Technical report, 2008.

[117] Jing Liu, Young-In Song, and Chin-Yew Lin. Competition-based user expertise score estimation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 425–434, New York, NY, USA, 2011. ACM.

[118] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a name?: An unsupervised approach to link users across communities. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 495–504, New York, NY, USA, 2013. ACM.

[119] Yabing Liu, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing facebook privacy settings: User expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 61–70, New York, NY, USA, 2011. ACM.

[120] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 2004.

[121] Chris Y.T. Ma, David K.Y. Yau, Nung Kwan Yip, and Nageswara S.V. Rao. Privacy vulnerability of published anonymous mobility traces. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 185–196, New York, NY, USA, 2010. ACM.

[122] Anshu Malhotra, Luam Totti, Wagner Meira, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying user footprints in different online social networks. In *CSOSN*, 2012.

[123] Sergio Mascetti, Dario Freni, Claudio Bettini, X. Sean Wang, and Sushil Jajodia. Privacy in geo-social networks: Proximity notification with untrusted service providers and curious buddies. *The VLDB Journal*, 20(4):541–566, August 2011.

[124] Jonathan R. Mayer and John C. Mitchell. Third-party web tracking: Policy and technology. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 413–427, Washington, DC, USA, 2012. IEEE Computer Society.

[125] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. Birds of a feather: Homophily in social networks, 2001. This article consists of 30 page(s).

[126] mediabistro, 2009. https://www.mediabistro.com/alltwitter/was-twitter-right-to-suspend-christopher-walken_b5021.

[127] Matthew Michelson and Sofus A. Macskassy. Discovering users' topics of interest on twitter: A first look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, AND '10, pages 73–80, New York, NY, USA, 2010. ACM.

[128] Mishari Al Mishari and Gene Tsudik. Exploring linkability of user reviews. In *ESORICS*, 2012.

[129] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 29–42, New York, NY, USA, 2007. ACM.

[130] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: Inferring user profiles in online social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 251–260, New York, NY, USA, 2010. ACM.

[131] Sonal Mittal. User privacy and the evolution of third-party tracking mechanisms on the world wide web. In *SSRN*. SSRN, 2010.

[132] Anthony Miyazaki. Online privacy and the disclosure of cookie use: Effects on consumer trust and anticipated patronage. In *Journal of Public Policy & Marketing*. Journal of Public Policy & Marketing, 2008.

[133] Abedelaziz Mohaisen, Dowon Hong, and DaeHun Nyang. Privacy in location based services: Primitives toward the solution. In Jinhwa Kim, Dursun Delen, Jinsoo Park, Franz Ko, and Yun Ji Na, editors, *NCM (1)*, pages 572–579. IEEE Computer Society, 2008.

[134] Marti Motoyama and George Varghese. I seek you: searching and matching individuals in social networks. In *WIDM*, 2009.

[135] Mihir Nanavati, Nathan Taylor, William Aiello, and Andrew Warfield. Herbert west: deanonymizer. In *HotSec*, 2011.

[136] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the feasibility of internet-scale author identification. In *Proceedings of the 33st IEEE Symposium on Security and Privacy (S&P)*, 2012.

[137] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, 2009.

[138] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakhani, Michael Hamburg, and Dan Boneh. Location privacy via private proximity testing. In *NDSS*, 2011.

[139] M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, December 2011.

[140] Carlton T Northern and Michael L Nelson. An unsupervised approach to discovering and disambiguating social media profiles. 2007.

[141] Dan Noyes, 2014. http://zephoria.com/social-media/top-15-valuable-facebook-statistics/.

[142] Facebook Terms of Service, 2014. https://www.facebook.com/legal/terms.

[143] Thomas Oscherwitz. Synthetic identity fraud: Unseen identity challenge. *BANK. SECURITY NEWS*, 2005.

[144] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, January 2008.

[145] Ponnurangam Kumaraguru Paridhi Jain and Anupam Joshi. @i seek 'fb.me': Identifying users across multiple online social networks. In *WoLE*, 2013.

[146] Peekyou.com. http://www.peekyou.com/.

[147] Olga Peled, Michael Fire, Lior Rokach, and Yuval Elovici. Entity matching in online social networks. In *SocialCom*, pages 339–344. IEEE, 2013.

[148] Daniele Perito, Claude Castelluccia, Mohamed Ali Kâafar, and Pere Manils. How unique and traceable are usernames? In *PETS*, 2011.

[149] B. Picart. Improved Phone Posterior Estimation Through K-NN and MLP-Based Similarity. Technical report, Idiap Research Institute, 2009.

[150] Royal Pingdom, 2012. http://royal.pingdom.com/2012/06/18/how-many-sites-have-facebook-integration-youd-be-surprised/.

[151] Murillo Pontual, Andreas Gampe, Omar Chowdhury, Bazoumana Kone, Md. Shamim Ashik, and William H. Winsborough. The privacy in the time of the internet: Secrecy vs transparency. In *Proceedings of the Second ACM Conference on Data and Application Security and Privacy*, CODASPY '12, pages 133–140, New York, NY, USA, 2012. ACM.

[152] Adrian Popescu and Gregory Grefenstette. Mining user home location and gender from flickr tags. In *ICWSM 2010 : International Conference on Weblogs and Social Media*, 2010.

[153] Foster J. Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 445–453, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[154] E. Raad, R. Chbeir, and A. Dipanda. User profile matching in social networks. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, pages 297–304, Sept 2010.

[155] David Recordon and Drummond Reed. Openid 2.0: A platform for user-centric identity management. In *Proceedings of the Second ACM Workshop on Digital Identity Management*, DIM '06, pages 11–16, New York, NY, USA, 2006. ACM.

[156] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, NSDI'12, pages 12–12, Berkeley, CA, USA, 2012. USENIX Association.

[157] Carmen Ruiz Vicente, Dario Freni, Claudio Bettini, and Christian S. Jensen. Location-related privacy in geo-social networks. *IEEE Internet Computing*, 15(3):20–27, May 2011.

[158] T. Scott Saponas, Jonathan Lester, Carl Hartung, Sameer Agarwal, and Tadayoshi Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 5:1–5:16, Berkeley, CA, USA, 2007. USENIX Association.

[159] Robert Schmid. Salesforce service cloud – featuring activision, September 2012. http://www.youtube.com/watch?v=eT6iHEdnKQ4&feature=relmfu.

[160] seattlepi, 2010. http://www.seattlepi.com/local/sound/article/Racism-and-Twitter-impersonation-prompt-lawsuit-893555.php.

[161] 6 Reasons Social Media Is Critical To Your SEO, 2013. http://socialmediatoday.com/stephaniefrasco/1901891/6-reasons-social-media-critical-your-seo.

[162] Naveen Kumar Sharma, Saptarshi Ghosh, Fabricio Benevenuto, Niloy Ganguly, and Krishna Gummadi. Inferring who-is-who in the twitter social network. *SIGCOMM Comput. Commun. Rev.*, 42(4):533–538, September 2012.

[163] Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Boudec. Quantifying location privacy: The case of sporadic location exposure. In Simone Fischer-HÃijbner and Nicholas Hopper, editors, *Privacy Enhancing Technologies*, volume 6794 of *Lecture Notes in Computer Science*, pages 57–76. Springer Berlin Heidelberg, 2011.

[164] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA, 2011. IEEE Computer Society.

[165] Andrew C. Simpson. On privacy and public data: a study of data.gov.uk. *Journal of Privacy and Confidentiality*, 2011.

[166] Social Intelligence Corp. http://www.socialintel.com/.

[167] Mudhakar Srivatsa and Michael Hicks. Deanonymizing mobility traces: Using social network as a side-channel. In *CCS*, October 2012.

[168] Jessica Staddon. Finding "hidden" connections on linkedin an argument for more pragmatic social network privacy. In *Proceedings of the 2Nd ACM Workshop on Security and Artificial Intelligence*, AISec '09, pages 11–14, New York, NY, USA, 2009. ACM.

[169] Tao Stein, Erdong Chen, and Karan Mangla. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, SNS '11, pages 8:1–8:8, New York, NY, USA, 2011. ACM.

[170] Andreas Stolcke. Srilm - an extensible language modeling toolkit. In *Proceedings of Int'l conference on Spoken Language Processing*, 2002.

[171] Dustin W. Stout, 2013. http://dustn.tv/active-users-2013/.

[172] San-Tsai Sun, Yazan Boshmaf, Kirstie Hawkey, and Konstantin Beznosov. A billion keys, but few locks: The crisis of web single sign-on. In *Proceedings of the 2010 Workshop on New Security Paradigms*, NSPW '10, pages 61–72, New York, NY, USA, 2010. ACM.

[173] Latanya Sweeney. Weaving technology and policy together to maintain confidentiality. *Journal of Law, Medicine, and Ethics*, 25(2–3):98–110, 1997.

[174] K. A. Taipale. Data mining and domestic security: Connecting the dots to make sense of data. *Columbia Science and Technology Law Review*, 5(2), 2003.

[175] Article 29 data protection working party. Letter to the online advertising industry., 2012. https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/privacy-considerations-of-online-behavioural-tracking/at_download/fullReport.

[176] ThinkProgress, 2013. http://thinkprogress.org/security/2013/12/31/3108661/10-biggest-privacy-security-breaches-rocked-2013/.

[177] Kurt Thomas, Chris Grier, and David M. Nicol. Unfriendly: Multi-party privacy risks in social networks. In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 236–252, Berlin, Heidelberg, 2010. Springer-Verlag.

[178] TomHCAnderson, 2009. http://www.tomhcanderson.com/2009/07/09/overlap-among-major-social-network-services/.

[179] Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. In *NDSS*. The Internet Society, 2010.

[180] M. Tranmer and M. Elliot. Binary logistic regression. *Cathie Marsh for Census and Survey Research, Paper 2008-20*.

[181] B. Viswanath, M. Mondal, A. Clement, P. Druschel, K.P. Gummadi, A. Mislove, and A. Post. Exploring the design space of social network-based sybil defenses. In *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, pages 1–8, Jan 2012.

[182] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2Nd ACM Workshop on Online Social Networks*, WOSN '09, pages 37–42, New York, NY, USA, 2009. ACM.

[183] Bimal Viswanath, Ansley Post, Krishna P. Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. In *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, pages 363–374, New York, NY, USA, 2010. ACM.

[184] J. Vosecky, Dan Hong, and V.Y. Shen. User identification across multiple social networks. In *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, pages 360–365, July 2009.

[185] Gang Wang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam J. Metzger, Haitao Zheng, and Ben Y. Zhao. Social turing tests: Crowdsourcing sybil detection. In *NDSS*. The Internet Society, 2013.

[186] Gang Wang, Manish Mohanlal, Christo Wilson, Xiao Wang, Miriam J. Metzger, Haitao Zheng, and Ben Y. Zhao. Social turing tests: Crowdsourcing sybil detection. In *NDSS*, 2013.

[187] Zachary Weinberg, Eric Y. Chen, Pavithra Ramesh Jayaraman, and Collin Jackson. I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 147–161, Washington, DC, USA, 2011. IEEE Computer Society.

[188] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 261–270, New York, NY, USA, 2010. ACM.

[189] D. Wetherall, D. Choffnes, B. Greenstein, S. Han, P. Hornyack, J. Jung, S. Schechter, and X. Wang. Privacy revelations for web and mobile apps. In *Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems*, HotOS'13, pages 21–21, Berkeley, CA, USA, 2011. USENIX Association.

[190] Christo Wilson, Bryce Boe, Alessandra Sala, Krishna P.N. Puttaswamy, and Ben Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys '09, pages 205–218, New York, NY, USA, 2009. ACM.

[191] William E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[192] William E. Winkler, William E Winkler, and Nov P. Overview of record linkage and current research directions. Technical report, Bureau of the Census, 2006.

[193] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.

[194] Ting-Fang Yen, Yinglian Xie, Fang Yu, Roger Peng Yu, and MartŠn Abadi. Host fingerprinting and tracking on the web: Privacy and security implications. In *NDSS*. The Internet Society, 2012.

[195] Gae-won You, Seung-won Hwang, Zaiqing Nie, and Ji-Rong Wen. Socialsearch: enhancing entity search with social network matching. In *EDBT/ICDT*, 2011.

[196] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. Sybilguard: Defending against sybil attacks via social networks. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, pages 267–278, New York, NY, USA, 2006. ACM.

[197] Reza Zafarani and Huan Liu. Connecting corresponding identities across communities. In Eytan Adar, Matthew Hurst, Tim Finin, Natalie S. Glance, Nicolas Nicolov, and Belle L. Tseng, editors, *ICWSM*. The AAAI Press, 2009.

[198] Reza Zafarani and Huan Liu. Connecting users across social media sites: A behavioral-modeling approach. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 41–49, New York, NY, USA, 2013. ACM.

[199] Hui Zang and Jean Bolot. Anonymization of location data does not work: a large-scale measurement study. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 145–156, New York, NY, USA, 2011. ACM.

[200] Chao Michael Zhang and Vern Paxson. Detecting and analyzing automated activity on twitter. In *Proceedings of the 12th International Conference on Passive and Active Measurement*, PAM'11, pages 102–111, Berlin, Heidelberg, 2011. Springer-Verlag.

[201] Elena Zheleva and Lise Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 531–540, 2009.