

Identifying Social Media Users Across Platforms and Highlighting Privacy Concerns

T.M.A. Mulder

Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2017

Abstract

The aim of this project is to help social media users understand that sharing certain types of information online makes it possible for third parties like advertisers and malicious actors to link supposedly anonymous Facebook and Twitter accounts of the same person together. To accomplish this, I conducted a qualitative study to find the design requirements of a feedback system that informs and helps users to avoid online identity resolution. I also examined several approaches to connect social media accounts together.

Previous research succeeded in finding users through cross-referencing of their other social media content on Twitter. I explored this type of behaviour among Twitter users on a large set of Tweets ($= 48.2$ million), and show that 2.98% of the Tweets ($N = 1.76$ million) includes a Facebook username. Furthermore, I show that 31.42% of the users in a smaller dataset ($N = 138,097$) use the same username for Twitter and Facebook.

In addition, I developed novel techniques to find and match users. Facebook requires user IDs to retrieve user information from their API. To query users by their username, I developed a username to user ID converter with impeccable results. Furthermore, a proof-of-concept for the use of browser automation has successfully shown how Facebook's extensive in-browser search can be utilised without intervention from a person. This uncovers a large set of new search query options to find users, such as full names and URLs, despite the decreased number of Facebook's API functions. Searching using full names included the correct user in the candidate set in 40% ($N = 1,959$) of the queried users. To match accounts, I demonstrate the performance of perceptual hashing and facial recognition.

Acknowledgements

My greatest gratitude goes out to my supervisors Dr. Kami Vaniea and Dr. Liane Guillou, for providing me with their endless knowledge and support during this project. Also, I'd like to thank Dr. Catherine Crompton for her help with the preparations of the focus group. Further, I'd like to thank P. Jain, P. Kumaraguru, and A. Joshi for providing me their dataset, and Brainnwave Inc. for providing me with their knowledge and Twitter data.

Of course, I could not have done this master's without the support of my parents Halbo Mulder and Marianne Willemsen. My girlfriend Anne Koopman has always been there for me as well, she provided me with great support, annotations for the images, and suggestions for the report.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*T.M.A. Mulder*)

In memory of my dear uncle Dirk Ringoir, who taught me the first steps to bring
me where I am now.

Table of Contents

List of Figures	ix
List of Tables	xi
1 Introduction	1
1.1 State of the Art	3
1.2 Project Goals	5
1.3 Results and Contributions	6
1.4 Structure of Thesis	6
2 Background	9
2.1 Definitions	9
2.1.1 Formulas for Measuring Performance	10
2.2 Twitter and Facebook	12
2.3 Identity Resolution	15
2.4 Perceptual Hashing	17
2.4.1 Discrete Cosine Transform	17
2.4.2 Radial Hash Projection	18
2.5 Face Recognition	18
2.6 Informing Users	20
3 Design Requirements for the Feedback System	23
3.1 Introduction	23
3.2 Methods	24
3.2.1 Focus Group Questions	24
3.2.2 Analysing the Focus Group	26
3.3 Participants	27
3.3.1 Westin's Privacy Index	31
3.4 Results	32
3.4.1 Staying Anonymous	32
3.4.2 Linking Accounts	32
3.4.3 Explaining Results	33

3.4.4	Presentation of Feedback	34
3.5	Design of Feedback System	34
3.5.1	Potential Information for a Feedback System	35
3.5.2	Explaining Results to the User	36
3.5.3	Presentation of Feedback	36
3.6	Resulting Design Requirements	37
4	Implementations for Identity Resolution	39
4.1	Introduction	39
4.1.1	Machine Info	41
4.2	Retrieving Information	41
4.2.1	Converting Facebook Usernames to Facebook IDs	42
4.2.2	Retrieving Profile Attributes	43
4.2.2.1	Facebook	43
4.2.2.2	Twitter	44
4.2.3	Summary	45
4.3	Finding Accounts	45
4.3.1	Identical Usernames	45
4.3.2	Self-mention of Facebook on Twitter	47
4.3.3	Browser Automation	51
4.3.3.1	Name Search Results	54
4.3.3.2	URL Search Results	55
4.4	Comparing Accounts	56
4.4.1	Username Similarity	56
4.4.2	Perceptual Hash	58
4.4.2.1	Annotation Task	61
4.4.2.2	pHash Results and Discussion	63
4.4.3	Face Recognition	65
4.5	Discussion of Implementations	69
5	Discussion	71
6	Conclusion	75
Bibliography		77
Glossary		85
Appendices		89
A Focus Group Survey and Consent Form		89

List of Figures

2.1	Twitter and Facebook profile page examples	13
2.2	Facebook's content creation screen	14
2.3	Twitter's content creation screen	14
2.4	The Histogram of Gradients of a face (Rojas Q. et al., 2011, p. 10).	20
3.1	Focus group Facebook and Twitter features examples	25
3.2	Focus group linking score mock-up	26
3.3	Demographics of focus groups	29
3.4	Social Media use of focus groups.	29
3.5	Frequency of cross-posting among participants	30
3.6	Information participants of the focus groups publicly share on Twitter	30
3.7	Welcome screen of the feedback system.	38
3.8	Feedback system personalised information	38
3.9	Feedback system FAQ	38
4.1	Identity resolution system overview	41
4.2	Venn-diagram of unintended Twitter and Facebook username overlap	46
4.3	Results of querying users on Facebook by their Twitter username.	46
4.4	Results of scanning 48.2 million Tweets for Facebook links	50
4.5	Facebook on Lynx browser	52
4.6	Photo of Tijger	53
4.7	Username Levenshtein distance histogram	58
4.8	pHash compression example	59
4.9	pHash colour example	59
4.10	pHash mismatches	60
4.11	Facial features example	62
4.12	Identifiable clothing	63
4.13	pHash image results	64
4.14	pHash face, animal, or object results	64
4.15	Face Recognition Precision and Recall	67

List of Tables

2.1	A typical confusion matrix.	11
3.1	Demographics of the focus group participants	28
4.1	pHash results	64
4.2	Face Recognition results	68

Chapter 1

Introduction

Online social media platforms allow users to make connections and share information across the globe. However, over-sharing of information, combined with the possibility of linking information between different platforms, makes users vulnerable to potential privacy risks (Rose, 2011). Users can manage their privacy through several approaches. Firstly, they can use controls provided by the social media platforms to shield some of their information from the public. This may work well in some cases, but these settings may not always be easy to find. Another approach is for users to disconnect their public profile from their identity, for instance by using an alias instead of their real name. Some platforms, however, make this difficult by maintaining a real-name policy. Alternatively, users can choose to post different content to different accounts. For example, information shared on the popular platform Twitter can be considered ‘very public’, as the default setting is to make the content a user creates publicly available (Twitter inc., 2017a). In contrast, users may wish to keep information on other platforms such as Facebook more discreet, for example only to be shared with friends. Information on another platform, LinkedIn, is meant to be issued for professional or career purposes. These platforms thus serve different purposes, and users might wish to keep the information shared on these platforms separated. Importantly, when third parties are able to link the different social media accounts of a person, these privacy managing approaches outlined above may not be sufficient.

Johnson et al. (2012) show that most users of social media have indeed put effort into staying anonymous to outsiders. However, Acquisti and Gross (2006) found that even privacy aware people share great amounts of information on Facebook, worldwide the most popular social media platform (Moreau, 2017).

Stutzman et al. (2013) write that between the years 2005 and 2011, users of Facebook have become more aware of the privacy risks attached to sharing great amounts of information on the platform. On the other hand, the same study revealed that Facebook has successfully developed methods for encouraging users to share more content.

Although users can to some extent manage what they share on Facebook themselves, they are less able to control content others post about them (Garfinkel and Lipford, 2014, p. 82). For example, a user can be tagged in photos and content owned by others without giving consent. Users also appear to have trouble regulating and judging which audience will be able to see their content (Wang et al., 2011). Moreover, a significant proportion of the Facebook users appear to be unaware of any privacy settings, and hence, do not alter them (Acquisti and Gross, 2006). Notably, Facebook has recently updated their platform, and has asked users to review their privacy settings (Goel, 2014).

Although both Facebook and Twitter publish content publicly by default, the use of certain terms to indicate connections on the networks make it seem as if these platforms serve two different purposes. On Facebook, the most common way to create a connection is by submitting a friendship request. To actually form a connection, the other party has to accept this request (two party consent). In contrast, on Twitter, one can *follow* other users, without any consent from that user (one party consent). These terms suggest that Facebook is meant to be used to interact with friends and family, whereas Twitter functions more similar to a public blog. Users of the two platforms might like to keep their content separated, targeted only to the assumed audiences. Being unable to determine the audience (Wang et al., 2011), may lead to frustration and privacy problems among users. Moreover, the audience could unexpectedly grow when a user's Facebook and Twitter accounts are coupled without them knowing. Accidentally publicly posted content or personally identifiable information (PII) included in a user's Facebook profile could suddenly become available to an unintended audience. In particular the coupling of PII to an anonymous Twitter account increases risks for the user. Anonymous opinions suddenly have an identity attached to it. People and organisations not favouring the opinion are now able to take actions against a real person, which puts that individual at risk. In summary, even though some social media users might deliberately want to make their content publicly available, a number of them may like to keep their real identities separated from their content.

The following recent example shows that having the ability to post content anonymously is important to not suppress people's right to freedom of speech. In the beginning of 2017, the U.S. government demanded Twitter to reveal the identity behind a Twitter account spreading unfavourable opinions about U.S. President Donald Trump (Wong, 2017). In reaction to this, Twitter successfully went to court in order to stop the government's pressure.

Although the latter example ended positively for the Twitter user, there are also examples in which the true identity behind an account has been revealed. The American news channel CNN revealed the identity behind a user of the platform Reddit, who had posted an edited video of president Trump (Kaczynski, 2017). In the video, Trump is engaged in a physical fight with someone whose face had been replaced with the CNN logo, presumably by the Reddit user. The video has even been shared on Twitter by the president himself¹. CNN nonetheless argued that the video incites violence against the press (Kaczynski, 2017). In response, they successfully attempted to find the owner of the Reddit account by matching his Reddit content history to a Facebook account, which revealed personal information including an email address and a phone number. They then proceeded to intimidate the user by asking him to remove selected content and apologise for what he did. CNN has been blackmailing the user, stating that "CNN reserves the right to publish his identity" (Kaczynski, 2017) if he would not comply with CNNs requests, now and in the future.

1.1 State of the Art

Over the past years, several methods have been developed to programmatically find and match users of social media across different platforms (e.g. Correa et al., 2012; Jain et al., 2013; Goga et al., 2013). Jain et al. (2013, p. 1259) termed this "Identity Resolution in Online Social Networks". Developing such methods serves several purposes. As shown in the CNN example above, matching accounts leverages personally identifiable information about users and their published content. Moreover, such bundled information can be used in recommendation systems, e.g., those of review websites, e-commerce websites (Ozsoy et al., 2015), or multimedia providers (Cantador et al., 2015). Enterprises and security specialists looking for a person's background also benefit from linked accounts (Jain, 2015).

¹<https://twitter.comrealDonaldTrump/status/881503147168071680>

Bundling information can be seemingly innocent. However, users might be unaware of their data being collected in this way, and they are not being asked for consent. Moreover, there is personal information, like sexual orientation, that users might want to keep private to the greater public (Rader, 2014). Yet, users have no means to control this form of data collection and conglomeration (Rader, 2014).

The development of identity resolution methods is considerably worrying for users' privacy. Especially those who deliberately try to stay anonymous for various reasons, such as the ones mentioned above, are at risk of being deanonymised. On the other hand, Facebook has improved users' privacy protection. They have limited the types of profile attributes that can be retrieved in an automated fashion from their Application Protocol Interface (API) called Graph (Facebook inc., 2014). Twitter ensured that the amount of image EXIF meta-data was reduced (Twitter inc., 2017c) after Borsboom et al. (2010) published their website Please Rob Me², and Jackson and Pesce (2012) their website I Can Stalk U³ showing who is out of town using, among others, the GPS location in EXIF meta-data of images posted to Twitter. In contrast to limiting their API functionality, Facebook's in-browser search, that is available to all logged-in users, has only expanded its functionality over the past few years (e.g. Linshi, 2014; Statt, 2015; Dashevsky, 2017). The latter search engine can search through all of Facebook's public content, including all public posts, specific URLs, and every user's profile.

Although several past studies have explored means for finding and matching profiles across different platforms, few have dealt with what a user can do to prevent this. To the best of my knowledge, merely one study, by Correa et al. (2012), has actually added a user interface to an identity resolution system. Users could log in using their Twitter account, and the system would return the person's potential other social media accounts. Nonetheless, they did not provide feedback to users on how to avoid having their accounts linked in the future. Neither did they include any design requirements for such a system, nor did they conduct a usability evaluation.

²<http://pleaserobme.com/>

³<http://icanstalku.com/>

1.2 Project Goals

The current project aims to deal with three of the problems introduced above. Firstly, to deal with the lack of available user feedback systems, a qualitative analysis has been conducted in the form of two focus groups. The aim of this study was to *find appropriate feedback that will help users prevent being identified across platforms*. In the focus groups, questions were asked to stimulate discussions on several topics. Discussed subjects included keeping a social media account anonymous, methods to link social media accounts, explaining to others how to avoid being linked, and how information in a potential feedback system should be presented.

Secondly, as Facebook's API functionality has changed since many of the studies outlined in the introduction were conducted, this study will *investigate what the current states of Facebook and Twitter are regarding publicly retrievable information from social media users*. Although most of the Twitter data has remained open-source, many of Facebook's previously available API functions have been deprecated since 2015 (Facebook inc., 2014). Moreover, Facebook explicitly requires requesting users using numerical user-IDs, rather than usernames. The current limits for retrieving public Facebook data are explored. Finding those limits include: retrieving publicly available profile attributes using Facebook's API, scanning the source-code of Facebook's HTML pages to find numerical user-IDs, and using browser automation to bypass Facebook's Graph API, to use their wide-ranging browser search instead.

Thirdly, once functions have been developed for retrieving profile information from Facebook, the methods can be used to perform identity resolution. *The current study focuses on finding and matching users' Facebook accounts, given their Twitter username*. Three methods are developed and tested to find accounts: by identical usernames, links to Facebook profiles from Tweets, and name and URL search using browser automation. After a candidate set of accounts was found, an attempt to match them was performed using username similarity, image comparison through perceptual hashing, and face similarity using a face recognition algorithm.

1.3 Results and Contributions

The current study found the design requirements for a system that can provide feedback to users who want to avoid their accounts being linked across different social media platforms. To generate personalised feedback, I designed, developed and implemented several methods to find and match users, which can in turn highlight aspects of their accounts that lack anonymity. The resulting implementations can provide a back-end for all of the the feedback system's personalised feedback requirements.

To deal with Facebook's API changes, including the requirement of a numerical user ID to retrieve user information, I firstly developed an algorithm to convert usernames to IDs with a precision of 1.00. Moreover, a method has been developed to search for users' profile attributes and content through Facebook's extensive browser search. This circumvents the need to use the API, which has less functionality than the browser search engine. The browser search allows, among others, to search for full names and URLs. I show that name search includes the correct user in the candidate set in 40% of the 1,959 queried users. Moreover, I show that URLs included in Tweets can also be searched for through browser automation.

On a large dataset of 48.2 million Tweets and Twitter profile descriptions, I show that nearly 3% of these include a link to Facebook. I also show how these links can be extracted to find the username belonging to the Facebook link. Furthermore, on a ground truth set of Facebook and Twitter account pairs ($N = 138,097$), I show that 31.42% of the users use the same username for Twitter and Facebook, which can also be leveraged to find users.

Finally, I show that calculating the Levenshtein distance between a Twitter username and the candidate set's usernames can be used to rank the candidate set. A confident match can then be calculated using perceptual hashing and facial recognition, of which the performance has been tested and and is reported in this thesis.

1.4 Structure of Thesis

Chapter 2 will provide background on relevant topics, and elaborates on the related work. The remainder of the thesis is structured with the aim of keeping details on similar topics together. Therefore, the methods and outcome of the focus group,

including the resulting design requirements, are discussed in Chapter 3. All of the developed methods, including their results to retrieve, find, and match accounts can be found in Chapter 4.

Chapter 5 starts with an overview of the results, and will then discuss which design requirements have been satisfied by the developed identity resolution methods. Possible future work has also been included in this chapter. Finally, the conclusion of this thesis can be read in Chapter 6.

Chapter 2

Background

Most of the related work comes from Jain et al. (2013), as they provided the most complete set of methods, of which some will be used in the presently proposed research. In addition, most of the definitions provided in their paper on identity resolution apply to the current project. Some of the definitions below were taken verbatim from the project proposal.

2.1 Definitions

Online Social Media Platforms Websites and applications where users create, share, and react to each other's content are called social media platforms (Obar and Wildman, 2015). With more than 2 billion monthly active users as of June 2017 (Welch, 2017), Facebook is considered the most popular social media network. Twitter, another well known social media platform, has 328 million monthly active users (Welch, 2017).

Identity The identity of a social media user can be divided into three main aspects, that is, profile, content, and network (Jain et al., 2013). The profile of a user includes, however is not limited to, attributes such as name, location, and gender. All content created by the user, and content attributes, including the text, time of creation, location, etc., is termed as content. In the case of Twitter, this would be a Tweet. A user's network are the connections a user makes to other users or organisations on the platform. In the case of Twitter, this would be the user's *followers* and *followees*. On Facebook these connections are called *friends*.

Identity Resolution The term “*Identity Resolution in Online Social Networks*”, or *identity resolution* for short, was coined by Jain et al. (2013, p. 1259) as describing the problem of finding a user’s identity on other social networks, given their identity on one social network. This problem can be divided into two subproblems: *identity search*, and *identity matching*.

Identity Search Using the identity attributes of a user as described above, generic search methods can be applied for finding users on other social media networks. The most common attributes to use are those of a user’s profile, such as username, real name, or location (e.g, Motoyama and Varghese, 2009; Irani et al., 2009; Malhotra et al., 2012). Nonetheless, Jain et al. (2013) proposed to exploit the attributes from the content and network dimensions as well. The search for identities yields a candidate set, on which one can perform identity matching. In the current research, the use of a user’s network has not been explored. Nevertheless, I will show novel methods for searching users through their content, including URLs.

Identity Matching Once a set of candidates has been established, similarity measures need to be performed on the candidate identities in order to find the closest match. Several methods for identity matching exist, including matching profile images, measuring similarity of names and usernames (Jain et al., 2013), and measuring similarity on the basis of the time content was created (Goga et al., 2013). The current research investigates username similarity, and advanced image and face comparison methods.

2.1.1 Formulas for Measuring Performance

For the image annotation task in Section 4.4.2.1, a confusion matrix and several formulas were used to calculate performance. A brief overview of how these formulas were calculated is given below.

Confusion Matrix The confusion matrix is defined in Table 2.1. In the annotation task, binary values were calculated per threshold class. These classes denote all image pairs that would be included when the threshold of the comparison technique would be set to that particular threshold. The annotators then marked per class whether these image pairs were correctly classified as belonging together.

Table 2.1: A typical confusion matrix.

		Truth		Total
		P	n	
Prediction	p'	True Positive	False Positive	P'
	n'	False Negative	True Negative	N'
Total		P	N	

See Section 4.4.2.1 for additional details. True positives (TP) are those marked as positive by the annotators, and included in the class. False positives (FP) are those not marked positive by the annotators, but classified as positive for the particular threshold level. False negatives (FN) are the true positives we would miss out on by setting the threshold too high. True negatives (TN) are all image pairs below the threshold not marked as positive by the annotators.

For example, by setting a threshold of > 0.30 , only image pairs with a similarity of > 0.30 are classified as similar by the algorithm. All image pairs manually annotated as similar, with an assigned similarity by the algorithm of > 0.30 are thus true positives. Other image pairs included in this group are false positives. Image pairs with an assigned similarity score of < 0.30 by the algorithm, that are classified as positive by the marker, are the false negatives. The rest, classified as not similar by annotators and not included in the > 0.30 group, are true negatives.

Sensitivity and Specificity From the confusion matrix results, the true negative rate (TNR) (Eq. 2.4), false positive rate (FPR) (Eq. 2.5), and false negative rate (FNR) (Eq. 2.6) can be calculated. Also the positive predictive value (PPV), better known as precision (Eq. 2.1), recall (Eq. 2.2), and their harmonic mean, the F_1 score (Eq. 2.3), can easily be found using those results. For completeness, their definitions are given below.

$$precision = \frac{TP}{TP + FP} \quad (2.1)$$

$$recall = \frac{TP}{TP + FN} \quad (2.2)$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \quad (2.3)$$

$$TNR = \frac{TN}{TN + FP} \quad (2.4)$$

$$FPR = \frac{FP}{FP + TN} \quad (2.5)$$

$$FNR = \frac{FN}{TP + FN} \quad (2.6)$$

2.2 Twitter and Facebook

On both Facebook and Twitter, similar content types are shared. These include text, URLs, photos, videos, and locations. Nevertheless, both content included on the platform by users, as well as the intended audiences, are considerably different. Content included on Twitter can be considered ‘very public’. The platform seems to encourage publicly publishing content, as their mission is to “give everyone the power to create and share ideas and information instantly, without barriers” (Twitter inc., 2017d). Moreover, most of their platform and data is open-source.

Facebook, on the other hand, is more commonly used among friends, family, and acquaintances. On Facebook, the most common connection type is to become *friends* with another user of the platform. This is a mutually accepted connection between two user-profiles. On Twitter, however, users *follow* each other. Anyone can follow any public profile without the need for mutual agreement.

On both platforms, users have semi-permanent *usernames* (Mariconti et al., 2017). Usernames, also called *screen names* on Twitter, are nicknames users give themselves through which they can be identified. A strictly permanent user identifier, internally used by both Facebook and Twitter, is a user’s numerical *user ID*. On both Facebook and Twitter, a user’s *profile page* shows up when requesting

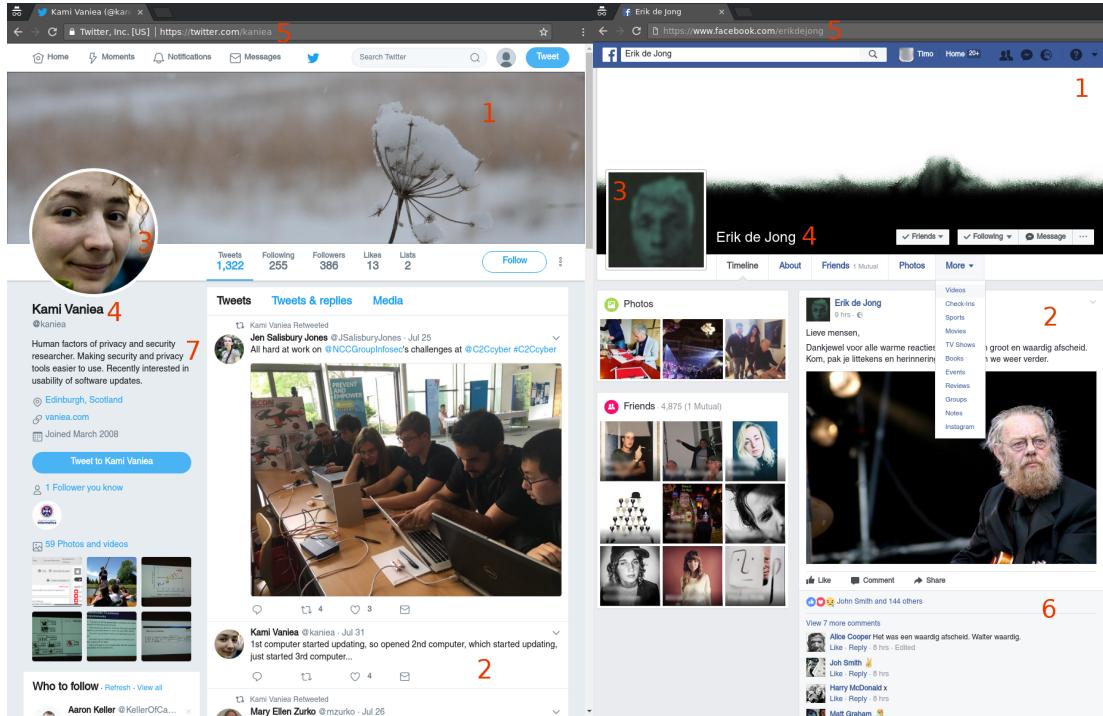


Figure 2.1: Examples of a Twitter (left) and a Facebook (right) profile page. 1. Cover photo, 2. User's Content, 3. Profile photo, 4. Full name, 5. Username, 6. (Facebook) Reactions to content, 7. (Twitter) Profile description.

<https://facebook.com/username>, and <http://twitter.com/username> in the browser, respectively. Examples of profile pages are shown in Figure 2.1. On Twitter, users are also *tagged* or *mentioned* by their username. Tagging or mentioning is the act of annotating content with someone else's profile. For example, on Facebook one could identify someone in a photo by tagging them. On Twitter, someone can be mentioned in a textual *Tweet*. For instance, “Attending an awesome lecture by @kaniea!”, where @kaniea is a hyperlink to the lecturer’s profile. A Tweet is a name for a post on Twitter, containing at most 140 characters. Due to Twitter’s character limit, and according to them to protect users, URLs on Twitter are automatically abbreviated to t.co links¹. For example, <https://www.linkedin.com/in/tmamulder> would become <https://t.co/0ACWVhJ1tY>.

Once the users have set-up their account on either of the platforms, they can adjust their privacy settings. By default, both platforms make the content that users create publicly available. On Twitter this is also available to people who are not logged in to the platform. In contrast, Facebook requires people to log

¹<http://t.co>

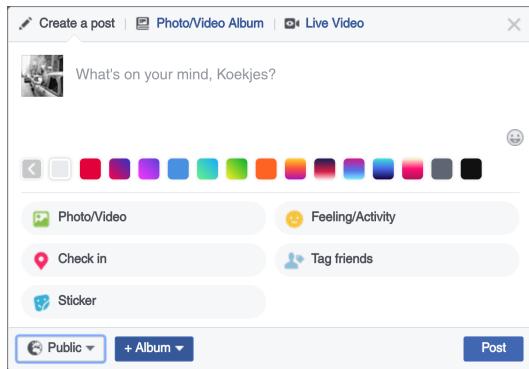


Figure 2.2: The screen a user sees when publishing content to Facebook.

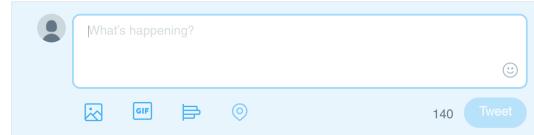


Figure 2.3: The screen a user sees when publishing content to Twitter.

in to view any user content. Moreover, as Figure 2.2 shows, Facebook explicitly states that content is made public by default when a user creates content, and the audience can easily be modified. In contrast, as shown in Figure 2.3, Twitter does not provide such easily accessible options. It is however possible to make Tweets private. However, this option needs to be set for all Tweets from the privacy settings menu.

Facebook's user IDs are used for both *user accounts* and *Facebook Pages*. User accounts are accounts that are supposed to be used by actual users. These accounts are meant to be kept more private than Facebook Pages. The latter are instead used for public messages such as those from organisations, artists, or to spread memes². Rather than becoming friends with a Facebook Page, users can *Like* a page, to show their interest. Liking a page is a similar connection to those users make on Twitter using the *follow* option. Most important is that Facebook distinguishes user accounts from Facebook Pages in that Pages are meant to be public, and to reach a larger audience. Therefore, using Facebook's Graph Application Protocol Interface (API), more information can be retrieved programmatically from Facebook Pages, than actual users.

As mentioned in Chapter 1, Facebook has significantly reduced their API functionality when they deprecated their API V1.0 and moved to V2.0 (Facebook inc., 2014). This should be taken into account when methods from related work are considered, since many of these methods cannot be used anymore. A brief overview of the most important changes and their implications from Facebook inc. (2014):

²Edited or composed images that are supposed to be funny.

1. Friend list is no longer available by default. This makes *Network Search* as described in Section 2.3 impossible through the API.
2. `username` cannot be requested anymore. User profile attributes can also not be requested through their username anymore. Therefore, a converter needs to be build to convert usernames to user IDs. Implications include that username similarity has been proofed to be a good method for matching users from a candidate set. Not being able to retrieve usernames makes it impossible to compare usernames. Also, the dataset retrieved from (Jain et al., 2013) contains usernames, which is used as a ground truth set.
3. Public Post search is no longer available. Therefore, the API cannot be searched for content anymore.

Due to the deprecation of many Graph API functions, most of the methods implemented for this project, as will described in Chapter 4, are novel.

2.3 Identity Resolution

In Jain et al. (2013) an overview has been given of several identity matching methods. In addition to the commonly used profile attributes, they introduced novel identity search methods. Jain et al. (2013) describe the following relevant search methods, and methods for identity matching.

Profile search exploits the fact that users might not take the effort to create different identities on different networks. Therefore, publicly available profile attributes can be used to find a user's identity on other networks. Note however that Facebook no longer allows this search behaviour through their API.

Content search helps to build certainty towards whether a user makes use of a particular online social media platform. Some users make use of functions that automatically publish posts to different platforms. The meta-data of a Tweet includes where such tweets come from, and can therefore be used to refine the search space. After deducing where the tweet comes from, the network can be searched for the content, and finally, candidates with zero cosine similarity between two texts can be excluded. Nevertheless, after Facebook reduced their API functionality, content can only be searched using the browser search function. However,

I will introduce a method in Section 4.3.3 which makes these types of queries possible again, with the use of browser automation. Unfortunately, searching for content creates a large amount of queries per user, and searching using browser automation is time consuming. Therefore, extensive research towards content search were excluded from the scope of this project.

Self-mention search is related to content search as it takes into account that users might refer to content of theirs on other platforms, also referred to as cross-pollination in Jain et al. (2011). A Twitter user could for example refer to a photo of theirs on Facebook. A set of candidates can be created from extracting the account information from the content owner of a link. This has been used in the current study by searching for URLs containing Facebook usernames in Tweets and Twitter user profile descriptions.

Syntactic matching is comparing all the candidates and their attributes from the search methods to the given identity and its respective attributes. Jain et al. (2013) computed a transposition based name and username comparison using the Jaro distance (Jaro, 1978). They report high mean average precision (MAP) for their findings on using syntactic matching on names and usernames from a candidate set. I have instead used the Levenshtein distance to compare usernames, which is the standard in string comparison (Navarro, 2001), and takes into account the structure of words.

Image matching is performed to match users with identical profile images on different networks. To find the similarity between two profile images they measure the difference in RGB-histograms. The current research will explore more refined approaches, namely perceptual hashing and face recognition. In the paper by Jain et al. (2013) RGB-histograms yielded high MAP as well. Their MAP is a measurement which incorporates the rank of the user in the candidate set. Therefore, it will be hard to compare the results of the current study to theirs. It is assumed, however, that using perceptual hashing and face recognition will work better, as they take more features into account than the colour histogram.

Other techniques for searching identities include the use of tagging patterns of users on Flickr, Delicious and StumbleUpon (Iofciu et al., 2011), and using

entropy measure on usernames (Perito et al., 2011), which are out of the scope of this project.

2.4 Perceptual Hashing

As mentioned above, to compare images Jain et al. (2013) used RGB-histogram comparison. This method is basic, as it merely compares images based on the amounts of colour similarity in an image. Therefore, it cannot distinguish between different images that have similar colour histograms (Pass and Zabih, 1999). Moreover, it does not take into account alterations made to an image, such as compression, cropping, colour, or contrast adjustments.

A more robust method for comparing images is perceptual hashing. A commonly used implementation is pHash (Zauner, 2010, p. 28-29), which contains a set of different methods that take the geometrical features of an image into account. Therefore, it is more robust against the above named issues RGB-histogram comparison suffers.

There are two main image hashing methods implemented by pHash: *discrete cosine transform (DCT)* and *radial hash projection (RHP)*. According to Klinger and Starkweather (n.d.), DCT is the most accurate. Nevertheless, from Zauner (2010), it did not become clear how much more robust DCT overall is against image transformations compared to RHP. On the other hand, DCT appears to be considerably slower compared to RHP (Zauner, 2010, p. 61). Zauner (2010) reports that on average, calculating a hash using DCT takes 9.7 seconds per image, whereas RHP merely needs 1.3 seconds. Thus, RHP is 7.5 times faster than the DCT based approach, making it more suitable for real-time web-based user requests. For the current project, the RHP method has been used, since it is faster, and more robust against JPEG compression (Zauner, 2010). A discussion of pHash's implementations of DCT and RHP will be given below.

2.4.1 Discrete Cosine Transform

A perceptual hash can be generated using Discrete Cosine Transform (DCT). This transformation can be calculated using a fast Fourier transform, from which only the cosine coefficients are leveraged (Ahmed et al., 1974). The coefficients generated using DCT contain the most distinct features of an image. The popular

JPEG image standard uses DCT as well to compress images (Wallace, 1992). In pHash, DCT is used to calculate a 64 bit hash of an image, which can then be compared using the Hamming distance (Zauner, 2010).

2.4.2 Radial Hash Projection

Radial variance based hash projections as implemented in pHash make use of the Radon transform (Zauner, 2010). The Radon transform, as defined in Radon (1986), projects a line L in \mathbb{R}^2 by calculating the integral from the line's angle α and distance from the origin s , see Equation 2.7.

$$\mathcal{R}f(\alpha, s) = \int_{-\infty}^{\infty} f((z \sin \alpha + s \cos \alpha), (-z \cos \alpha + s \sin \alpha)) dz \quad (2.7)$$

Standaert et al. (2005) extended the transform such that it can be used on discrete images. Let $L = x \cos(\alpha) + y \sin(\alpha)$, whose integral can be approximated with the summation of a pixel-wide strip:

$$L - \frac{1}{2} \leq x \cos(\alpha) + y \sin(\alpha) \leq L + \frac{1}{2} \quad (2.8)$$

To capture changes in luminance better, Standaert et al. (2005) incorporated variance into Equation 2.8, yielding:

$$\frac{1}{2} \leq (x - x') \cos(\alpha) + (y - y') \sin(\alpha) \leq \frac{1}{2} \quad (2.9)$$

iff $(x, y) \in \Gamma(\alpha)$ where x and y are pixel coordinates on the projection line $\Gamma(\alpha)$ with a given angle α , and x' and y' are the coordinates of the centre of the image (Standaert et al., 2005). Using the luminance of the pixel $I(x, y)$, Standaert et al. (2005) then define the radial variance vector as:

$$R[\alpha] = \frac{\sum_{(x,y) \in \Gamma(\alpha)} I^2(x, y)}{\#\Gamma(\alpha)} - \left(\frac{\sum_{(x,y) \in \Gamma(\alpha)} I(x, y)}{\#\Gamma(\alpha)} \right)^2 \quad (2.10)$$

for $0 \leq \alpha < 180$, not 360, due to the symmetry of Radon (Lefebvre et al., 2002).

2.5 Face Recognition

For the current project the face recognition implementation *Face Recogniton* has been used to calculate the distance between two faces. Geitgey (2016) wrote a

description on how the underlying algorithms work. In this section, a brief three part summary will be given.

First, face detection needs to be used to detect faces in an image. Face Recognition uses this to tell whether an image contains a face or not, and if so, how many. To do face detection, Face Recognition uses a *Histogram of Oriented Gradients* or HOG for short. To calculate the HOG, each pixel and its neighbourhood in a grey scale version of the image are analysed to create a map of gradient orientations, i.e. as seen from an individual pixel, which direction of neighbouring pixels are how much darker. After this, the average gradients of 16×16 pixel quadrants are calculated to reduce the amount of detail. This yields a more general pattern of a face, as shown in Figure 2.4. A model for detecting faces can then be built by training using a machine learning algorithm on many manually classified HOG patterns of faces and non-faces.

Second, to deal with different face orientations, the detected faces need to be projected using face landmark estimation. For this Face Recognition uses the method developed by Kazemi and Sullivan (2014), which tries to find 68 specific points per face, including chin, eye edges, nose, and mouth. Once the landmarks have been found, an affine transformation can be performed on the image to straighten it.

The third step is to encode faces based on their own distinct features, such that they can be compared to one another. A Deep Convolutional Neural Network can be trained to find an embedding of 128 different measurements per face. Instead of training the network themselves, Geitgey (2016) took a model from OpenFace³. The trained network can be fed with a new picture of a face, which generates their face encoding.

After the faces have been encoded, two faces can be compared to each other by calculating their euclidean distance using the following equation:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.11)$$

where d is the euclidean distance between encoded faces \mathbf{p} and \mathbf{q} .

³<https://cmusatyalab.github.io/openface/>

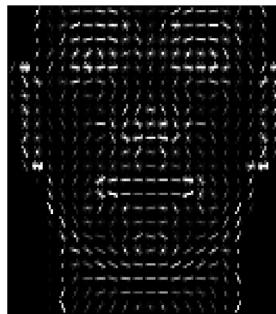


Figure 2.4: The Histogram of Gradients of a face (Rojas Q. et al., 2011, p. 10).

2.6 Informing Users

Correa et al. (2012) also searched for users by their automated sharing patterns of their activity feeds across different networks. Although they do not mention exploiting the Twitter API for finding the origin of the post, they do utilise the self-mention behaviour of users. In their paper, a table is included which shows the publicly available profile attributes of a user for different social media sites, including Flickr, Foursquare, YouTube, LastFM, Twitter, and Facebook. It shows what personally identifiable information (PII) can be extracted from each of the six different social media platforms, and the reader can infer how much more information can be extracted when two or more accounts are linked together. This table should already raise concerns for users of social media, as linking an account can suddenly turn four public PII attributes from Facebook into 20 attributes from YouTube. Correa et al. (2012) also produced a web application where users could log in using their Twitter credentials and see which of their corresponding accounts were found on other networks. Nevertheless, their research did not include a design requirements study, nor the effect of showing such information to the user. For the current project, a user study has been conducted in the form of a focus group, in order to find the design requirements for giving appropriate, clear, and comprehensible feedback to users.

Measures to prevent leaking of personally identifiable information on the side of all parties that are involved (e.g., user, platform, or advertisement company) have been discussed by Krishnamurthy and Wills (2009). Unfortunately, the proposed measures that users can take are either radical, i.e., ‘stop sharing anything’, or somewhat technical to handle for most users. For example, even for advanced users blocking third-party cookies is not an easy task.

Other studies, such as Roesner et al. (2012) describe possible ways for aiding

users with problems related to unwanted tracking. Eckersley (2010) took this a step further by creating a website⁴ which, after a test, provides concise information to users about how they are being tracked. They also provide a one click solution *privacy badger*, which was found to be “a perfect example of the balance between privacy and usability” (Kaplas, 2016, p. 32) in a usability evaluation. Another example of such a website is browserleaks⁵, which shows several tools for testing different parts of one’s web browser that could leak information, such as JavaScript, Geolocation, or canvas fingerprinting. Although none of the measures discussed in the literature help users protect themselves from identity resolution as described above, the latter two websites, and the interface by Correa et al. (2012), can be used as examples for some initial designs of the interface.

⁴<https://panopticlick.eff.org/>

⁵<https://browserleaks.com/>

Chapter 3

Design Requirements for the Feedback System

3.1 Introduction

One of the main goals of the project is to inform users about how their accounts can be linked. Therefore, the design requirements for a feedback system need to be found to give appropriate and helpful feedback to users. In order to find the design requirements, two focus groups have been conducted. A focus group is a qualitative research method to gather opinions and attitudes on a product or service from a number of participants (Hanington and Martin, 2012, p. 92). Sharing ideas on how they or their acquaintances use and think about products can be used to find and improve essential parts of the system that is to be built.

There are three main problems to address with help from the focus groups. Firstly, it is important to know what kind of information would be useful and novel to present to users of the system. Information returned by the system should not be too obvious, such that users might feel like the system is uninformative. Returning more complex information, however, yields a second problem to address: how can more complex information be presented to the average computer user in a comprehensible way? The focus groups can help by discussing how they would explain the way that two accounts are linked, and how they would propose possible resolutions to an average computer user. Lastly, the style and level of detail of the results need to be determined. The questions that came forward from these problems can be found in Section 3.2.1.

3.2 Methods

Both focus groups took around 20 minutes, and included seven and six participants, respectively. The Moderator's and Participants' voices were recorded with a Zoom H2n microphone to ensure surround recording and professional audio quality.

At the start, participants were asked to fill out a survey and a consent form; the originals are included in Appendix A. A short introduction followed, in which the concept of a focus group and the project itself were briefly explained. Also, brief explanations were given on the interfaces of Twitter and Facebook, in case participants had not seen them before. In both focus groups, the same four questions were asked.

In the survey, participants were presented with questions about their demographics and their social media use. These questions were designed to find out about participants' backgrounds, level of privacy concerns, and what kind of information they share on social media platforms. Questions 2.9 and 2.10 of the survey in Appendix A are not relevant to the current study. They were included for another project, for which a focus group was conducted simultaneously with those participants not participating in the current study's focus group.

3.2.1 Focus Group Questions

Question 1. What measures would you take to ensure the employer would not find out it is your account?

The first question of the session aimed to find out what precautions the participants would take if they felt a strong need to stay anonymous. The question was introduced with the following scenario: “*Imagine you have a strong political view, and you're expressing this on your public Twitter profile. Your employer however, forbids you from doing this and if they find out that it's you posting, you'd be immediately fired.*”. The participants were then challenged with Question 1.

Throughout the focus group, the scenario described above was used to remind participants of what threat model they should keep in mind when thinking about other questions. A threat model is the person or organisation users try to protect themselves from.

Question 2. In what ways do you think social media profiles on different platforms can be linked to each other?

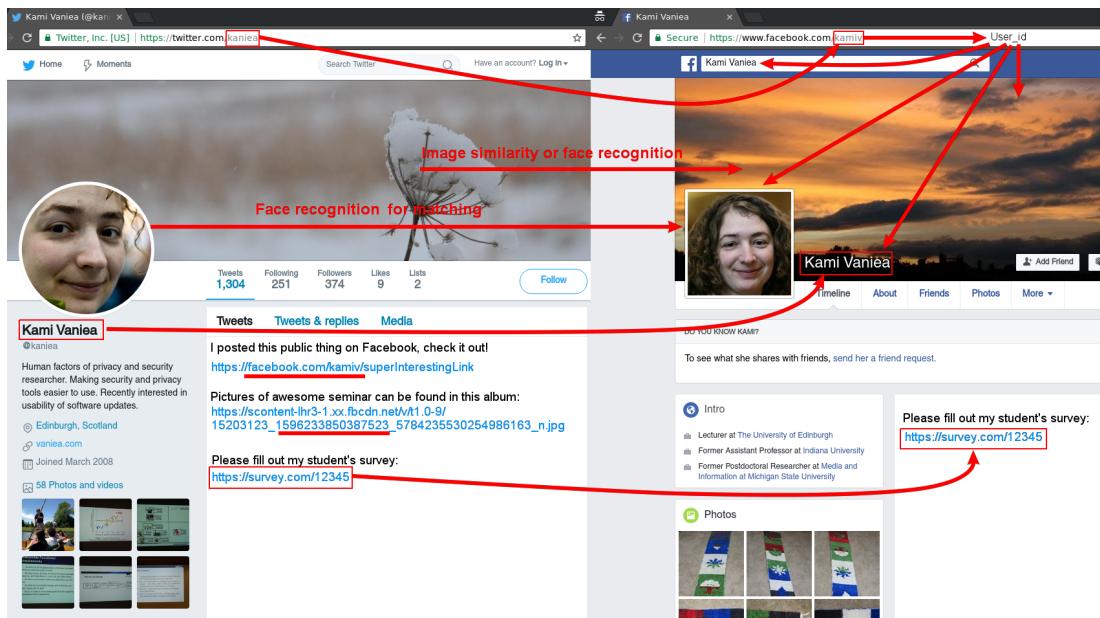


Figure 3.1: An example of some possible features in a Facebook and Twitter profile that could help to find or match two accounts.

To add detail to the first question, the second question was designed to provoke a discussion about their ideas on how social media profiles from different platforms can be linked to each other. A discussion on this topic should both reveal ways of linking accounts not considered before by the author, as well as outline profile attributes that could obviously be used to link two accounts.

Question 3.A. How would you explain the way that we've linked these two profiles to an average computer user? (e.g. your parents)

Question 3.B. How would you explain to your parents how they can prevent this from happening in the future?

During the third question, Figure 3.1 was shown, to give a visual representation of which profile attributes and content could assist in searching and matching profiles. The final feedback system should provide users with useful information on how social media accounts can be linked cross-platform. Therefore, the participants were asked Question 3.A, concerning how they would explain to an average computer user how the two accounts had been linked. To make Question 3.A more explicit, the participants were then asked Question 3.B. The question asks how they would explain to their parents how they should proceed to prevent their accounts from being linked in the future.

Username Similarity	70%
Profile photo similarity	60%
Other photos similarity	0%
First and Last name identical	Yes
Link from one account to another	No
Crossposted unique link	Yes
Facebook photo reference	No
certainty that profiles match: 60% +	

Figure 3.2: A very basic mock-up of a score calculation that could be presented to a user of the system after requesting information about profiles that are associated with their account.

Question 4. Does the score, or do parts of the score help you understand what you should do next?

The final slide contained an example of how the calculation of a score could be presented to a user of the system. Hanington and Martin (2012, p. 170) state that artistic details should provide enough context, but should however not distract participants from the purpose. Therefore, to ensure that the ideas coming from the participants focus on the content, rather than the design, the lay-out was kept as simple as possible, as shown in Figure 3.2. They were asked Question 4 on whether the score or parts of the score would help them understand how they should proceed to break the link between their accounts.

3.2.2 Analysing the Focus Group

To analyse the outcomes of the focus groups, initially a rough transcript was made from the recordings. The transcript was then coded using *initial coding*, in combination with *In Vivo coding*. Coding transcripts serves the purpose of summarising the main points, and making the transcripts easier to search for the reader (Saldaña, 2009, p. 3). *Initial coding*, or *open coding*, is a very basic method for dividing participants' opinions into discrete parts, such that they

can be compared and analysed (Saldaña, 2009, p. 81). With *In Vivo coding*, the words of the speaker are coded in a literal way, such that the terms used by the participants are preserved (Saldaña, 2009, p. 74). The latter coding method is more strict than the former, and the methods were used interchangeably based on the importance of preserving a participant's exact words.

In order to protect the privacy of the individuals that took part in the focus group, neither the literal transcripts nor the coded transcripts are included in this thesis. Nonetheless, parts of it, and the resulting outcome, can be found in Section 3.4.

3.3 Participants

As mentioned, at the start of the focus group participants were asked to fill out a survey. All except one of the participants in the focus groups were either students or researchers in the field of computer science or related subjects. The majority of the group comes together on a weekly basis to participate in the research of another student in the group. As Figure 3.3c shows, four participants have a bachelor's degree, five a master's, two a PhD and two are Postdoctoral researchers. The group was separated in two, balanced based on degrees as much as possible, see Table 3.1. Because one of the two postdoctoral researchers (P6) did not come from a computer science related background, these researchers could be placed in the same group.

As Table 3.1 shows, all participants have above average computer and privacy self-efficacy. The score, out of a 5-point Likert scale, was calculated by asking them how often they ask other people for help regarding these subjects, and how often other people ask them for help with such problems, respectively. The response number (1 to 5) to the former question was then inverted, added to the response of the latter question, and subsequently divided by 2, as shown in Equation 3.1.

$$\text{self-efficacy} = \frac{Q_2 + (6 - Q_1)}{2} \quad (3.1)$$

where Q_1 represents how often participants ask for help, and Q_2 how often others ask them for help regarding a subject, both on a scale from 1 to 5.

Figure 3.3a shows that there was an almost equal distribution over binary genders. However, there is a clear mode in the age distribution from 20 to 25

Table 3.1: Demographics of the focus group participants. The horizontal line between P7 and P8 depicts the split between the two groups. Computer and privacy self-efficacy are based on a 5-point Likert scale, based on the answers to questions in 1.7 of the survey in Appendix A.

Id	Gender	Age	Nationality	English (Years)	Finished Degree	Field	Computer Self-efficacy	Privacy Self-efficacy	Privacy Index
P1	M	21	Bulgarian	12	UG	CSc	3	3	F
P2	M	23	Chinese	5	UG	CSc	4	4	U
P3	F	27	Saudi Arabia	8	Master's	CS	3.5	3.5	P
P4	M	29	Mexican	2	Master's	SE	3.5	3.5	P
P5	F	35	British	N/A	PhD	AI	4	4	P
P6	F	28	British	N/A	Postdoc	HCI	3	4	P
P7	F	54	Romanian	7	Postdoc	HCI	4	4	P
P8	M	20	Czech	20	UG	CSc	4	3	F
P9	F	25	German	15	UG	SE	3.5	3	P
P10	F	33	Saudi Arabia	10	Master's	USP	3	3	P
P11	M	23	Scottish	N/A	Master's	Inf	4	4.5	P
P12	M	24	Chinese	14	Master's	CSc	4	3.5	U
P13	F	34	American	N/A	PhD	-	3.5	3.5	F

UG: Undergraduate.

AI: Artificial Intelligence; HCI: Human-Computer Interaction; CS: Computer Security; CSc: Computer Science; Inf: Informatics SE: Software Engineering; USP: Usable Security and Privacy.

F: Fundamentalist; P: Pragmatist; U: Unconcerned.

years old, as can be seen in Figure 3.3b. Having such a young group is the result of the group being primarily advertised to master's and undergraduate students.

In the survey, participants were also asked about their social media use. The social media platforms in Figure 3.4 were selected based on a combination of the *Alexa Top 500 sites on the web*¹ and platforms that were investigated in related work (such as Facebook, Twitter, and FourSquare). Piazza was added as this is the main platform used by the University to ask course related questions. Currently popular platforms that could have been included are Instagram² and Snapchat³, as 4 and 3 users, respectively, wrote them down under other social media platforms they frequently use.

Figure 3.4 shows that all users used at least one social media platform. Most frequently used platforms seem to be Facebook and YouTube. We did however not ask the participants how often they publish content on these platforms. YouTube

¹<http://www.alexa.com/topsites>

²<https://www.instagram.com/>

³<https://www.snapchat.com/>

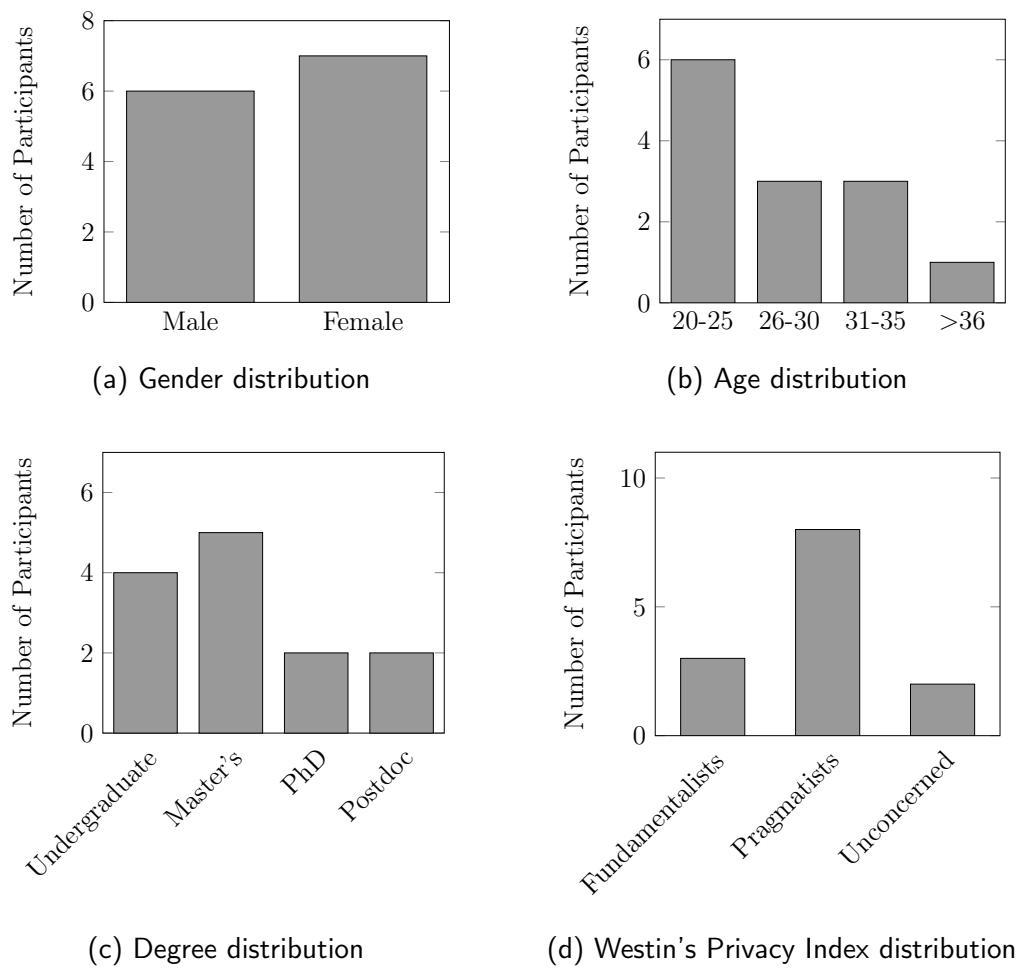


Figure 3.3: Demographics of focus groups

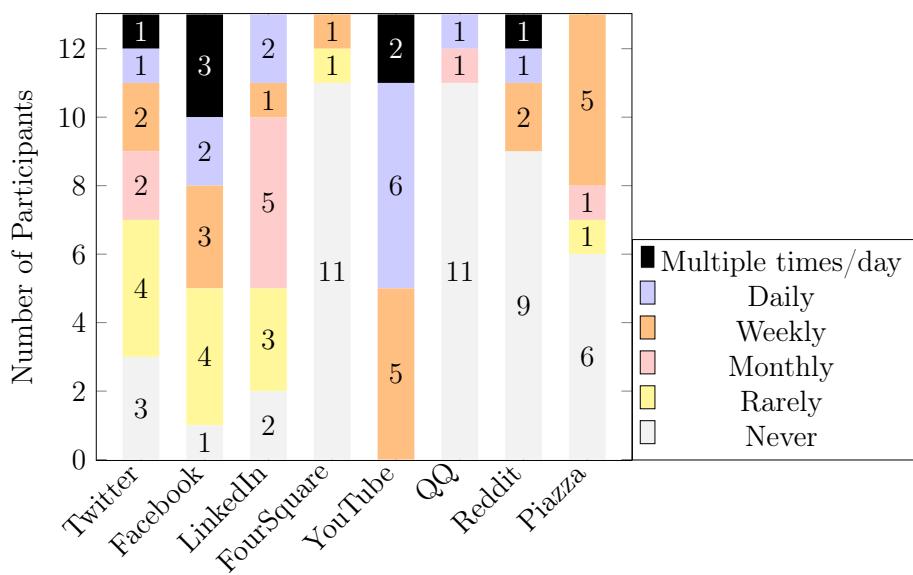


Figure 3.4: Social Media use of focus groups.

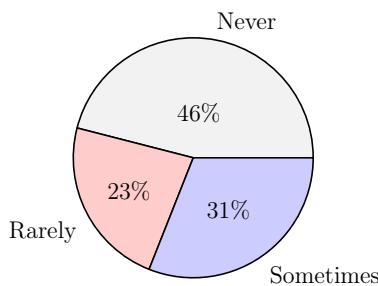


Figure 3.5: Frequency of participants posting the same content to multiple social media platforms.

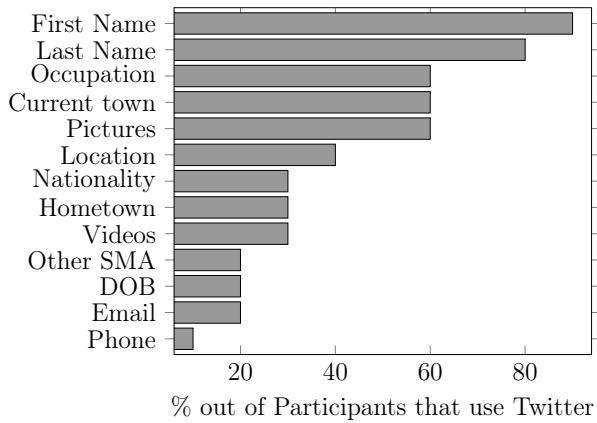


Figure 3.6: What information participants of the focus groups publicly share on Twitter. (SMA: Social Media Accounts, DOB: Date of birth).

is a platform where videos can be watched without the need to sign in. Therefore, the likeliness of the YouTube users being *lurkers* (users who observe, but do not contribute (Dennen, 2008)) is higher than for other social media platforms, where they are required to log in. Twitter and Reddit also do not require users to log in to view content. However, Figure 3.6 shows that most of the participants who use Twitter share at least their first name on the platform.

LinkedIn and Twitter also seem to be frequently used by most of the participants, as Figure 3.4 shows. All but three of the participants use Twitter. This is a particularly interesting platform since, as was mentioned in Chapter 1 and Section 2.2, all data on Twitter is public by default. Therefore we also asked the Twitter users among the group what information they included on their profile. The results of this question are shown in Figure 3.6. From the 10 participants using Twitter, 90% includes their first name. Eight out of ten Twitter users include their last name. With the purpose of linking accounts in mind, noting that even a relatively privacy concerned group includes this information is promising. Moreover, the fact that some participants include links to their other social media accounts, their date of birth, email address, and one person even their phone number, show that these attributes could be of use when searching for Twitter users' Facebook accounts. Furthermore, Figure 3.5 shows that 54% of the users have posted the same content on different platforms. When the content is only posted by few users, their account on another platform can be revealed when

searching that platform for their public content on the first platform. For example, ‘Alice’ can publicly post a URL linking to her survey on both Twitter and Facebook. The URL could be scraped from her tweets, and searched for on Facebook, likely returning merely one result: Alice’s profile. This will be elaborated upon in Section 4.3.3.1.

3.3.1 Westin’s Privacy Index

Since the current project deals with privacy related issues, information about the participants’ privacy concern could be useful when analysing the results of the focus group. To calculate the participants’ privacy index, a modified version of Westin’s “Consumer Privacy Concern Index” by Harris and Westin (1991) (as cited in Kumaraguru and Cranor, 2005) has been used. Our survey included three statements related to corporate use of consumers’ data. The original questions were written by Harris and Westin (1991) (as cited in Kumaraguru and Cranor, 2005), and were modified by Vaniea (2016). From their responses to the statements, participants can be classified into three privacy concern categories: unconcerned, pragmatists, and fundamentalists (Kumaraguru and Cranor, 2005). The following three statements regarding consumer privacy were included in the survey:

Statement 1. Consumers have lost all control over how personal information is collected and used.

Statement 2. Most businesses handle the personal information they collect about consumers in a proper and confidential way.

Statement 3. Existing laws and organisational practices provide a reasonable level of protection for consumer privacy today.

Agreeing or strongly agreeing with Statement 1 is considered privacy-oriented. For Statements 2 and 3, disagreement or strong disagreement is considered to be more privacy-oriented.

To classify participants, a modified version of the classification by Kumaraguru and Cranor (2005) has been used. Participants are considered privacy unconcerned if they did not pick the privacy-oriented option in any of the statements. Privacy fundamentalists are considered those who selected the privacy oriented option in all statements. All other combinations were classified as pragmatists. See Figure 3.3d and Table 3.1 for the results.

3.4 Results

3.4.1 Staying Anonymous

Profile Details In response to Question 1, about staying anonymous, most participants agreed on creating a new account, without any of one's actual names, links, or information about their identity. Furthermore, some think one should not write any details about oneself, no description, no location, and no pictures. Moreover, P9 pointed out that one should not have too many friends from their actual location. Additionally, she suggested that generating confusion might work better than leaving out details. For example, one could write down a different city for the place they live in on their profile description, and make sure they have friends or followers from different places. Some participants mentioned that one should also be careful with their writing style.

Email Address Something that was mentioned in both focus groups, brought up by P6, and P13, respectively, is to use a different email address to sign up with. P13 even suggested to make use of a public mailbox such as Mailinator or Trashmail to increase plausible deniability.

Usernames Everyone agreed that it is important to pick a random username, something that is not linked to your real identity in any kind of way. On top of that, P5 suggested that on Twitter one can change their username, so one might want to do that every once in a while. Nevertheless, the participants were unsure whether it is possible to see the history of usernames, which would make this measure redundant.

3.4.2 Linking Accounts

Question 2 asked the participants how a link between two accounts from different social media platforms could be established.

Name Participants note that an obvious link from one profile to another is using the same name. On the other hand, it was mentioned that this might be dependent on how common someone's name is. If one has a statistically infrequent name it might be particularly easy to use Google to find someone's other account(s).

Email Address Email addresses as a privacy concern were brought up earlier. However, during the second question, it was mentioned more explicitly that these could be linked to each other. Moreover, the root domain of an email address (i.e. the part on the right side of the @) might be indicative.

Pictures The possibility of matching pictures was also mentioned again in both groups. P13 added that not just the same person, but also the same item or animal (e.g. the same cat) could provoke a match. Pictures can sometimes also lead to establishing links outside of a user's control. This is particularly the case when someone is tagged in another person's pictures, or tagged in someone's tweet.

Location Both groups also agreed that sharing a location on both profiles might reveal one's identity. This specifically applies when a user would 'check-in' at the same places with the two accounts. P11 mentioned that sometimes logging-in using a social media account or even checking in using that account is required to use a cafe's wireless internet.

Automated cross-posting The integration of different social media accounts on other platforms was further mentioned as an easy way to link accounts by P9 and P11. An example of this is automatically posting content from Twitter or Instagram to Facebook.

3.4.3 Explaining Results

Removing or changing information After showing Figure 3.1 and asking Question 3.A, the participants generally thought that the image was quite self explanatory. P1 bluntly said "break the link", which showed understanding of which links need to be broken. If there is a link between two accounts, one should break it by, for example, removing or changing information. Some of the participants also mentioned that if one does not want to be identified, they should not include any of the same information or content on both platforms. Nonetheless, everyone agreed that it is easy to accidentally slip some information through.

Pictures The participants stressed that information should be included about using different pictures for different accounts. It might not occur to average users that technology exists which is able to recognise that the same person appears in

two different pictures. Furthermore, they might not know that face recognition could still be performed on a different picture of the same person, even with an age difference. Important here is that they might not see themselves as the same person due to there being an age gap between the two photos, P13 said. Therefore, some suggested using a placeholder instead of a photo of yourself.

Links P13 noted that the hyperlinks in Figure 3.1 are presented in a clear and obvious way, whereas a shortened or embedded link, or a link with different text around it, might not be as obvious.

General information P6 also suggested that there are more general concerns they might not be aware about, such as that public content is visible to anyone, even if they are not a connection on the platform.

3.4.4 Presentation of Feedback

Score comparison In reaction to Question 4, the focus group revealed that the score table from Figure 3.2 needs many adjustments. Most importantly, the score itself is confusing. The percentage did not make sense to the participants. Rather, they would like to see how their profile match compares to other matches made.

Reduce amount of information Participants mentioned that Figure 3.2 includes too much information to comprehend at once. Moreover, some of the terms, such as ‘crossposted’ might puzzle average users. Therefore P11 suggested to only give the most identical features, for example the one on which the system had matched them. The second focus group also agreed on the importance of using examples.

3.5 Design of Feedback System

The focus group yielded several suggestions on what information would be trivial to average users, and what they would not have thought of. Moreover, it was outlined what methods would work best to make complex information comprehensible. Additionally, ideas were gathered for the visual presentation style of the feedback.

3.5.1 Potential Information for a Feedback System

From the focus groups it can be concluded that when someone tries to stay anonymous, one should not use their real name. Other rather obvious defensive tactics included picking a random username, and leaving out any personal details. Although such characteristics were marked as obvious, Section 3.4.3 revealed that leaving out personal details can be difficult.

Since using a personal, or indeed, the same email address has been mentioned as a possible risk in both focus groups, this might be a clear mistake as well. However, it was not explicitly repeated by other participants. Moreover, it is dependent on the threat model. That is, if a threat agent would have access to your email address from one of the social media accounts in the first place, then this would comprise a vulnerability. It is possible for all Facebook and Twitter users to search the databases by an email address. Therefore, this is a reasonable concern in case the threat agent is an employer, who likely has access to one's email address. Users who do not want to be found by email address or phone number have to explicitly turn this off, as the default setting on both Facebook and Twitter is to allow everyone to find a user given such data. Nevertheless, if one does not have access to someone's email address, which is not publicly shared by default on either platform, then the threat ceases.

A less evident way to determine the actual identity from a social media profile seems to be to have many friends or followers from one's actual location. The latter could play an important role in narrowing down the search for a user's profile on another social media platform. It is however not trivial to build a completely novel network. For example, adding random people as friends on Facebook can attract scrutiny, especially when little personal information is available on the profile.

Some of the participants seemed to know that two profile pictures can be matched using face recognition. However, not all participants explicitly showed to be aware of this. Moreover, it was mentioned that average users are unlikely to know about such techniques. Therefore, it is important to make this explicit to users of our feedback system.

3.5.2 Explaining Results to the User

Initial reactions to Question 3.A showed that the way that links were created in the example of Figure 3.1 is quite self-explanatory. Therefore, either the image is very clear, and should hence be kept in the final system, or the included examples are too simple. To elaborate, the URLs in Figure 3.1 were unmodified ones (e.g. not shortened by Twitter), and exactly the same on both profiles. On the contrary, URLs included in tweets are automatically abbreviated (Twitter inc., 2017b), concealing their similarity with the original URL.

From Section 3.4.3 it appeared that users of the system should be advised in simple language on how pictures can be matched. A possible solution would be to include a separate tool where users can upload a picture, to see how it compares to their profile picture.

Finally, including general information on commonly made mistakes should be considered as well. According to the participants, help on who can see what content, what information to exclude from your profile to stay anonymous, and information on comparable images could support users in finding their mistakes. Moreover, it can help prevent users from accidentally leaking data in the first place.

3.5.3 Presentation of Feedback

As discussed above, visual representations of the links laid between people's accounts should be considered for the interface. Nevertheless, the information that is presented in general should be kept to a necessary minimum. Particularly Section 3.4.4 makes clear that Figure 3.2 shows too much, and too complicated information. A possible solution would be to use examples on how accounts were linked, and give users the possibility to request more information.

Section 3.4.4 shows that the score needs to be changed into something that users can relate to. The percentage is not meaningful in itself. Therefore, the possibility of comparing oneself to other profiles should be considered. A scale might help. For example, the user could find out whether their identity is as easily retrievable as that of a public figure, or, on the other side of the spectrum, someone who tries to stay anonymous.

3.6 Resulting Design Requirements

From the information that was gathered in Chapter 3, the following design requirements have been composed:

- Reduce the amount of information shown to users in two ways:
 1. Use personalised information that applies to vulnerabilities detected in their accounts.
 2. Not including obvious information, e.g. names are identical.
- Comparison to other profiles, for example, showing where the user is on a scale between anonymous and public figure.
- Use of (personalised) examples such as the means through which their accounts were actually linked.
- Visual examples, for instance, arrows drawn from their Twitter profile to their Facebook profile that caused the link.
- Information on (shortened) URLs, included in a FAQ or a dedicated page.
- Additional information on pictures. Faces, objects, animals can be matched to some extent, this should be included in the FAQ or perhaps an additional tool should be provided where two pictures can be uploaded.
- Additional information on who can see what. This could be in the form of a YouTube video explaining the privacy settings for content and profile visibility.

Figures 3.7 to 3.9 illustrate what the feedback system could look like when the above stated design requirements are considered. Clicking the ‘personalised information’ button in Figure 3.7, the welcome screen, brings the user to Figure 3.8. Figure 3.9 shows a screen that could be returned to the user when ‘general information’ is requested from Figure 3.7.

Are My Social Media Accounts Anonymous?

I'd like to get

Personalised Information
(Requires logging in using Twitter)

General Information

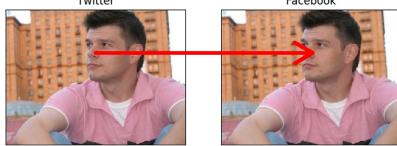
[about this project](#)

Figure 3.7: Welcome screen of the feedback system.

Feedback

We were able to identify your Facebook account

- Your profile pictures are very similar. Try not
 - to include the same face, animal, or objects on both profiles.



Anonym -O- meter



You

Anonymous

Public Figure

- You use the same username for both accounts.
- Information on how to change this on Twitter can be found [here](#), for Facebook click [here](#).

Figure 3.8: Personalised feedback page.

FAQ

How to include Facebook content on Twitter without revealing my Facebook account?



Who can see my posts? Watch this video to learn about, and optimise your privacy settings.



Figure 3.9: General information page with explanatory videos.

Chapter 4

Implementations for Identity Resolution

4.1 Introduction

As mentioned in Chapter 1, the second and third goals of this thesis are to investigate what the current states of Facebook and Twitter are regarding publicly retrievable information from social media users. Given this information an attempt will be made to find and match users' Facebook accounts, given their Twitter username. Performing identity resolution ourselves is crucial for giving personal feedback to users on how their accounts can be linked. Therefore, this chapter will discuss several implementations I developed which, when combined, can perform identity resolution.

With the release of Facebook's Application Protocol Interface (API) v2.0 in May 2014, a lot has changed in the amount of publicly retrievable information using their Graph API (Facebook inc., 2014). Most importantly, Facebook started denying requests which use usernames, and instead requires numerical user IDs (Facebook inc., 2014). Moreover, after the website <http://icanstalku.com> (Jackson and Pesce, 2012) was published, Twitter stopped posting meta-data of images (Twitter inc., 2017c). This all lead to reduced information availability from Twitter and Facebook's API, making several developed methods for linking two accounts obsolete (e.g. some of the methods by Jain et al. (2013) as discussed in Chapter 2). On the other hand, the search engine that is available through the browser interface on Facebook appears to be much more powerful than their API. This allows any text-based search through the complete public Facebook

database, without any additional permissions from the users. Therefore, to retrieve the largest amount of information, a method needs to be found to gain access programmatically to the browser search engine, for example through browser automation. Also, the public information that can be retrieved using the API needs to be utilised to its fullest extent. The pitfall of browser automation could be that Facebook has built-in mechanism for detecting such queries, as it is doubtful whether it complies with their *Automated Data Collection Terms* (Facebook inc., 2010). Nonetheless, below I will show a proof-of-concept to search for full names and URLs through browser automation.

Since I do not have the means to scrape Facebook for pictures, and, moreover, they disallow scraping their database (Facebook inc., 2010), the search for possible candidates will be mostly text based. On the other hand, due to Facebook's API disallowing the retrieval of public posts, but allowing the downloading of profile and cover photos, comparing accounts will be performed mostly using image based methods.

I have developed and evaluated three methods to create a candidate set, and three methods to confirm two accounts are from the same person. To find Facebook accounts, I have evaluated searching for Twitter usernames on Facebook. Moreover, I have developed and evaluated methods to extract *self-mention* of Facebook accounts in Tweets, and browser automation to search full names and URLs. These methods yield a candidate set of accounts, potentially belonging to the Twitter user. To find which account matches the Twitter user's account, I introduce Levenshtein distance for username comparison, and perceptual hashing and face recognition to compare profile photos. An overview of these methods and how they could be used in an actual identity resolution system can be seen in Figure 4.1

Next, in Section 4.2, several methods will be introduced to retrieve all possible information of interest from Facebook and Twitter. This information can be used to find accounts and build a candidate set. Methods that leverage this information are described in Section 4.3. Finally, in Section 4.4, tests for implementations for comparing accounts through usernames and profile photos will be discussed.

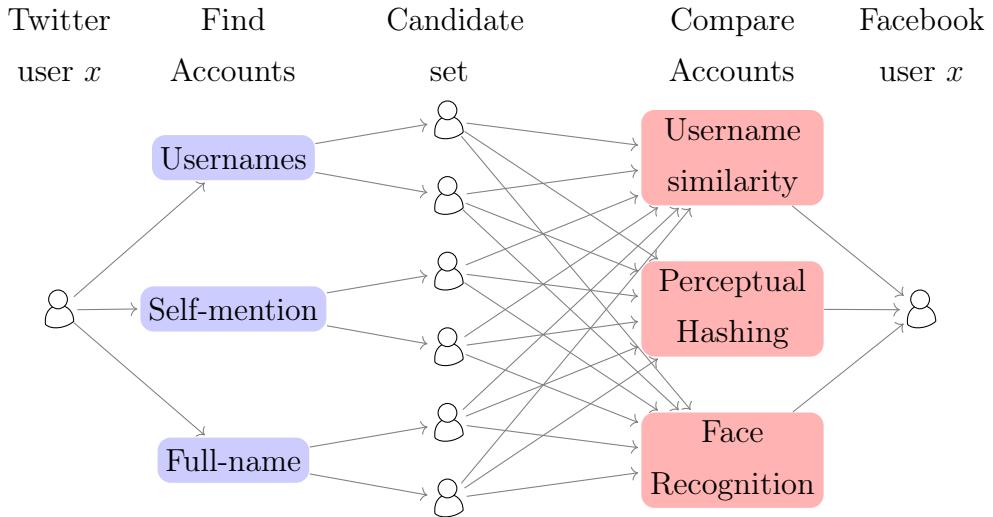


Figure 4.1: Overview of the methods I implemented and tested and how they could be connected to perform identity resolution on a Twitter user.

4.1.1 Machine Info

Experiments were conducted on a laptop running 64-bit Ubuntu 16.04 with an i7-4600U processor which has 4 threads on 2 cores @ 3.30 GHz, 12GB DDR3 RAM, 256GB Toshiba Multi-level cell SATA-III SSD, and an Intel 7260AC wireless network card connected to the University of Edinburgh's wireless Eduroam network. For the annotation task, a 24 inch HP E241i, and a 24 inch iiyama ProLite B2483HS monitor were used by the first and second annotator, respectively.

4.2 Retrieving Information

After email correspondence, Jain et al. (2013) provided me with a dataset which contains ground truth Twitter and Facebook username pairs. To assert the validity of the dataset, I firstly accessed the Facebook and Twitter APIs using Python bindings. Since this dataset did not include the user IDs, I was unable to request user accounts from the Facebook API. Nevertheless, it appeared that Facebook's API returns a different error message depending on whether a user exists or not, and it returns actual information for Facebook Pages. If a user exists, the API returns “(#803) Cannot query users by their username (*username*)”. If the user does not exist, the API returns “(#803) Some of the aliases you requested do not exist: *username1*”. Based on the word ‘cannot’ in the error message, a different error message, or no error message, the account pairs in the dataset were labelled

as existing user (1), non-existing (0), or existing Page (2), respectively. Existing Twitter users could simply be separated from non-existing users based on whether an error was returned or not by the Twitter API. Since Twitter has a rate limit of 180 queries per 15 minutes¹, the script to validate account pairs first asserted the existence of the Facebook account, before retrieving information from Twitter.

After running this script for several days on the dataset from Jain et al. (2013), 138,097 existing account pairs were obtained. Of these accounts, 116,520 Facebook accounts appeared to be user accounts, and 21,577 Facebook Pages. The following sections describe how information from these accounts can be obtained.

4.2.1 Converting Facebook Usernames to Facebook IDs

Since the deprecation of Facebook API v1.0 in April 2015, Facebook requires the use of numerical user IDs instead of usernames for all API calls on user accounts (Facebook inc., 2014). Indeed, they do not provide a tool to convert usernames to user IDs, or vice versa. There are two reasons why it would be useful to have the ability of looking up users by their username. Firstly, our dataset contains usernames. Secondly, as Section 4.3.1 will show, 31.42% of our dataset contained the same Twitter username for a linked Facebook account, which can thus help with finding accounts.

Online tools exist to look-up Facebook user IDs from usernames, such as lookup-id.com². However, no open-source methods were found to automate this.

To build our own method, first, without being logged-on to Facebook, a few users were requested in the browser using <https://www.facebook.com/username>. Both public and non-public profiles were retrieved. The source code of the resulting pages was examined, to see whether the numerical ID was returned. From lookup-id.com, the actual user IDs were known. Hence, it was possible to search for the actual ID in the source code. It appeared that, even though the visible part of the browser returns “page not available”, some meta-data about the profile is still being transmitted through the HTML source code. Particularly, the user IDs are sent along with an attribute called `entity_id`.

Using the Python package named `requests`, the source code of the above mentioned webpages could be retrieved. A first attempt was then carried out by running a regular expression search on the source code, and keeping the line

¹<https://dev.twitter.com/rest/public/rate-limiting>

²<http://lookup-id.com>

containing `entity_id`. However, it appeared that the structure of the source code changed for each request, and sometimes the ID itself was cluttered by irrelevant code. Therefore, surrounding lines needed to be kept, and a split of strings into a list was performed based on the following characters: `, \-!?:& =/[]}`", yielding a list of separate entities that appeared in the line. The user ID could then be retrieved by looking up the index of the word `entity_id` in the list and returning the next item in the list.

Sometimes retrieving a user ID using the above method would fail at the first few attempts. Therefore, the function would call itself again when it would fail, making it recursive. Nevertheless, this would sometimes lead to reaching a maximum recursion depth. Consequently, a counter was added to make the function only recurse 99 times at most, which gives a maximum waiting time of 27 seconds. The average query time was 0.5 seconds, and the mode 0.3 seconds ($N = 6,659$). If the algorithm still fails after 99 tries, it will return 18 consequent zeros, which is the same length as the longest username. This should make it easy to search for non-existent, or non-retrievable users in the returned data.

On the set of 138,097 Facebook usernames, the resulting function returned a numerical user ID in 99.98% of the cases. The remaining usernames could have been changed, or the account could have been deleted. When querying by hand the 29 profiles for which an ID could not be retrieved, it appeared that merely 2 of these usernames were actually known on Facebook. Therefore, it is assumed that most of these usernames have become deprecated between the two weeks they were last verified by the algorithm from the start of Section 4.2, and the time I requested their numerical ID. Apart from these two, one ‘username’ appeared to be a Facebook help page. When the accounts that did not seem to exist anymore are not considered, the recall can be updated to approximately 100.0% ($N = 138,070$).

4.2.2 Retrieving Profile Attributes

4.2.2.1 Facebook

Once the user ID of a Facebook profile has been obtained, there are several profile attributes that can be freely retrieved using Facebook’s Graph API, albeit in most cases an easily retrievable access token is required.

Among these attributes, dependent on a user’s privacy settings, are their full

name, cover photo, context (e.g. mutual likes), currency, age range, gender, locale (e.g. timezone), time the profile was last updated, profile picture, tagged photos, and uploaded photos. Unfortunately, tagged photos and uploaded photos were never returned by the API. The Facebook API documentation³ reports conflicting information about what permissions are necessary to retrieve these photos. On the one hand Facebook says that “Any valid access token for any photo with public privacy settings⁴” should work. On the other hand, the documentation states that a user’s explicit consent is required. After using Facebook’s Graph API explorer tool⁵, the console returned a debug message confirming the more strict permission requirements.

Using Facebook’s Graph API it is possible to retrieve full-size profile photos from any Facebook user, as long as the numerical user ID is known. In fact, it appeared possible to retrieve these photos even though a user explicitly sets the privacy settings of the profile photo to non-public. Moreover, it is not necessary to use any authentication, e.g. an access code from an app or a user⁶. The URL to retrieve full size profile photos is <https://graph.facebook.com/userID/picture?width=99999&redirect=true>. In contrast to the policy for retrieving profile photos, an access token is required to request a user’s cover photo (a banner-like photo, as explained in Chapter 2). Nonetheless, such tokens are considerably easy to obtain - anyone with a Facebook account can get one by creating an app on Facebook’s developers pages. Once someone is in possession of a token, they can request every user’s cover photo, since the privacy setting for these is always ‘public’.⁷.

4.2.2.2 Twitter

Twitter allows retrieval of most of their users’ information and content using their provided API. Users can be queried using their username, which they can change as often as they want (Mariconti et al., 2017). Some snippets of what the Twitter API returns⁸ are included in Listings 1, 2, and 3 on pages 48 to 49. For example, in Sec-

³<https://developers.facebook.com/docs/graph-api/reference/user/photos/>

⁴See footnote 3

⁵<https://developers.facebook.com/tools/explorer/>

⁶Detailed information on access tokens for Facebook graph API can be found here: <https://developers.facebook.com/docs/facebook-login/access-tokens>

⁷<https://en-gb.facebook.com/help/193629617349922>

⁸An exhaustive list of attributes can be found here: <https://dev.twitter.com/overview/api/entities-in-twitter-objects>

tion 4.3.2 I will use the attributes: profile description, public Tweets, and URLs in profile description and Tweets. The profile picture can be retrieved from a username by requesting http://twitter.com/username/profile_image?size=original.

4.2.3 Summary

In summary, I developed and implemented functions to successfully retrieve information from the Facebook and Twitter API. Moreover, I invented a fast algorithm that can retrieve Facebook user IDs given Facebook usernames with perfect results. These are problems with the new Facebook Graph API that, to the best of my knowledge, have not been solved before. Moreover, this section summarised what the current state of the Facebook (V2.10) and Twitter (V1.1) API are, including what publicly available information about users can actually be retrieved.

4.3 Finding Accounts

4.3.1 Identical Usernames

As explained in Section 4.2.1, a function has been written to convert Facebook usernames to numerical user IDs. This yields the possibility to query information using Facebook usernames. To find a user's Facebook account given their Twitter username, an easy approach is to request information from Facebook based on their Twitter username. To find out whether this is a reasonable approach, this section will show some statistics that were calculated using the dataset by Jain et al. (2013).

On a set of 138,097 Twitter and Facebook usernames, 31.42% used the exact same username for their Twitter account as on their Facebook account. When Facebook Pages (i.e. non-user profiles) were filtered out of this set, 116,520 users remained. It appeared that using the same username on different platforms is somewhat more common to people maintaining Facebook Pages, as the fraction of username reuse for user profiles decreased slightly to 30.76%, whereas a higher percentage of 35.00% was measured for Facebook Pages.

From the above measurements, it seems that, to find a user's Facebook account, given their Twitter account profile name, looking for the same profile name on Facebook is something that should be considered. In the dataset used for this

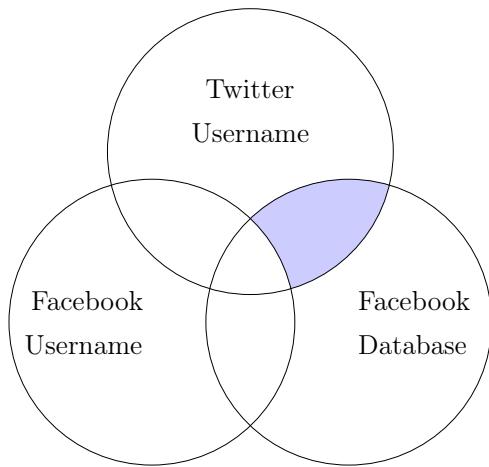


Figure 4.2: A person's Twitter and Facebook usernames are not the same, but the Twitter username exists on Facebook.

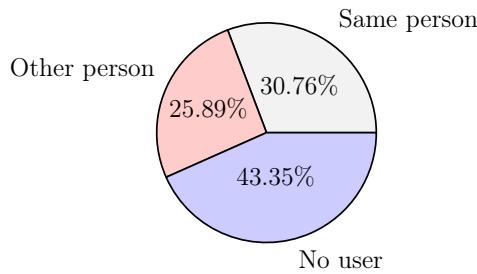


Figure 4.3: Results of querying users on Facebook by their Twitter username.

project, in nearly one third of the cases this would have resulted in the correct Facebook account. Nevertheless, it is questionable how representative our dataset is in comparison to real world examples. The dataset used for this project originally came from the discontinued project called ‘Google Social Graph’, which claims to only have formed connections based on those that were publicly declared. To deliberately form a link between two accounts, users might use the same username for both accounts on purpose. For example, to use it as a brand name.

Requesting a user on Facebook by their Twitter username could also be misleading. The username may also exist on Facebook, but may not belong to the same person. To find out how reasonable this concern is, a test was performed on our dataset of 116,520 linked user accounts. Of this dataset, 80,681 users used different usernames for their Facebook and Twitter accounts. Thus, if their Twitter usernames exist on Facebook, it would belong to a different Facebook user. The usernames’ existence on Facebook was checked. For clarity, Figure 4.2 shows a Venn diagram of the problem. As shown in Section 4.2, a function has been written to test whether a username redirects to a user profile, or a Facebook Page. We leverage this information to only consider Facebook user accounts (not Pages). It appeared that in 37.39% of the non-identical username pairs, the Twitter username was in use by another user on Facebook.

Figure 4.3 shows a summary of querying users on Facebook by their Twitter

username. As shown, when querying 116,520 user accounts on Facebook by their Twitter username, 25.89% of the queries (non-identical+identical) would return the wrong account. When querying Facebook users by their Twitter username, 30.76% of the queries would return us the same user, and 43.35% would not return a user at all.

To summarise, in almost one third of the cases it is feasible to find a user's Facebook account through their Twitter username. Nevertheless, additional steps need to be taken to assert the account actually belongs to the same person, since the wrong person might be returned in more than a quarter of the cases. As mentioned, these numbers should merely be used as an indication. Actual numbers may vary due to possible bias in the used dataset.

4.3.2 Self-mention of Facebook on Twitter

Chapter 3 showed that the participants of the focus group were relatively privacy aware. Yet, 20% of them included links in their profile description to other social media accounts (termed self-mention by (Jain et al., 2013) see Chapter 2). This shows that users of Twitter sometimes share links to their or others' Facebook accounts in their profile description or Tweets. Sometimes links are included on purpose, because they want their Facebook account to be found. However, as Chapter 3 also revealed, less obviously structured URLs such as abbreviated ones, or URLs surrounded by other text, may lead to accidental sharing of their Facebook account. Moreover, a user might not realise they are revealing their Facebook profile by linking to a picture, uploaded using their Facebook account.

To see how often Facebook URLs were shared in Tweets and descriptions, a dataset of approximately 48.2 million Tweets was scanned for Facebook URLs. The dataset consisted of randomly sampled Tweets, harvested between the 29th of March 2017 and the 12th of April 2017 by Brainnwave inc.

Each Tweet in the dataset has the profile attributes of the person who posted the Tweet attached to it. When a link appears in a profile description, it is usually stored as plain text, rather than using an appropriate JSON attribute. To illustrate this, Listing 1 shows an anonymised Tweet in which a Facebook URL was included in the profile description. Other places a link to a Facebook account can appear are the actual `url` attribute of the profile as shown in Listing 2, and in the `expanded_url` attribute of a Tweet, as shown in Listing 3. To deal with

```
{
  "created_at": "Mon Jan 01 00:00:00 +0000 2017",
  ...,
  "user": {
    ...
    "url": "http://www.personalwebsite.com",
    "description": "This is the official twitter account of alice.
https://www.facebook.com/alice Contact; alice@gmail.com",
    ...
  },
  ...
}

{
  "created_at": "Mon Jan 01 00:00:00 +0000 2017",
  ...,
  "user": {
    ...
    "description": "Award-winning original reporting on defence &
aerospace in Netherlands & the neighbourhood. | Ed: @johnsmith
| Also http://facebook.com/johnsmith | johnsmith@gmail.com",
    ...
  },
  ...
}
```

Listing 1: Tweet snippets with examples of the appearance of plain text Facebook links in profile descriptions.

this great variety of locations a URL could appear, including plain text, every line in the data, containing one Tweet each, was split on commas and spaces. Next, each line needed to be scanned for the regular expression `*facebook*`. Usually, the actual link was surrounded by clutter, which had to be removed in order to fully extract the link. Clutter includes `u' ' "`, `[] \n \r href=` and all unicode characters (e.g. `u"\u2220"` is unicode for \forall). Also, all Facebook links referring to the Twitter page had to be filtered, as they seemed irrelevant for the problem.

As Figure 4.4 shows, from all of the scanned Tweets (including profiles), it appeared that in total 2.98% of them contained a Facebook link. The dataset contained 1.76 million unique Twitter accounts, from which 307,531 links to different Facebook accounts were found. In total 1.44 million links to Facebook accounts were recorded. Comparing that latter figure to the number of unique Twitter usernames shows that 78.6% of the accounts were posted more than once.

```
{
  "created_at": "Mon Jan 01 00:00:00 +0000 2017",
  ...,
  "user": {
    ...,
    "url": "https://www.facebook.com/bob/",
  },
  ...
}
```

Listing 2: Tweet snippet with example of the appearance of a Facebook link in the profile URL attribute.

```
{
  "created_at": "Mon Jan 01 00:00:00 +0000 2017",
  ...,
  "entities": {
    "urls": [
      {
        "url": "https://t.co/2Ky4Zg6",
        "expanded_url": "https://www.facebook.com/jack/posts/1284249812688?pnref=story",
        ...
      }
    ],
    ...
  },
  ...
}
```

Listing 3: Tweet snippet with example of an extracted Facebook link from a tweet post. The link is nevertheless cluttered, as the username is what needs to be extracted.

This either means that many people have deliberately included their username, or that users who include it by accident do this more often.

The total amount of unique Facebook accounts over the total amount of Tweets kept declining, even after 48.2 million Tweets, see Figure 4.4. Since the dataset did not exclusively contain unique users, this ongoing decline can be explained by links appearing in profile descriptions. As these profiles can appear multiple times in the dataset by posting several Tweets, their profile descriptions, possibly

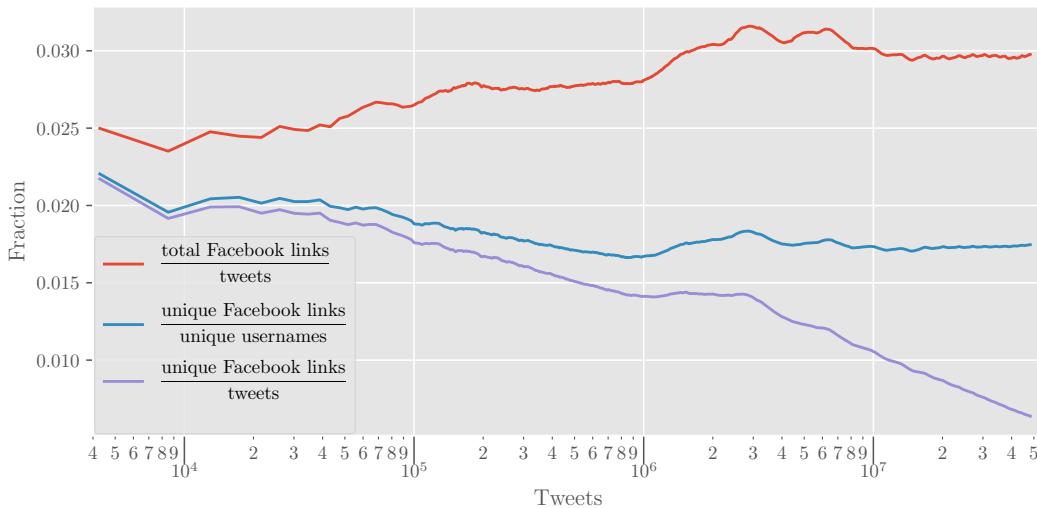


Figure 4.4: Results of scanning 48.2 million Tweets and Twitter profile descriptions for Facebook links. The values on the y -axis are the fractions after the amount of Tweets on the x -axis were processed. The scaling of the x -axis is logarithmic, the scaling of the y -axis is linear.

including a Facebook account, can appear more than once. Furthermore, users can post a link to their Facebook account more than once, i.e. include the link in multiple Tweets.

Extracted Facebook links from a user can be leveraged to retrieve the profile photo from the link on Facebook, and compare this to the Twitter user's profile photo, e.g. using perceptual hashing or face recognition, as shown in Section 4.4.2 and 4.4.3. A simpler method, that could be used initially, is to assert username similarity, see Section 4.4.1.

It should be noted that, although at a glance the majority of the algorithm returned actual usernames, in rare cases random characters were returned. It was however hard to filter this, and therefore, also hard to count. Although this might influence the statistics, it does not yield considerable problems for a real-world implementation of the algorithm, as the username to user ID converter I built would simply return a fixed number of zeros for non-existing usernames.

When Facebook links would refer to a public photo on Facebook, the IDs in the photos can be extracted using a method I wrote, based on an obsolete version from a Github source⁹. In short, the method works by extracting the photo album URL from the image link. Then, when retrieving the photo, the user IDs of the

⁹<https://github.com/chokepoint/fbpic2id>

tagged people can easily be extracted from the source code of the page, similarly to the methods described in Section 4.2.1. I have provided this method here as a proof-of-concept, but leave a more complete and thorough test for future work due to the time limitations.

In summary, a Facebook username is included in nearly 3% of Tweets and profiles. These usernames can be included in the candidate set. Potentially more information could be extracted by following URLs to Facebook; this could be considered for future work.

4.3.3 Browser Automation

As mentioned in Section 4.1, many functions of Facebook’s Graph API have been removed compared to what is retrievable in the browser. Several profile attributes require explicit permission from users to obtain them through Facebook’s API. Most importantly, the functionality of the API’s *search* function has been markedly reduced. In contrast, the search functionality users see when logged in to Facebook on their browsers has expanded in available query types over the past few years (e.g., Linshi, 2014; Statt, 2015; Dashevsky, 2017). Therefore, it would be useful to be able to make use of the full Facebook search functionality, as is given in the browser.

To make use of such functionality, methods needed to be found to use the browser in an automated way (i.e. without the intervention of a human). There are several options for doing this.

The first candidate is the oldest maintained browser Lynx¹⁰ (Davies, 2012). The advantage of this browser is that it is text-based (Dickey, 2017), can run in a terminal, and is therefore lightweight. The downside is that Lynx only supports HTML, and therefore no Cascading Style Sheets (CSS) or JavaScript. As Figure 4.5 shows, due to the latter, it was unfortunately not possible to log in to Facebook using Lynx.

Another candidate for browser automation is the widely used Selenium¹¹ (e.g. Englehardt and Narayanan, 2016; Gogna, 2014; Motwani et al., 2015). This software suite automates various browsers, including popular ones such as Chrome and Firefox through their WebDriver API. Selenium supports Python bindings, which were used since all of the code for the current project has been written in

¹⁰<http://lynx.invisible-island.net/>

¹¹<http://www.seleniumhq.org/>

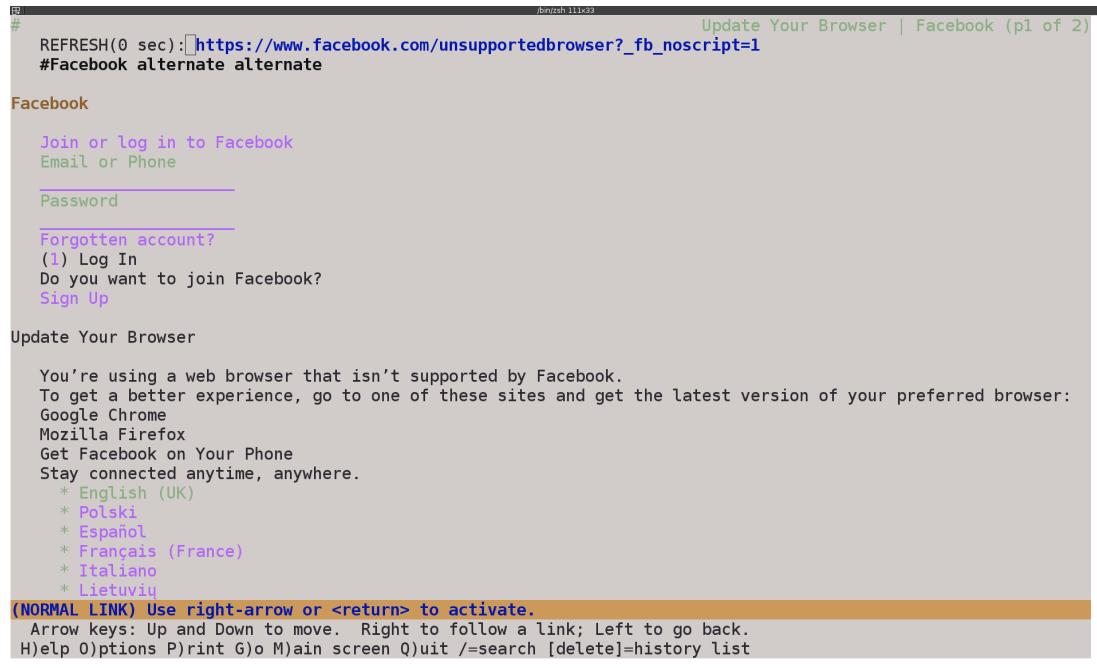


Figure 4.5: The page Facebook returns when accessing their website through Lynx browser.

Python as well. To avoid suspicion of Facebook, a full Firefox browser instance was used using the *geckodriver*¹², in contrast to running a headless browser.

Selenium WebDriver has several functions to search the source code for HTML or CSS elements by their attributes, and can even search for regular expressions in plain text. Further, it is possible to enter text in obtained fields, and send specific keys such as the return key.

As I will show below, Facebook may find out about the malicious activity, which may lead them to block the account. Therefore, firstly, a fake account was created using a fake Outlook¹³ email address. To convince Facebook that the account was real, a photo I took of my neighbour's cat was set as a profile picture, see Figure 4.6.

The code to initialise a Selenium WebDriver and subsequently logging in to Facebook was taken from a code snippet on Github¹⁴. Once logged in, it is not necessary to find the search element on the page. Rather, a new URL containing the search query could be requested from the WebDriver. For example, if one wants to search for 'John Smith', the URL <https://www.facebook.com/search/top/?q=John>

¹²<https://github.com/mozilla/geckodriver/releases> V0.18.0

¹³<https://signup.live.com/>

¹⁴<https://gist.github.com/ostera/3535568>



Figure 4.6: A photo of Tijger. This photo was used as a profile picture for the fake Facebook account.

Smith can be requested directly using the WebDriver. To specifically return people only, the word *top* in the URL can be changed to *people*. Requesting queries through URLs, however, introduces the problem that the WebDriver may start requesting queries before Facebook has finished the log in process. To deal with this, code was added to require the WebDriver to wait until the profile picture element is present, which appears immediately after log in completion.

In response to a query, Facebook initially returns a limited amount of results. More results will only show up when one scrolls down in the search results window. To deal with this, if the text ‘End of Results’ had not been found by WebDriver, JavaScript code was executed to scroll down until this text appears, or until a maximum amount of 5 attempts had been reached.

In the context of requesting many queries, loading the search results can take up considerable amounts of time. Therefore, it is important to immediately move on when no results are found. There are two messages Facebook can return when no results are found. One of these returns can be identified by an element id with `empty_results_error`. The other message by the text ‘*No results found for*’ appearing on the page. To not waste time on such futile queries, the query function is set to immediately continue to the next query when either of these messages is encountered.

When a search query returns results, the user IDs and usernames need to be extracted. Facebook stores user IDs in JSON format under the `id` key of an attribute called `data-bt` of a `div` class (a CSS block-element) with a seemingly random name, such as `_2yer_401d_2xje_2nuh`. As these class values seem too

random to be consistent, and the `data-bt` attribute does not appear anywhere else, `data-bt` was chosen to be looked for in the source code of the page. Finally, additional deception was added by making the program pause for a random amount of seconds in the interval [0.0, 2.0) between each query.

As will be shown in Section 4.4.1, comparing the usernames from the Twitter source account and the set of Facebook search results can be used for ranking the candidate set before they are passed on to a matching method. Therefore, if the usernames are included in the source code of the search results as well, it is useful to save them. Initially it seemed that the username is always contained in the source code of the page. In some cases it is returned under the only two hyperlink (`href`) attributes of the `div` block that contains the user ID. However, this appeared not to be true in all cases. Sometimes this link was simply a link to the profile, formatted using the user ID in the form of `https://facebook.com/profile.php?id=id_number`. Nevertheless, the username can be obtained by simply requesting their profile by their ID in the browser (i.e. following the aforementioned link), which should redirect to a link in the format `https://facebook.com/username`, from which the username can thus be extracted. Note, however, that doing this for each user in the search results can be rather slow. This will be elaborated upon in Section 4.3.3.1.

4.3.3.1 Name Search Results

From the relatively privacy aware participants in the focus group, 90% shared their first name on Twitter. Moreover, 80% even included their last name. As shown in Section 2.2, Facebook is seen as a more private environment than Twitter. In addition, Facebook requires real names to be used (Holpuch, 2015). It can thus be assumed that on Facebook an even larger percentage of the people include their real name. Therefore, full names fetched from a person's Twitter account can be searched for on Facebook to build a candidate set.

Although names can be searched for using Graph API, users can specifically opt-out to have their results shown here, or in any other external search engine¹⁵. It was found that even when users do not opt-out, they are sometimes still not shown in the API results. It does not seem to be possible to opt-out of Facebook's in-browser search. Therefore, a search from a browser instance should return more comprehensive results when a name is queried.

¹⁵Under the *Privacy* tab in *Settings* once logged-in.

With approximately 10 seconds per query using the configuration stated in Section 4.1, browser automation can be considered slow and in the time allowed impossible to test on the complete dataset of 138,097 users. Therefore, 1,959 unique users were randomly sampled to search for on Facebook. I then created a candidate set per user in three steps.

Firstly, I extracted their full names as they appear on Twitter. Secondly, I searched Facebook for these names using the *people* filter, as described above. Thirdly, I collected the user IDs using the methods described before, yielding a candidate set with on average 21 ± 9 user IDs per user.

To find the success rate, each user and their candidates were matched to the ground truth data of linked accounts. From the 1,959 users requested users, the correct user ID was included in the candidate set for 781 users (= 39.9%). For 421 users (= 21.5%) their username could be obtained from the source code as well.

To find out which account belongs to the user, methods for comparing accounts, such as profile picture comparison or username similarity, can be used, as will be shown in Section 4.4.

4.3.3.2 URL Search Results

In Section 3.3, it was found that more than half of the privacy concerned focus group participants includes content posted to other social media on their Twitter profile. As mentioned in Chapter 2, Jain et al. (2013) also showed that users can be found through such *cross-posting* behaviour. Particularly when certain content is only shared by a small amount of users, this can yield a discriminating candidate set.

Plain text included in content is likely to vary in length and wording when cross-posted, due to the 140 character limit on Twitter. On the other hand, URLs are fixed pieces of text. The in-browser search of Facebook allows searching for URLs on Facebook, and can thus be leveraged to search for content coming from Twitter users.

To search for users on Facebook through URLs they included in their Tweets on Twitter, the expanded URLs included in Tweets from the 1,959 users from the name search were retrieved. It appeared that on average users included 607 ± 754 URLs in their entire Tweet history. As mentioned in the previous section, running queries through browser automation is slow. Therefore, an attempt was made using the URL history of a few users.

In an attempt to speed up the process, and search for more URLs in parallel, an additional (isolated) browser window was started using WebDriver. Doing so made it clear that WebDriver requires a large amount of RAM, as it made the machine crash. At the same time, requesting queries in two browser windows with one account also attracted scrutiny to the Facebook account, and got it blocked. Since conducting a study on URL search through browser automation with a sufficient sample size would have been impossible in the limited amount of time given for the project, it was decided to discontinue the investigation.

Out of the final six users that were retrieved, selected based on having a relatively small amount of URLs per user, two correct user IDs were included in the candidate set. For one user it appeared that the person had cross-posted a unique Instagram URL to both their Twitter and Facebook accounts. The other user included a URL to a video. Both URLs were only shared by those particular users on Facebook. This could imply that searching Facebook through Twitter users' URLs returns a candidate set with higher precision, and could potentially replace account comparison, or reduce the amount of accounts to compare. However, it should be taken into account that the unique appearance of a URL on Facebook does not necessarily mean that it has been shared by the same person. Nevertheless, a potential candidate set could be matched using the methods described in the sections below.

4.4 Comparing Accounts

4.4.1 Username Similarity

In Section 4.3.1 it was revealed that many people use the same username for Facebook and Twitter. However, the username of users could also be non-identical, but similar. This information can be leveraged to filter or rank the candidate set, based on their username distance.

To find out how similar the usernames belonging to the same people are, the Levenshtein distance (Levenshtein, 1966) has been calculated on the dataset of 138,097 account pairs. This returns the distance between two strings t and f , where a length of $1L$ is added for each insertion, deletion, or substitution. Therefore, identical usernames would result in a Levenshtein distance of zero, two similar usernames would result in a small Levenshtein distance, whereas two

different usernames would result in a larger difference. Levenshtein's distance is recursively defined as follows:

$$d_{t,f}(i,j) = \min \begin{cases} d_{t,f}(i-1,j) + 1 \\ d_{t,f}(i,j-1) + 1 \\ d_{t,f}(i-1,j-1) + 1_{(t_i \neq f_j)} \end{cases} \quad (4.1)$$

where t is a Twitter username, and f the Facebook username from the same person. For each extra character, or substitution in f compared to t , a value of 1 is added to the distance d . On the other hand, if the i^{th} character of t is equal to the j^{th} character of f , nothing is added to the distance value. For instance, comparing the Twitter username 'goodstudent' to three Facebook usernames 'goodstudent1', 'goodstunt', and 'lovelymarker' would result in the following Levenshtein distances: $d('goodstudent', 'goodstudent1')$ is $1L$: one insertion; $d('goodstudent', 'goodstunt')$ is $2L$: two deletions; and $d('goodstudent', 'lovelymarker')$ is $11L$: ten substitutions and one insertion.

Although the Python implementation of Levenshtein distance that was used, `editdistance`¹⁶, takes into account the differences between upper and lowercase characters, it did not influence distances, as all usernames in the dataset were saved as lowercase.

Figure 4.7 shows that the average Levenshtein distance between two usernames on the larger set ($N = 138,097$), calculated using Equation 4.1 is $5.371L \pm 5.062$ including identical usernames, and 7.809 ± 4.254 excluding identical usernames. Comparing each username to 1,000 randomly chosen usernames from the dataset returned a mean Levenshtein distance of 11.29 ± 2.645 , which is expected as this is also the average username length. These average distances calculated only for the user accounts or the Facebook Page accounts within the dataset did not differ significantly from the results calculated over the whole dataset, and are hence not reported.

The results show that users often use similar usernames, or reuse parts of their usernames for their Twitter and Facebook accounts. Possible reasons for using a similar username could be that users genuinely want to be found, are privacy unconcerned, are privacy unaware, or simply because similar usernames are easier to remember.

¹⁶<https://pypi.python.org/pypi/editdistance>

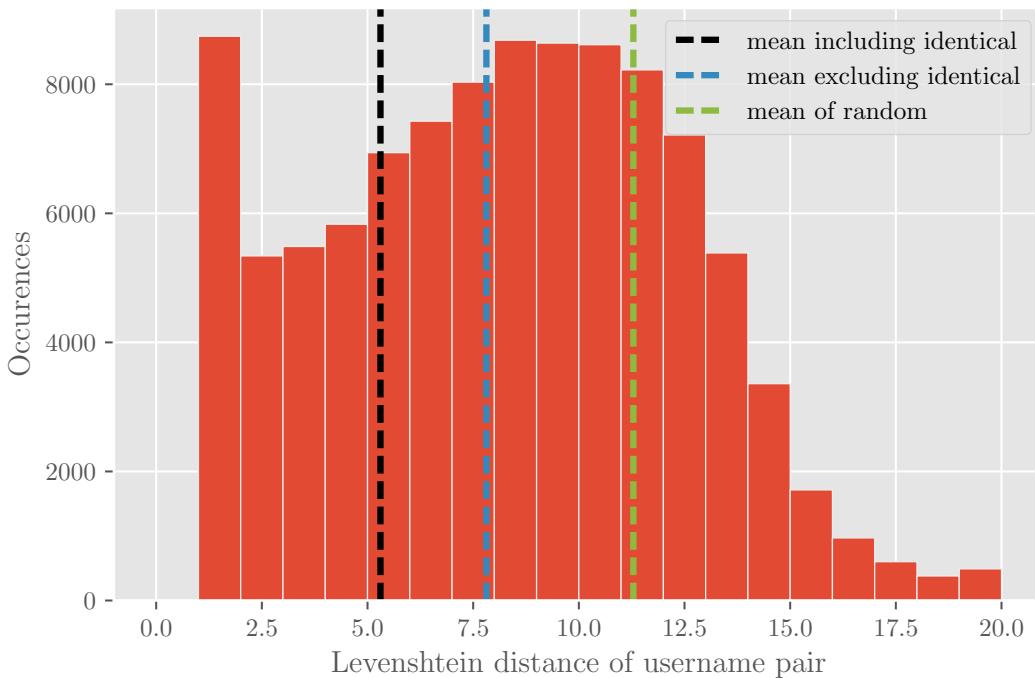


Figure 4.7: A histogram of the Levenshtein distances between the accounts of a person in the dataset. The black line shows where the average distance is including the identical usernames ($L \geq 0$). The blue line is the average distance between usernames excluding identical usernames ($L > 0$). The green line shows the average distance between each username compared to 1,000 randomly sampled usernames from the dataset.

As mentioned in Section 4.3.3, retrieving the usernames can take considerable amounts of time when usernames are not known. Nevertheless, the above shows that calculating the Levenshtein distance between candidates' and a user's usernames can provide beneficial information. Moreover, calculating the Levenshtein distance between usernames is computationally cheap ($1.73\text{e}{-06}$ seconds per comparison). Therefore, if the usernames are known, comparing them using the above described methods can be a successful approach to rank the candidate set, before they are matched using the more expensive algorithms pHash and Face Recognition, as will be described below.

4.4.2 Perceptual Hash

In Section 4.2.2, it was explained how profile images can be retrieved from both Facebook and Twitter. Images from two accounts, one from Twitter, and one

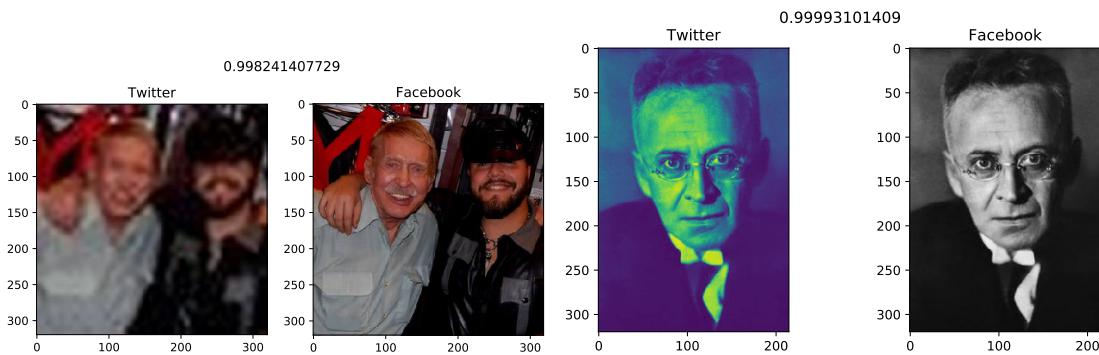


Figure 4.8: The same original images, but the left image being highly compressed.

Figure 4.9: The same original images, but one of them being transformed by altering the colours.

from Facebook, can be compared to see if they belong to the same person. A fast and relatively robust image comparison technique is *radial variance based perceptual hashing* (Zauner, 2010, p. 64-65). A commonly used implementation is pHash¹⁷ (Zauner, 2010, p. 28-29), which will be used for the current project as well. According to the developers of pHash, Klinger and Starkweather (n.d.),

“a perceptual hash is a fingerprint of a multimedia file derived from various features from its content. Unlike cryptographic hash functions which rely on the avalanche effect of small changes in input leading to drastic changes in the output, perceptual hashes are ‘close’ to one another if the features are similar”.

In other words, a Twitter and a Facebook profile photo could have been the same original photos. However, minor alterations to the images can highly influence the hash value of the image. Perceptual hashing is supposed to be robust against such transformations, including rotations, cropping, compression, or colour and contrast adjustments (Zauner, 2010, p. 64-65). For example, the images in Figure 4.8 and 4.9 were classified as very similar, both with more than 99.8% Peak of Cross Correlation (PCC), even though they are visibly altered. Peak of Cross Correlation is the maximum correlation measured using signal correlation, and normalised cross correlation (Zauner, 2010).

As discussed in Section 2.4, there are two main image hashing methods implemented by pHash: *discrete cosine transform (DCT)* and *radial hash projection (RHP)*. According to Klinger and Starkweather (n.d.), DCT is the most accurate. However, it appears to be 7.5 times slower than RHP (Zauner, 2010, p. 61). Zauner

¹⁷<https://www.phash.org>

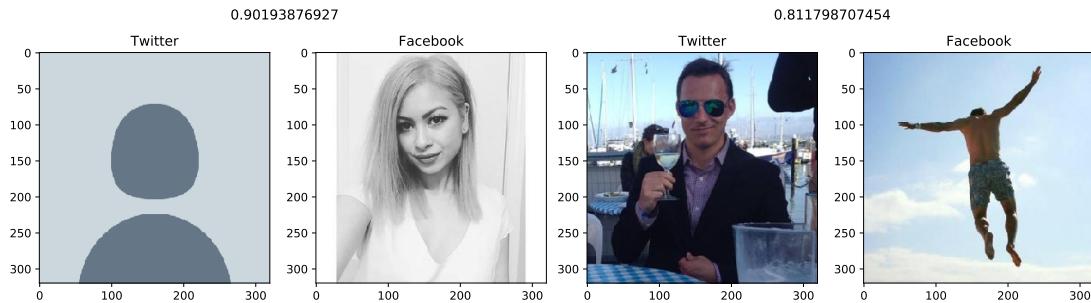


Figure 4.10: Two clearly mismatched images were pHash assigned relatively high PCC values, 0.90 and 0.81, respectively.

(2010) reports that on average, calculating a hash using DCT takes 9.7 seconds per image, whereas RHP merely needs 1.3 seconds. Moreover, RHP appears to be more robust against JPEG compression (Zauner, 2010). The faster RHP is more suitable for real-time web-based user requests, and will therefore be used.

Since most of the code for this project has been written in Python, an implementation of Python bindings for pHash was taken from a GitHub repository¹⁸. The Python bindings made it convenient to immediately compare images using RHP. It contains a function named `compare_images`, which takes two image filenames as an argument and returns the PCC score from their RHPs.

To test how well perceptual hashing using RHPs works for certain thresholds of PCC values, the images from 1,167 randomly sampled Twitter and Facebook account pairs were retrieved, and their PCC scores calculated. They were then placed into five categories according to their PCC values: [0 to 0.5] (1), [0.5 to 0.75] (2), [0.75 to 0.85] (3), [0.85 to 0.95] (4), and [over 0.95] (5). These class bounds were chosen based on Klinger and Starkweather (n.d.) finding a threshold of 0.91 to work best, and the default threshold (which slightly differs for unexplained reasons) being 0.90. Since lower values, e.g. class 1 and 2, are considerably distant from the before mentioned optimal value (0.90), the expectation is that their precision is less significant. Therefore, these classes are larger. Nevertheless, even for the default threshold of 0.90, pHash still made some seemingly obvious mistakes, such as shown in Figure 4.10.

Using the annotation task that will be described in Section 4.4.2.1, all of the images were classified by hand, to find how many images in each category were indeed the same, and how many images at least contained the same person, animal, or object. This yields a gold standard to which we can measure the accuracy of

¹⁸<https://github.com/polachok/py-phash>

pHash. Although pHash “probably cannot even detect similar artefacts from two different source files - e.g. two different photographs of the same person” (Klinger and Starkweather, n.d.), it might still detect faces from their geometrical features. Therefore, the performance of pHash on the recognition of faces or objects, rather than merely images, should be measured as well.

4.4.2.1 Annotation Task

The images were classified by two annotators separately: the author (1) and a PhD student in the psychology department (2). Annotator 1 classified all 1,167 image pairs. From the classes with many pictures, i.e. class 1 and 2, annotator 2 reviewed a randomly selected subset containing 20% and 30% of the image pairs, respectively. Annotator 2 reviewed all images in the remaining classes.

Each annotator individually viewed (a subset of) images containing a Facebook profile photo, and a Twitter profile photo side-by-side, see Figures 4.8,4.9, and 4.10. For each picture combination in each class, two types of classifications were considered: the two images contain the same face, object, or animal (1), the two images are or were originally the same (2). Note that classification 2 implies classification 1. Especially for the face, object, and animal classification task, it is important to note that the images are assumed to come from accounts belonging to the same person. Both annotators have this knowledge, yielding a prior that increases the likelihood that the photos belong to the same person. Therefore, in doubt, annotators can more confidently say that two faces are the same, merely based on facial features such as a nose, hairstyle, teeth, ears, and even accessories or distinguishable clothing. Figure 4.11 shows an example of two faces which clearly have the same nose and hairstyle. Without the high likelihood of these pictures belonging to the same person, certainty towards annotating them as the same face would drop. Another example of this is shown in Figure 4.12. Her face is barely visible in the photo on the right. However, her easily identifiable clothing makes it likely that it is the same person.

The second annotator reviewed a subset of the image pairs classified by annotator 1. Therefore, the second annotator provides validation of the first annotator’s annotations. Cohen’s kappa coefficient is used to test the inter-rater agreement of the two annotators (Cohen, 1960), defined by:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \quad (4.2)$$

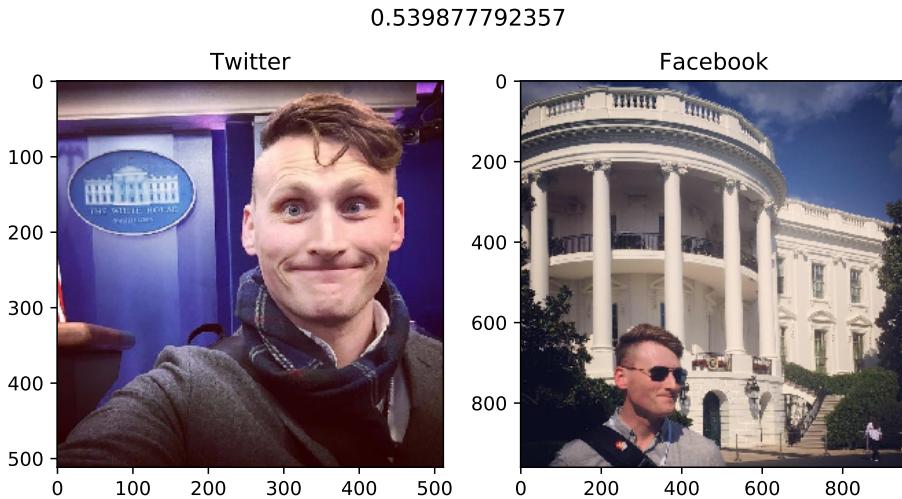


Figure 4.11: The same person with an occluded face on the right, but highly distinguishable hair and nose.

where we define the relative observed agreement p_o , and the probability of random agreement p_e as:

$$p_o^{(c)} = \frac{PPV_a^{(c)} + FPR_a^{(c)}}{PPV_a^{(c)} + FPR_a^{(c)} + PPV_{na}^{(c)} + FPR_{na}^{(c)}} \quad (4.3)$$

$$p_e = \overline{PPV_1} \times \overline{PPV_2} \quad (4.4)$$

where $PPV_a^{(c)} = \min(PPV_1^{(c)}, PPV_2^{(c)})$ and $FPR_a^{(c)} = \min(FPR_1^{(c)}, FPR_2^{(c)})$ depict the proportion of observations between the two annotators in agreement a per category c . $PPV_x^{(c)}$ is the positive predictive value (or precision) per class from annotator $x \in \{1, 2\}$, and $FPR_x^{(c)}$ their false positive rate (also known as false alarm rate) coming from the annotations. The relative amount of disagreement can be measured by calculating the difference between annotators' PPV and FPR per category, calculated as $PPV_{\neg a}^{(c)} = \max(PPV_1^{(c)}, PPV_2^{(c)}) - PPV_a^{(c)}$ and $FPR_{\neg a}^{(c)} = \max(FPR_1^{(c)}, FPR_2^{(c)}) - FPR_a^{(c)}$.

Due to the use of different categories for the images, the mean of relative observed agreement \bar{p}_o over categories is used to calculate κ . Cohen's kappa is calculated separately for the image classification and the face, object, or animal classification task.

If $\kappa \leq 0$, the annotations are below chance, thus annotators are considered to be in disagreement (Cohen, 1960). A κ score equal to 1 represents a perfect agreement between the reviewers.

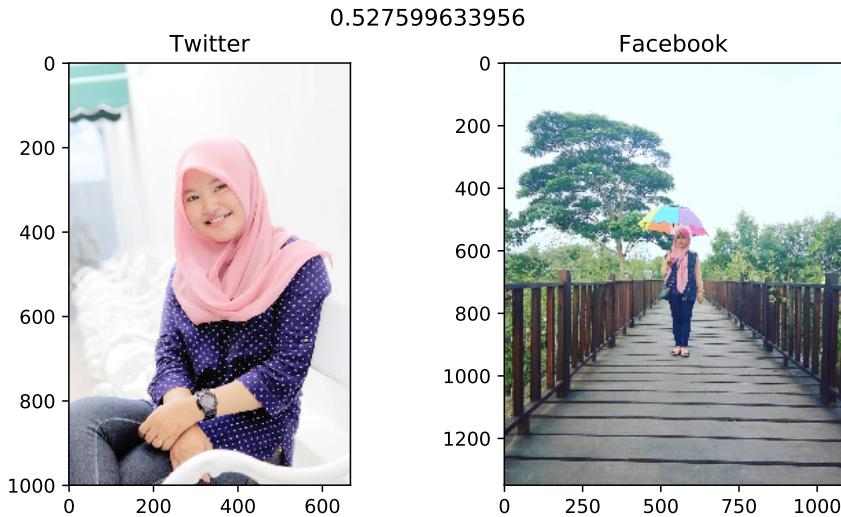


Figure 4.12: The same person with a barely visible face on the right, but easily identifiable clothing.

The annotators appeared to be in strong agreement on the face, object, or animal task ($\kappa = 0.73$). Taking all categories into account, the reviewers reached less, but still above chance, agreement on the image similarity task ($\kappa = 0.27$). The weaker agreement seemed to be caused by a disagreement in category 1 ([0 to 0.5]). Note that the image similarity task is supposed to be simpler than classifying content similarity of the image. Therefore, it is assumed that the difference in scores for the lowest category is due to randomness of the second annotator's subset. Leaving out category 1 shows excellent agreement ($\kappa = 0.75$).

From these kappa values, we can determine that the annotation task is well defined, and that the annotations of the first annotator can be trusted.

4.4.2.2 pHash Results and Discussion

Table 4.1 shows the results of RHPs PCC classifications, and the actual classifications made by annotator 1. As it is assumed that a higher PCC value threshold should return a better judgement accuracy, numbers were accumulated with their higher categories. For example, if there are 10 similar images in class 4, and 67 in class 5, then setting the threshold to > 0.85 would result in 77 correctly classified images. Because of this, category 1, which represents the data when all images are taken into account, shows that 10% of all account pairs indeed had the same picture, and 40% of all image pairs contain the same face, object or animal.

Table 4.1: The amount of manually classified ground truth (GT) similar images, or faces and objects, and their positive predictive value (PPV) in percentage, compared to pHash's judgement, based on different thresholds for the PCC value. The numbers presented are cumulative, i.e. all images classified in higher categories have been added to the numbers in lower categories. ‘faces’ includes faces, objects and animals occurring in both the Twitter and Facebook images. ‘images’ includes images coming from the same origin.

Cat.	PCC Value	GT image pairs			GT face pairs			pHash	
		PPV	F_1	#	PPV	F_1	#	#	
1	> 0.00	0.10	0.18	116	0.40	0.57	462	1,167	
2	> 0.50	0.21	0.34	103	0.49	0.50	239	488	
3	> 0.75	0.68	0.72	90	0.83	0.37	111	133	
4	> 0.85	0.88	0.75	77	0.97	0.31	85	88	
5	> 0.95	1	0.73	67	1	0.25	67	67	

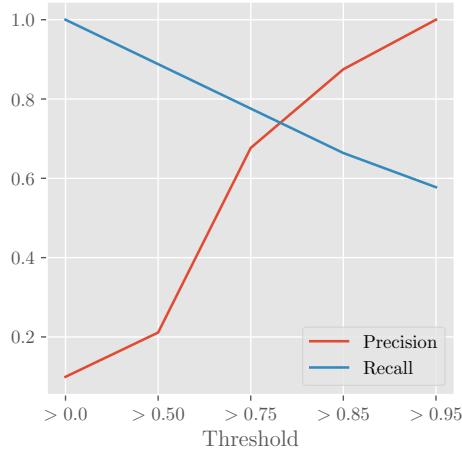


Figure 4.13: Precision and recall of image similarity classification per threshold of the PCC values pHash returns.

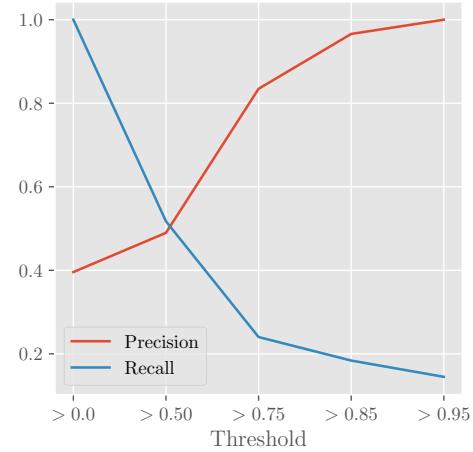


Figure 4.14: Precision and recall of face, animal, or object similarity classification per threshold of the PCC values pHash returns.

Figure 4.13 shows the precision (or PPV) and recall (see Section 2.1.1 for calculations) for image similarity when using different thresholds on the returned PCC values. The calculated F_1 scores from the precision and recall, shown in Table 4.1, indicate that the best threshold would be 0.75 and up. The F_1 scores

for the three highest threshold levels are similar. Moreover, precision and recall cross each other around the > 0.75 threshold level. Therefore, it would depend on whether precision or recall is valued more. A higher recall can be favourable for comparing user accounts when a second algorithm, such as username similarity or face recognition, would filter the remaining images. Nevertheless, if pHash is the sole classifier, higher precision, thus a higher threshold, might be desirable to avoid returning too many mismatches. It appears that for a PCC score of > 0.95 we can be certain that photos are the same. Therefore, if such values are obtained, the feedback system can immediately return this account as belonging to the user.

The thresholds can also be used for the user interface of the feedback system, to give an indication of how similar their Twitter and Facebook profile pictures are. Since a PCC value of > 75 returns similar F_1 scores, images with such high similarity could be classified as ‘likely similar’. Peak of Cross Correlation values of $0.5 \leq \text{PCC} \leq 0.75$ could be classified as ‘possibly similar’. Image similarities below 0.5 could return ‘likely dissimilar’.

Klinger and Starkweather (n.d.) correctly judged pHash on not being very suitable for face, object, or animal detection. Figure 4.14 shows that when picking a threshold of > 0.5 , this causes a steep decrease in recall compared to considering all images to be similar ($\text{PCC} > 0.0$). The amount of precision does not make up for this decline. One threshold category higher (> 0.75), the precision increases to above 0.8. However, the recall is merely 0.24. The F_1 scores, shown in Table 4.1, also show that considering all images would give the best results. Returning the highest F_1 score when considering all images suggests that pHash performs similar to random for recognising faces, objects or animals. Nevertheless, as said above, pHash was not designed as a face recognition algorithm, and it performs considerably well for recognising image similarity. Since most of the ‘face, object, and animal’ images contained faces, a powerful option could be to combine pHash with a face comparison method, which will be discussed in Section 4.4.3.

4.4.3 Face Recognition

Since the to be compared images are profile pictures, it can be assumed that most of these pictures contain faces. Therefore, a face recognition algorithm can provide an addition or alternative to perceptual hashing. As there is commonly only one picture of a person’s face available in the dataset, an algorithm that

can do *one-to-many* face comparison is required. Moreover, due to limited data, training a model was out of scope for this project. I therefore used a pre-trained model.

An implementation of a face recognition algorithm with key bindings in Python called *Face Recognition*, was found on Github¹⁹. The developers claim that this algorithm, based on the machine learning toolkit dlib²⁰ can achieve 99.38% accuracy on the Labeled Faces in the Wild²¹ benchmark. An explanation of how the face recognition algorithm works can be found in Chapter 2.

The authors of Face Recognition have also published their successful model, which will thus be used for the current project. To compare faces, they use a default euclidean distance threshold of 0.6. Note that this is not a linear scale, and hence different from the PCC comparison in perceptual hashing. Moreover, the face similarity function of Face Recognition returns a distance. Therefore, higher similarities are indicated by values closer to zero.

Firstly, an attempt has been made to find the best threshold for classifying 1,122 profile picture pairs, randomly picked from user pairs in the dataset of Jain et al. (2013). Image pairs are classified by hand on whether they contain the same face or not using annotation task 1 from Section 4.4.2.1, but without classifying objects and animals. A bin size of 0.1 was chosen to categorise the images, yielding 10 classes, [below 0.1] (1), [0.1 to 02] (2), [0.2 to 03] (3), ..., [0.8 – 0.9] (9), and [over 0.9] (10). Face Recognition does not return a score if it cannot detect faces in the images. If multiple faces were detected, I made the algorithm only return the shortest distance from the list of face distances.

Secondly, the optimal number of *jitters*, indicating the amount of times the algorithm re-samples the face, needs to be found. By default²², this value is set to 1. The documentation²³ mentions that a higher value should be more accurate, but makes the algorithm also linearly slower, i.e. re-sampling 100 times takes 100 times longer than sampling once. Due to time constraints, image pairs were only generated and classified for a jitter of 5, which should be better than the default, but can still be calculated in reasonable time. An extensive test on finding the optimal balance between accuracy and computation time could be conducted in

¹⁹https://github.com/ageitgey/face_recognition

²⁰<http://dlib.net/>

²¹<http://vis-www.cs.umass.edu/lfw/>

²²https://face-recognition.readthedocs.io/en/latest/face_recognition.html

²³See footnote 22

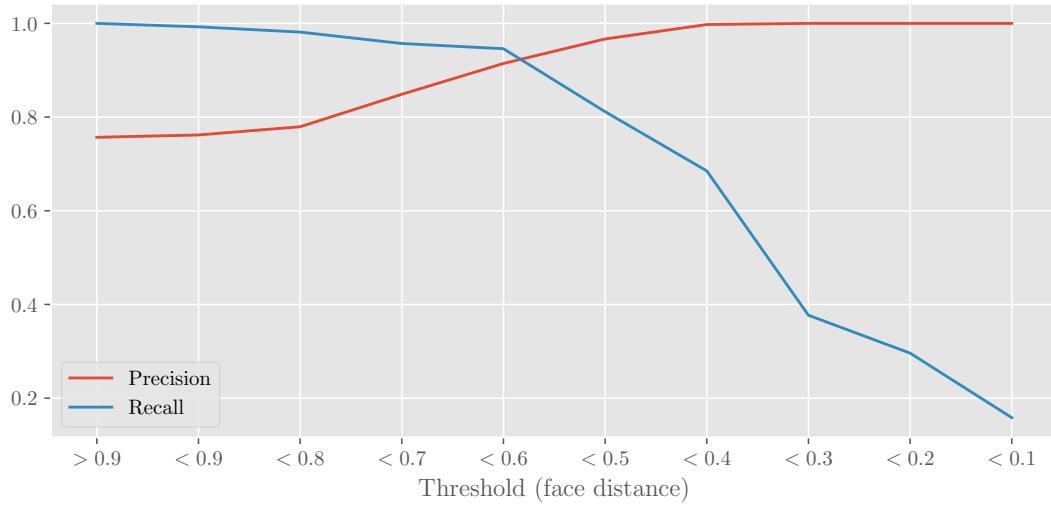


Figure 4.15: Precision and recall of face classification using Face Recognition per face distance threshold.

future work.

As mentioned, in most cases a one-to-many comparison will be performed between the Twitter user’s profile photo, and the photos from the candidate set. On the machine with specifications as stated in Section 4.1.1, the combination of retrieving the images, and subsequently performing face recognition takes on average 8.0 ± 7.5 seconds per image pair. Of this, the face recognition itself occupies 5.6 ± 2.7 seconds. Therefore, parallelising Face Recognition is desirable, and can be especially beneficial when running Face Recognition as a back-end to the feedback system on a cluster computer. It should be noted that these images were retrieved using the University of Edinburgh’s fast wireless internet connection, when parallelising Face Recognition over 4 threads. Using a slower connection would increase the time per image pair. The high standard deviation for the retrieval-recognition combination also indicates that downloading the images is the least stable factor. Explanations for this fluctuation are the use of a wireless connection, changes in response times of Facebook’s and Twitter’s servers, and differences in image size.

Table 4.2 shows the results for manually annotating the image pairs ($\kappa = 0.37$, a fair agreement) that were put into classes based on Face Recognition distance. In line with what the authors of Face Recognition wrote in their documentation²⁴, a threshold of 0.6 seems appropriate, since this yielded the highest F_1 score on

²⁴See footnote 19

Table 4.2: The amount of manually classified ground truth (GT) equal faces in a Twitter and Facebook profile photo, and their positive predictive value (PPV) in percentage. This is compared to Face Recognition’s judgement, based on different thresholds for the euclidean distances it calculates. The numbers presented are cumulative, i.e. all images classified in lower categories have been added to the numbers in higher categories, i.e. a threshold with higher tolerance contains more images.

Cat.	Distance	GT face pairs			Face Recognition #
		PPV	F_1	#	
1	< 0.1	1	0.27	92	92
2	< 0.2	1	0.46	172	172
3	< 0.3	1	0.55	219	219
4	< 0.4	1	0.81	398	399
5	< 0.5	0.97	0.88	581	601
6	< 0.6	0.91	0.93	716	783
7	< 0.7	0.85	0.90	757	892
8	< 0.8	0.78	0.87	791	1,015
9	< 0.9	0.76	0.86	806	1,058
10	> 0.9	0.76	0.86	812	1,073

the dataset. As shown in Figure 4.15, on average the recall drops with 15 percent point per stricter chosen threshold category. Nevertheless, when the threshold is set to < 0.5, a precision of 0.97 can be achieved, while still 81% of the images that contain matching faces are included. On the other hand, if an exclusive match is not desired by users of the feedback system, lower thresholds could be considered as well.

Note that the precision of 76% in the largest class of Table 4.2, which denotes the ground truth proportion of face pairs in the classified dataset, is inflated due to prior filtering of images not containing faces by Face Recognition. The total amount of Face Recognition queries ($N = 1,643$) should be used to calculate a new precision for the set containing all queried images. This returns a precision of 49%, which is more comparable to the amount of face pairs in the pHash dataset: 40% over the whole classified dataset. In retrospect, the same dataset should have been used for the evaluation of both pHash and Face Recognition, to make these results easier to compare. Note, however, that Face Recognition was still able to

correctly classify a larger proportion of the face pairs, with a higher recall.

In summary, this section reported on the results of using Face Recognition to compare profile images. The results are promising, as Face Recognition can correctly classify all faces above a certain threshold, while maintaining a remarkably high recall as well. Moreover, it can distinguish between images that contain faces and images that do not. Image pairs without faces could efficiently be passed on to a perceptual hashing algorithm for general image comparison.

4.5 Discussion of Implementations

In this chapter several novel methods for building a complete identity resolution system were described. Despite Facebook’s effort to incorporate privacy measures in their API, which has made several methods from related work obsolete, I have developed and showed several methods that can be used to retrieve users’ information from Facebook. Using a distribution of username similarity over a large dataset ($N = 138,097$), I showed how comparing usernames can help to find and compare Twitter users’ potential Facebook accounts using this simple measurement. In addition, with an even larger dataset ($N = 48.2$ million) I showed how, and how frequently, Facebook usernames can be retrieved from Tweets and Twitter profile descriptions. Furthermore, perceptual hashing and Face Recognition have been demonstrated to be successful methods for comparing and matching accounts. Finally, I also outlined which attributes from Facebook’s API can still be leveraged once a user ID has been obtained, and how to circumvent limitations imposed by changes to Facebook’s API using browser automation.

With 31.42% of the users in the dataset ($N = 138,097$) reusing their username, this provides a simple and effective method for finding Twitter users on Facebook. Scanning a user’s Tweets for self-mentions of Facebook could help to build a candidate set as well. However, merely 3% of the Tweets or profile descriptions were found to include possible self-mentions. Large candidate sets were retrieved by exploiting the possibilities offered by browser automation for Facebook’s in-browser search engine. Searching for full names returned the correct Facebook user ID in 40% of the resulting candidate sets. The results from the proof-of-concept of searching URLs seem promising as well.

Future work could explore how browser automation performs in real-world settings, and if it is feasible to use it on a long-term, and high demand basis.

Given the large amount of URLs per user, searching for URLs with browser automation seems especially impractical. On the other hand, intelligent ways to sample URLs from users could be explored to reduce the amount of browser requests. Additionally, the use of several VPNs and multiple Facebook accounts could be investigated to parallelise the process. An option to deal with the high memory use of Selenium would be to look into PhantomJS, a browser which might occupy less memory due to running headless. Additional research could also try to scroll down for more iterations when searching Facebook in-browser, this might return the correct user ID in the candidate set more often.

As mentioned, the Facebook URLs included in Tweets could also reveal information when following the links to, for example, a Facebook photo. A proof-of-concept on how tagged users can be extracted from photos has briefly been explained.

Although Face Recognition takes more time per image to compute than perceptual hashing, the results are better, as it returned more account pairs with relatively high confidence and high recall. Images that are considered not to have faces by Face Recognition could be passed on to pHash for image comparison in a two step scenario, which would lead to greater accuracy than using pHash alone. Furthermore, in future work also the matching of cover photos can be investigated. Cover photos are expected to have faces less frequently than profile pictures, and hence it would make sense to compare these using perceptual hashing.

All of the methods explained in this chapter can be tied together as shown in the framework in Figure 4.1. It could then run in the back-end of a website to generate personal feedback for social media users on how their accounts can be linked.

Chapter 5

Discussion

For this study research has been conducted to find appropriate feedback that will help users avoid being identified across different social media platforms. The design requirements for a feedback system have been obtained through two focus groups. To the best of my knowledge, no other study has investigated the design requirements for such feedback systems. Since this system needs personalised feedback, methods needed to be developed to find and match users' Facebook accounts, given their Twitter username. However, the freely obtainable information from Facebook has significantly changed since previous studies were conducted. Therefore, this study also investigated what user information is currently publicly retrievable through Facebook and Twitter.

During the focus group, the participants suggested several ideas for informing social media users with average computer competency. Participants agreed that information used to form a link between two accounts, which is obvious to most average users, should be excluded from the feedback. Some of this obvious information has also been identified. Moreover, to not overwhelm users of the feedback system, the amount of information should be reduced by providing merely relevant and personalised information. Furthermore, the participants thought that the use of visual examples would be appreciated. Some participants wished for a system in which they can compare their results in terms of matching to other people's results. This would add more context to the feedback.

Although the need for personal feedback was emphasised, general information for users could help them as well. This information could include methods for matching accounts that most of the average users would not have thought about. An example, mentioned during the focus group, is the use of face recognition to

match accounts. Additional information on URLs, and the scope of who can see their content might be appreciated by the users of the system as well.

For the identity resolution system, the limits for retrieving information from Twitter's and Facebook's API have been explored. This needed to be done to find out what methods are still applicable after other researchers such as Correa et al. (2012), Jain et al. (2013), and Goga et al. (2013) conducted their studies. Further, several methods for finding Facebook accounts given a Twitter account were explored, including a synthetic search for identical usernames, finding Facebook usernames in Tweets, and browser automation to search Facebook for full names and URLs. This yields a candidate set of Facebook accounts which can be compared with the user's Twitter account. Profiles can then be matched with username similarity, image comparison, or face recognition. Most of these methods have, to the best of my knowledge, not been explored before.

The implementations for identity resolution described in Chapter 4 can successfully fulfil some of the design requirements outlined in Chapter 3.

Since users' own accounts can now be matched with a certain precision through several means, personalised information can be shown that applies to vulnerabilities detected in their accounts. Moreover, using this information, personalised examples can be created. I have developed and demonstrated the results of methods to find users based on their full names, usernames, content that includes URLs, and self-mentions.

Perceptual hashing and Face Recognition use scores to indicate similarity between images or faces. Therefore, the system would be able to show where a user is according to their profile picture on a scale between anonymous and public figure. Username similarity could also be taken into account to measure this.

If the system is very certain about a match, for example because of a low face distance, visual examples, for instance, arrows drawn from their Twitter profile to their Facebook profile that provoked the link can be created.

Face recognition and perceptual hashing can also be used as separate tools for the feedback system. For example, a page separated from the main page can be created where users can upload two photos, and get the similarity returned. This can help users prevent the use of images that can be resolved to the same origin.

The above shows that the developed identity resolution implementations can deal with all the design requirements from Chapter 3 that require such implementations for the back-end of the feedback system. Nevertheless, additional

methods could be developed to obtain a higher success rate for finding users' Facebook profiles given a Twitter account.

Moreover, to actually make the system work, the methods need to be combined. This could be done by connecting the different methods proposed in Chapter 4 in the most optimal manner. Potentially additional methods to find and match accounts should be taken into account as well. These methods include the exploitation of the possibilities that browser automation gives, including, among others, content search, URL search, connections of users, and the retrieval of additional photos. Moreover, the Facebook picture to user ID method described in Chapter 4 can be used to find user IDs of users tagged in photos. Means for exploiting the information that this algorithm might yield could be explored.

Furthermore, the front-end of the system needs to be designed, built, and evaluated. One way to evaluate the front-end would be through several think aloud studies, in which users express confusion, relief, or frustration while completing a set of relevant tasks on the system (Hanington and Martin, 2012). Additional information needs to be found to add to the user interface as well, such as a frequently asked questions page, and how to optimise Twitter and Facebook's privacy settings.

Chapter 6

Conclusion

This thesis demonstrated several methods which form a building block for providing social media users with feedback on staying anonymous. Chapter 1 described the importance for users to remain anonymous, and showed that this is not a trivial task. Even when information is not published on social media platforms where one desires to keep certain information private, potential additional information can be retrieved by resolving accounts from different platforms to one identity.

No systems exist to give users personalised feedback on how they can prevent leaking of personally identifiable information. Therefore, the design requirements for such a feedback system have been collected and analysed, and the results were outlined.

Furthermore, a review of the state-of-the-art on identity resolution systems has been given. I found that many of such systems would not work anymore, given the current Facebook API. Therefore, I successfully proposed, implemented, and tested novel methods to retrieve information from Facebook.

Two datasets were used to evaluate the results. The first dataset ($N = 138,097$) contains a ground truth set of Facebook and Twitter account pairs belonging to the same person. The second data set consists of 48.2 million random Tweets (Twitter posts) from 1.76 million different Twitter accounts.

On the first dataset, I showed how to achieve 100% precision on retrieving Facebook users' numerical IDs, given their usernames. Having these IDs allowed me to query Facebook users by their username. In the dataset, it appeared that 31.42% of the users use the same username for Twitter and Facebook. Therefore, requesting users on Facebook using their Twitter username helps with finding associated accounts.

A proof-of-concept for the use of browser automation to find additional information on Facebook has been conducted. I have shown that, despite a user's privacy settings disallowing such search behaviour, Facebook can be queried using full names in an automated way. Using this method on a set of 1,959 randomly sampled users, 40% of the candidate sets contained the true user ID. Searching Facebook for URLs that Twitter users have included in Tweets has been explored as well. Nevertheless, I show that the need for performing full browser requests makes browser automation slow, especially with many URLs per user. Due to time limitations, a comprehensive study towards identity resolution through these means fell outside of the current project's scope.

Using a script that I designed, on the second dataset I showed that nearly 3% of the Tweet content or profile descriptions included a URL to a Facebook account.

Furthermore, username comparison using Levenshtein distance shows that a user's own Twitter and Facebook usernames correlate more than one of their usernames to others'. This could give an indication of how the candidate set can be ranked.

Finally, I enhanced existing methods for matching Facebook accounts to Twitter accounts with more advanced methods than had been used in related work. Moreover, I showed their potential when they would be used in an identity resolution system. I showed that perceptual hashing can be used to compare image pairs, but not faces, animals, or objects from different images. The performance of Face Recognition was therefore explored as well, which can be used in addition to perceptual hashing to accurately match faces.

Combining these methods would result in a system that could deal with all of the feedback system's design requirements that need content generated by an identity resolution system. The remaining design requirements that need to be satisfied are static pages that include explanations of several concepts related to anonymity on social media.

Bibliography

- Acquisti, A. and Gross, R. (2006). Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In *Privacy Enhancing Technologies*, Lecture Notes in Computer Science, pages 36–58. Springer, Berlin, Heidelberg.
- Ahmed, N., Natarajan, T., and Rao, K. R. (1974). Discrete Cosine Transform. *IEEE Transactions on Computers*, C-23(1):90–93.
- Borsboom, B., van Amstel, B., and Groeneveld, F. (2010). Please Rob Me. <http://pleaserobme.com/>. [online; accessed 03 April 2017].
- Cantador, I., Fernández-Tobías, I., Berkovsky, S., and Cremonesi, P. (2015). Cross-Domain Recommender Systems. In Ricci, F., Rokach, L., and Shapira, B., editors, *Recommender Systems Handbook*, pages 919–959. Springer US. DOI: 10.1007/978-1-4899-7637-6_27.
- Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.
- Correa, D., Sureka, A., and Sethi, R. (2012). WhACKY! - What anyone could know about you from Twitter. In *2012 Tenth Annual International Conference on Privacy, Security and Trust*, pages 43–50.
- Dashevsky, B. E. (2017). 24 Hidden Facebook Features Only Power Users Know. <https://www.pcmag.com/slideshow/story/324797/19-hidden-facebook-features-only-power-users-know>. [online; accessed 28 July 2017].
- Davies, M. (2012). What browsers other than IE and NN are there? <http://www.html-faq.com/browser/?browsers>. [online; accessed 03 August 2017].
- Dennen, V. P. (2008). Pedagogical lurking: Student engagement in non-posting discussion behavior. *Computers in Human Behavior*, 24(4):1624–1633.

- Dickey, T. E. (2017). Lynx - The Text Web-Browser. <http://lynx.invisible-island.net/>. [online; accessed 03 August 2017].
- Eckersley, P. (2010). How Unique Is Your Web Browser? In *Privacy Enhancing Technologies*, pages 1–18. Springer, Berlin, Heidelberg.
- Englehardt, S. and Narayanan, A. (2016). Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1388–1401. ACM.
- Facebook inc. (2010). Automated Data Collection Terms. https://www.facebook.com/apps/site_scraping_tos_terms.php. [online; accessed 23 July 2017].
- Facebook inc. (2014). Facebook platform changelog. <https://web.archive.org/web/20140521164802/https://developers.facebook.com/docs/apps/changelog>. [online; accessed 23 July 2017].
- Garfinkel, S. and Lipford, H. R. (2014). *Usable security: History, themes, and challenges*. Morgan & Claypool Publishers, San Rafael, California.
- Geitgey, A. (2016). Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning. <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>. [online; accessed 10 August 2017].
- Goel, V. (2014). Some Privacy, Please? Facebook, Under Pressure, Gets the Message. *The New York Times*.
- Goga, O., Lei, H., Parthasarathi, S. H. K., Friedland, G., Sommer, R., and Teixeira, R. (2013). Exploiting Innocuous Activity for Correlating Users Across Sites. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13*, pages 447–458, New York, NY, USA. ACM.
- Gogna, N. (2014). Study of browser based automated test tools watir and selenium. *International Journal of Information and Education Technology*, 4(4):336.
- Hanington, B. and Martin, B. (2012). *Universal methods of design: 100 ways to research complex problems, develop innovative ideas, and design effective solutions*. Rockport Publishers, Beverly, MA.

- Harris, L. and Westin, A. F. (1991). Harris-Equifax Consumer Privacy Survey 1991. *Atlanta, GA: Equifax Inc.*
- Holpuch, A. (2015). Facebook adjusts controversial 'real name' policy in wake of criticism. *The Guardian*. [online; accessed 14 August 2017].
- Iofciu, T., Fankhauser, P., Abel, F., and Bischoff, K. (2011). Identifying Users Across Social Tagging Systems. In *ICWSM*.
- Irani, D., Webb, S., Li, K., and Pu, C. (2009). Large Online Social Footprints—An Emerging Threat. In *2009 International Conference on Computational Science and Engineering*, volume 3, pages 271–276.
- Jackson, B. and Pesce, L. (2012). I Can Stalk U - Raising awareness about inadvertent information sharing. <http://icanstalku.com/>. [online; accessed 05 August 2017].
- Jain, P. (2015). Automated Methods for Identity Resolution Across Heterogeneous Social Platforms. In *Proceedings of the 26th ACM Conference on Hypertext & Social Media, HT '15*, pages 307–310, New York, NY, USA. ACM.
- Jain, P., Kumaraguru, P., and Joshi, A. (2013). @I Seek 'Fb.Me': Identifying Users Across Multiple Online Social Networks. In *Proceedings of the 22Nd International Conference on World Wide Web, WWW '13 Companion*, pages 1259–1268, New York, NY, USA. ACM.
- Jain, P., Rodrigues, T., Magno, G., Kumaraguru, P., and Almeida, V. (2011). Cross-Pollination of Information in Online Social Media: A Case Study on Popular Social Networks. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 477–482.
- Jaro, M. A. (1978). *Unimatch: A record linkage system: Users manual*. Bureau of the Census.
- Johnson, M., Egelman, S., and Bellovin, S. M. (2012). Facebook and Privacy: It's Complicated. In *Proceedings of the Eighth Symposium on Usable Privacy and Security, SOUPS '12*, pages 9:1–9:15, New York, NY, USA. ACM.

- Kaczynski, A. (2017). How CNN found the Reddit user behind the Trump wrestling GIF. <http://www.cnn.com/2017/07/04/politics/kfile-reddit-user-trump-tweet/index.html>. [online; accessed 05 August 2017].
- Kaplas, J. (2016). Possibilities and usability of various privacy preservation browser add-ons. *Bachelor's Thesis, Lappeenranta University of Technology*.
- Kazemi, V. and Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874.
- Klinger, E. and Starkweather, D. (n.d.). pHSh.org: Home of pHSh, the open source perceptual hash library. <http://www.phash.org/>. [online; accessed 20 July 2017].
- Krishnamurthy, B. and Wills, C. E. (2009). 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. ACM.
- Kumaraguru, P. and Cranor, L. F. (2005). Privacy indexes: a survey of Westin's studies. *Institute for Software Research International, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Technical Report CMU-ISRI-5-138*.
- Lefebvre, F., Macq, B., and Legat, J.-D. (2002). RASH: Radon soft hash algorithm. In *Signal Processing Conference, 2002 11th European*, pages 1–4. IEEE.
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.
- Linshi, J. (2014). Facebook's Search Function Just Got So Much Better. <http://time.com/3623964/facebook-search-old-posts/>. [online; accessed 28 July 2017].
- Malhotra, A., Totti, L., Jr, W. M., Kumaraguru, P., and Almeida, V. (2012). Studying User Footprints in Different Online Social Networks. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1065–1070.

- Mariconti, E., Onaolapo, J., Ahmad, S. S., Nikiforou, N., Egele, M., Nikiforakis, N., and Stringhini, G. (2017). What's in a Name?: Understanding Profile Name Reuse on Twitter. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1161–1170, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.
- Moreau, E. (2017). The Top Social Networks People Are Using Today. <https://www.lifewire.com/top-social-networking-sites-people-are-using-3486554>. [online; accessed 07 August 2017].
- Motoyama, M. and Varghese, G. (2009). I Seek You: Searching and Matching Individuals in Social Networks. In *Proceedings of the Eleventh International Workshop on Web Information and Data Management*, WIDM '09, pages 67–75, New York, NY, USA. ACM.
- Motwani, A., Agrawal, A., Singh, N., and Shrivastava, A. (2015). Novel Framework for Browser Compatibility Testing of a Web Application using Selenium. *International Journal of Computer Science and Information Technologies*, 6(6):5159–5162.
- Navarro, G. (2001). A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88.
- Obar, J. A. and Wildman, S. (2015). Social media definition and the governance challenge: An introduction to the special issue. *Telecommunications Policy*, 39(9):745–750.
- Ozsoy, M. G., Polat, F., and Alhajj, R. (2015). Modeling Individuals and Making Recommendations Using Multiple Social Networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 1184–1191, New York, NY, USA. ACM.
- Pass, G. and Zabih, R. (1999). Comparing images using joint histograms. *Multi-media systems*, 7(3):234–240.
- Perito, D., Castelluccia, C., Kaafar, M. A., and Manils, P. (2011). How Unique and Traceable Are Usernames? In *Privacy Enhancing Technologies*, pages 1–17. Springer, Berlin, Heidelberg.

- Rader, E. J. (2014). Awareness of Behavioral Tracking and Information Privacy Concern in Facebook and Google. In *SOUPS*, pages 51–67.
- Radon, J. (1986). On the determination of functions from their integral values along certain manifolds. *IEEE transactions on medical imaging*, 5(4):170–176.
- Roesner, F., Kohno, T., and Wetherall, D. (2012). Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 12–12. USENIX Association.
- Rojas Q., M., Masip, D., Todorov, A., and Vitria, J. (2011). Automatic Prediction of Facial Trait Judgments: Appearance vs. Structural Models. *PLoS ONE*, 6(8):e23323.
- Rose, C. (2011). The Security Implications Of Ubiquitous Social Media. *International Journal of Management and Information Systems; Littleton*, 15(1):35–40.
- Saldaña, J. (2009). *The coding manual for qualitative researchers*. Sage, Los Angeles, Calif.
- Standaert, F.-X., Lefebvre, E., Rouvroy, G., Macq, B., Quisquater, J.-J., and Legat, J.-D. (2005). Practical evaluation of a radial soft hash algorithm. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on*, volume 2, pages 89–94. IEEE.
- Statt, N. (2015). Facebook is unleashing universal search across its entire social network. <https://www.theverge.com/2015/10/22/9587122/new-facebook-search-all-public-posts>. [online; accessed 28 July 2017].
- Stutzman, F., Gross, R., and Acquisti, A. (2013). Silent listeners: The evolution of privacy and disclosure on facebook. *Journal of privacy and confidentiality*, 4(2):2.
- Twitter inc. (2017a). About public and protected Tweets. <https://help.twitter.com/articles/14016>. [online; accessed 09 April 2017].
- Twitter inc. (2017b). Posting links in a Tweet. <https://support.twitter.com/articles/78124>. [online; accessed 10 July 2017].

- Twitter inc. (2017c). Posting photos or GIFs on Twitter. <https://support.twitter.com/articles/20156423>. [online; accessed 09 August 2017].
- Twitter inc. (2017d). Twitter - Company. https://about.twitter.com/en_us/company.html.
- Vaniea, K. (2016). Human-computer interaction in-class survey. <https://www.inf.ed.ac.uk/teaching/courses/hci/1617/materials/survey.pdf> [online; accessed 16 July 2017].
- Wallace, G. K. (1992). The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv.
- Wang, Y., Norcie, G., Komanduri, S., Acquisti, A., Leon, P. G., and Cranor, L. F. (2011). I regretted the minute I pressed share: A qualitative study of regrets on Facebook. In *Proceedings of the seventh symposium on usable privacy and security*, page 10. ACM.
- Welch, C. (2017). Facebook crosses 2 billion monthly users. <https://www.theverge.com/2017/6/27/15880494/facebook-2-billion-monthly-users-announced>. [online; accessed 07 August 2017].
- Wong, J. C. (2017). Government seeks to unmask Trump dissident on Twitter, lawsuit reveals. *The Guardian*. <https://www.theguardian.com/technology/2017/apr/06/twitter-lawsuit-anonymous-account-trump-alt-uscis> [online; accessed 08 April 2017].
- Zauner, C. (2010). *Implementation and Benchmarking of Perceptual Image Hash Functions*. Master's thesis, Austria.

Glossary

Online Social Media Platforms

Websites and applications where users create, share, and react to each others' content.

Content	Content includes text, URLs, photos, videos, and locations.
Post	Content shared by a user.
Username	An often semi-permanent nickname a user gives themselves.
User ID	A permanent numerical ID appointed to an account.
Profile	An account and the attributes containing the information about a person, such as name and <i>profile photo</i> .
Profile page	A web page were a user's individual profile and content are shown.
Profile photo	A photo, normally containing the face of the person behind the account.
Tagging	The act of annotating content with someone else's profile.
Cover photo	A photo, commonly in the form of a banner at the top of someones profile page.
Full name	The first and last name (when given) included in the profile.
Connections	Can be mutual, e.g. friendship, or one-way, e.g. following a user on the platform.
Privacy settings	User settings to control who can see what content or profile attributes, and who can search them through which engines.
Fake account	An account deliberately not containing any information about one's true identity, often used for malicious activity.

Facebook

World's most popular social media network. Connections commonly include friends, family, and acquaintances. Data is generally less public than on Twitter.

User-profiles/accounts	Personal profiles or accounts, i.e. for individual users.
(Facebook) pages	Non-personal, public 'profiles' which often represent an organisation, artist, or other non-personal Facebook users.
Friends	A mutually accepted connection between two user-profiles.
Graph API	The Application Protocol Interface yielding access to Facebook's database using computer programming queries. With the deprecation of V1.0 in 2015, a great deal of publicly retrievable information cannot be retrieved through this API anymore.
Facebook App	An application published on Facebook, necessary to get access to the Graph API.
App access token	An access token that allows using Facebook's Graph API which can automatically be renewed. App access tokens can retrieve slightly less of users' public information by default than a user access token.
User access token	An access token that allows using Facebook's Graph API which expires with one or two hours. It can obtain all public profile attributes that Graph API allows.
In-browser search	The search engine that is available to users of the platform. This engine can search all publicly available content and profiles, and those of friends of the user.

Twitter

A social media platform that encourages making content publicly available. They are well known for their hashtags to categorise content. Nearly all public data on

Twitter is freely accessible through their API.

Screen name	Same as username. Users can change their screen names, and discarded names become available.
Tweet	A post on Twitter, containing at most 140 characters.
Profile description	A description on the user's profile page describing the user.
Followers	Connections on Twitter. Anyone can follow any public profile without the need for mutual agreement.
URL abbreviation	URLs on Twitter are often shortened. E.g. http://twitter.com would become t.co/xyz .

Web and Browser Related

URL	A URL or web address is a string referring to a location on the internet. E.g. http://www.ed.ac.uk .
HTML	The primary text-based language for creating rich text on web pages.
CSS	Style sheet language to enrich the presentation of a web page.
Headless browser	Web browser without a user interface, which is able to obtain and interact with web pages in an automated manner.
Full browser	Web browser with graphical user interface. Well known are Mozilla Firefox, Google Chrome, and Microsoft Edge.
Browser automation	Automating interactions with web pages that are obtained by a browser, such as entering credentials and clicking elements, without the need for user intervention.
(REST) API	An Application Protocol Interface (API) provides functions to retrieve data from an application or service. E.g., to retrieve profile photos from Twitter or Facebook.

(Hyper)links	A piece of text that, when clicked on, takes one to a URL.
Scraping	The act of downloading large amounts of data from a website or service.
Metadata	Data that describes (attributes of) other data.
Source-code	Uncompiled programming code, sometimes including comments, presented in plain text.

Other

Candidate set	Set of Facebook accounts that may belong to the Twitter user. The true account should be obtained by using a matching method.
Face Recognition	Finding characteristics of faces in images using machine learning, or a previously generated model, such that faces can be compared and identified.
Perceptual Hashing	Converting an image to a hash based on characteristics in the image, rather than pixels.

Appendix A

Focus Group Survey and Consent Form

The following pages contain the original questionnaire, followed by the consent form participants filled out and signed during the focus group, and finally the original ethics application.

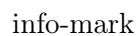
infotext

info-cross:



info-select

info-correct:



info-mark

Thank you for participating in this survey and focus group. You are free to choose not to answer any question. The researchers: Alexander Caughey and Timo Mulder, commit to process your information anonymously. Any information used in the final report will be anonymised and used to describe a group rather than individuals.

1 Demographics

1.1 What is your gender?

- Female
- Male
- Other
- Prefer not to say

1.2 What is your age in years?

1.3 What is your Nationality?

1.4 If English is not your native language, how many years have you (approximately) spoken English?

1.5 What is your current degree level?

- Undergraduate
- Master's
- PhD
- Postdoc
- Other:

1.6 What is your degree program title or specialisation? (e.g., Artificial Intelligence, Computer Security, ...)

1.7 How often do the following happen?

	Never	Rarely	Sometimes	Often	Always
I ask other people for help with computers	<input type="checkbox"/>				
Other people ask me for help with computers	<input type="checkbox"/>				
I ask other people for help with privacy problems (e.g. how to not disclose too much personal identifiable information on Social Media)	<input type="checkbox"/>				
Other people ask me for help with privacy problems (e.g. how to not disclose too much personal identifiable information on Social Media)	<input type="checkbox"/>				

1.8 For each of the following statements, how strongly do you agree or disagree?

	Strongly disagree	Disagree	Neither agree or disagree	Agree	Strongly agree
Consumers have lost all control over how personal information is collected and used	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Most businesses handle the personal information they collect about consumers in a proper and confidential way	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Existing laws and organisational practices provide a reasonable level of protection for consumer privacy today	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2 Social Media Usage

2.1 How often do you use each of the following social media platforms?

	Do not use	Rarely	Monthly	Weekly	Daily	Multiple times/ day
Twitter	<input type="checkbox"/>					
Facebook	<input type="checkbox"/>					
LinkedIn	<input type="checkbox"/>					
FourSquare	<input type="checkbox"/>					
YouTube	<input type="checkbox"/>					
QQ	<input type="checkbox"/>					
Reddit	<input type="checkbox"/>					
Piazza	<input type="checkbox"/>					

2.2 Please list other social media platforms you use at least once a month

2.3 How often do you post similar content or links to multiple social media accounts? For example, posting the same news article, or the same picture to both Twitter and Facebook.

Never	Rarely	Sometimes	Often	Always
<input type="checkbox"/>				

2.4 If you use Twitter, what information does your public profile or description include?

- | | | |
|--|--|--|
| <input type="checkbox"/> Email | <input type="checkbox"/> Real First Name | <input type="checkbox"/> Real Last Name |
| <input type="checkbox"/> Pictures | <input type="checkbox"/> Videos | <input type="checkbox"/> Current Location |
| <input type="checkbox"/> Hometown | <input type="checkbox"/> Current town | <input type="checkbox"/> Occupation |
| <input type="checkbox"/> Date of Birth | <input type="checkbox"/> Nationality | <input type="checkbox"/> Phone Number |
| <input type="checkbox"/> Other Social Media accounts | Others: | <div style="border: 1px solid black; width: 150px; height: 20px; float: right;"></div> |

2.5 Do you have any social media accounts that you try to keep separate from your real name?

- Yes
- No



3420821876 0002

2.6 If your answer to the previous question was yes, on which platform(s) do you try that?

- | | | |
|-------------------------------------|-----------------------------------|-----------------------------------|
| <input type="checkbox"/> Twitter | <input type="checkbox"/> Facebook | <input type="checkbox"/> LinkedIn |
| <input type="checkbox"/> FourSquare | <input type="checkbox"/> YouTube | <input type="checkbox"/> QQ |
| <input type="checkbox"/> Reddit | <input type="checkbox"/> Piazza | |

Other:

2.7 Are there platforms on which you have more than one account? (e.g. two Facebook accounts)

- Yes
 No

2.8 If your answer to the previous question was yes, on which platform(s) do you have that?

- | | | |
|-------------------------------------|-----------------------------------|-----------------------------------|
| <input type="checkbox"/> Twitter | <input type="checkbox"/> Facebook | <input type="checkbox"/> LinkedIn |
| <input type="checkbox"/> FourSquare | <input type="checkbox"/> YouTube | <input type="checkbox"/> QQ |
| <input type="checkbox"/> Reddit | <input type="checkbox"/> Piazza | |

Other:

2.9 For each of the following social media platforms that you use, how often do you knowingly disclose information about your location (e.g. checking-in/geotagging Tweets/openly saying where you are), or information that could be used to work out your location?

	Do not use	Never	Rarely	Monthly	Weekly	Daily	Multiple times/day
Twitter	<input type="checkbox"/>						
Facebook	<input type="checkbox"/>						
LinkedIn	<input type="checkbox"/>						
FourSquare	<input type="checkbox"/>						
YouTube	<input type="checkbox"/>						
QQ	<input type="checkbox"/>						

2.10 How concerned would you be if your location was identified to the following levels of accuracy using information you have purposely shared?

	Not at all concerned	Slightly concerned	Somewhat concerned	Moderately concerned	Extremely concerned
Country level	<input type="checkbox"/>				
City level	<input type="checkbox"/>				
Town/village level	<input type="checkbox"/>				
Address level	<input type="checkbox"/>				

'Extracting Personal Information from Twitter User Accounts' Consent Form

We are conducting projects aimed at finding privacy issues regarding personal information on Twitter, and giving feedback to users.

Today we will be running a focus group session with you to discuss possible ideas and issues you have regarding privacy and social media use. A questionnaire will be used to gather background information about the group. During the focus group, we will also present you with a mock-up of our designs and seek some feedback on it.

We will be audio recording the discussion. If you feel uncomfortable about this at any time, you may ask us to stop the recording at any time or tell us that the next bit should not be quoted.

Recordings will be used to inform us about the design requirements and usability aspects of our systems. The audio and transcripts will be kept for a maximum of one year and then destroyed. Anonymized quotes or short audio clips may be retained longer for use by future students on this project.

Our projects are co-supervised by Dr Kami Vaniea (kvaniea@inf.ed.ac.uk) and Liane Guillou (liane.guillou@gmail.com), conducted by ourselves, Alexander Caughey (s1328266@sms.ed.ac.uk) and Timo Mulder (s1624905@sms.ed.ac.uk)

[] I understand that I am participating in a study as part of the "Extracting Personal Information from Twitter User Accounts" project.

[] I am willing for the audio to be digitally recorded and transcribed for the use as part of the research project

[] The researcher may use **audio/ literal quotes** from the session/ questionnaire in publications provided that the quote is anonymized and cannot be connected back to me.

Participant: _____

Date: _____

Researchers: Alexander Caughey and Timo Mulder

Date: _____

Contact details:

Alexander Caughey, email address: s1328266@sms.ed.ac.uk

Timo Mulder, email address: s1624905@sms.ed.ac.uk

Ethical Review Procedures: Level 1

Project Details & Self-assessment

This document is closely modelled on documents used in School of Philosophy, Psychology and Language Sciences provided by Ellen Bard and Cedric MacMartin.

This form is to be filled in and submitted at the same time as the project proposal or the funding application it applies to. The form should be submitted by the Principal Investigator, except in the following cases:

- Post-doctoral fellowships – the proposed postdoc mentor.
- UG, MSc, and PhD research projects – the supervisor.
- Visiting researcher – the staff hosting the visitor.

Please submit the completed form by email to: infkm+ethics@inf.ed.ac.uk

This address, with appropriate RT number once issued, should be used for all correspondence (including forms and attached documents). This is essential to ensure proper record keeping. No signature is required if the form is sent from a valid University email address.

Project Details

1 Type Of Project:

- | | | |
|---|--|--|
| <input type="checkbox"/> Research grant proposal | <input type="checkbox"/> UG final year project | <input checked="" type="checkbox"/> MSc project |
| <input type="checkbox"/> Post-doctoral fellowship | <input type="checkbox"/> PhD project | <input type="checkbox"/> Research performed by visiting researcher |
| <input type="checkbox"/> Personal research | <input type="checkbox"/> Other: _____ | |

2 Is there a sponsor/ funding body?

YES / NO *8*

3 Does the sponsor/funder require formal prior ethical review?

YES / NO *8*

If yes, by what date is a response required ?

4 Is any other institution and/or ethics committee involved?

YES / NO

If YES, give details and indicate the status of the application at each other institution or ethics committee (i.e., submitted, approved, deferred, rejected):

- 5 Title of Project *Extracting personal information from Twitter user accounts*
- 6 Researchers' names, affiliations, emails *Timo Mulder (S1624905@sus.ed.ac.uk)*
Alexander Caughey (S1328266@sus.ed.ac.uk)
 Include student/supervisor, post-doc/mentor, PI, or visitor/host.
- 7 State which professional organisation guidelines you are using:
- School of Informatics research ethics code: <http://www.inf.ed.ac.uk/research/ethics/>
- Other ethics code as required by funding body or professional organization:
 Title: _____ URL: _____

Self-assessment

Refer to Level 2 form for details on any of the following points.

1. Protection of research participants' confidentiality

Are there any issues of CONFIDENTIALITY which are NOT ADEQUATELY HANDLED by normal tenets of academic confidentiality? YES / NO

These include well-established sets of procedures that may be agreed more or less explicitly with collaborating individuals/organisations, for example, regarding:

- (a) Non-attribution of individual responses;
- (b) Individuals and organisations anonymised in publications and presentation;
- (c) Specific agreement with respondents regarding feedback to collaborators and publication.

2. Data protection and consent

Are there any issues of DATA HANDLING AND CONSENT which are NOT ADEQUATELY DEALT WITH, and compliant with established procedures? YES / NO

These include well-established sets of procedures, for example regarding:

- (a) Compliance with the University of Edinburgh's Data Protection procedures (see <http://www.recordsmanagement.ed.ac.uk>);
- (b) Respondents giving consent regarding the collection of personal data (via consent form).

3. Significant potential for physical or psychological harm, discomfort or stress

Are there any risks of :

- (a) psychological harm or stress for the participants?
- (b) physical harm or discomfort for the participants?
- (c) any kind to the researcher?

YES / NO

YES / NO

YES / NO

4. Vulnerable participants

Are any of the participants in the research vulnerable, e.g., children, patients, disabled participants? YES / NO

5. Moral issues and researcher/institutional conflicts of interest

Are there any SPECIAL MORAL ISSUES/CONFLICTS OF INTEREST? These include:

- (a) Conflict of interest: potential benefit to the researcher, friends or family of a particular research outcome which might compromise the researcher's objectivity or independence;
- (b) The need to keep the purposes of research concealed;
- (c) Use of participants who are unable to provide informed consent (e.g., children);
- (d) Situations where research findings would impinge negatively/differentially upon the interests of participants.

YES / NO

6. Bringing the University into disrepute

Is there any aspect of the proposed research which might bring the University into disrepute? For example, could any aspect of the research be considered controversial or prejudiced? YES / NO

7. Use of animals

Does the research involve animals?

YES / NO

8. Developing countries

Does the research involve developing countries?

YES / NO

9. Dual use

Is the research classified or does it have specific adversarial military applications?

YES / **10. Terrorist or extremist groups**Does your research concern groups which may be construed as terrorist or extremist? YES / **Can you stop now?**

You may want to assure yourself that your 'NO' answers are correct by checking the detailed form in the next section.

If all the YES / NO answers are NO, the self assessment has been conducted and confirms the ABSENCE OF REASONABLY FORESEEABLE ETHICAL RISKS. This form should be signed by the researchers and submitted. The researchers may retain a copy for their own records.

If any answer is YES, please complete the relevant section in the Level 2 form below.

26/06/2017
ACoyle

26/06/2017
ACoyle