

Privacy Management Tool For Twitter

Mingrui Cai

Master of Science
Computer Science
School of Informatics
University of Edinburgh
2021

Abstract

The issue of personal privacy breaches is gaining increasing attention in the information age. With the rise of social media, the efficient protection of personal privacy has become a study topic. Although social media such as Twitter have privacy safeguards and settings in place, they are still unable to detect new content created and shared in order to prevent potential privacy breaches in advance of high-risk behaviour. This project will provide a solution that intelligently detects the possible privacy risk from the new content and remind users on Twitter. The application can extract the information from the user's previous tweets, generate a privacy model and check the new content via the model. This dissertation will fully discuss the design, implementation and evaluation of the privacy management tool.

Acknowledgements

This Master of Science project is a component of the degree of computer science at the School of Informatics, University of Edinburgh.

Firstly, I would like to express my gratitude to my supervisor Dr Nadin Kokciyan for her patient guidance and support. Under her supervision and encouragement, I overcame many difficulties in the project and gained a lot. At the same time, her professional guidance helped me to solve many problems especially in ethics application and implementation. Without her help this project could not have been completed.

Also, I would like to appreciate the School of Informatics and the University of Edinburgh which provide opportunities for me to engage in useful courses and academic resources under such a pandemic environment.

I would like to thank my parents and my girlfriend for their love and support during my study. Thanks to all my friends who gave me suggestions about the project and engaged with the user study.

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.

(Mingrui Cai)

Table of Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problem Specification	2
1.3	Goals	3
1.4	Feasibility	3
1.5	Dissertation Structure	4
2	Background	5
2.1	Privacy Definition	5
2.2	Privacy in Social Media	5
2.3	Topic Models	6
3	Methodology	9
3.1	Privacy Model	9
3.2	Project Management	10
3.3	Ethics	11
4	Design	13
4.1	Requirement Analysis	13
4.2	User Interface Design	14
4.2.1	Workflow Design	15
4.2.2	Mockup Design	16
4.3	Twitter App Design	18
5	Implementation	23
5.1	User Interface	23
5.2	Twitter App	24
5.2.1	Data Collection	24

5.2.2	Privacy Model Building	26
5.2.3	Working Process	29
6	Experiment and Evaluation	32
6.1	Experiment Environment	32
6.2	Parameter Experiment	33
6.2.1	Hashtag Experiment	33
6.2.2	Topic Words Experiment	33
6.2.3	Similarity Experiment	34
6.3	User Study	35
6.3.1	Participant Information Survey	35
6.3.2	User Feedback	35
7	Conclusions	39
7.1	Limitations	39
7.2	Future Work	40
	Bibliography	41
A	Twitter Developer Application Email	45
B	Participant Information Sheet	49
C	Participant Survey Sheet	54

Chapter 1

Introduction

1.1 Motivation

Personal privacy has always been a sensitive and controversial subject. Since the advent of modern society, moral standards have gradually risen and privacy has become embedded in all aspects of life, from our bodies to our personal information [1]. The point of contention on privacy issues is that different people have different perceptions of privacy issues [2]. For each individual, their own definition of privacy differs. For example, some people consider age to be private, while others do not. This difference in perception depends on the family environment, the living environment and even the cultural environment [3]. Although there are privacy laws in place in modern society that govern privacy violations, the privacy violations defined in these laws are serious, and in most cases, privacy protection lies more in the awareness of privacy protection for all.

With the spread of the Internet, mankind has entered the information age and the number of online Internet users has exploded in the last decade. In the beginning, the Internet was a virtual world relatively independent of the real world, where people used screen names to represent their identities, without class distinctions or status distinctions, everyone was an equal and free individual in this new virtual world [4]. This isolation did not last long, and with the development of internet services, the online business eventually came to serve real life. As the real world invades the virtual world, personal information also penetrates from the real world to the virtual world, due to the fact that many Internet services require contact details of customers [5]. Since the Internet is not completely secure and can even be dangerous, the transmission of personal data packaged in data streams over the Internet can lead to data leakage problems,

and these data possibly contain personal privacy information [6][7]. The disclosure of personal information or personal privacy can lead to serious problems, the most immediate of which is the psychological damage to the person concerned, who may feel uncomfortable about having this personal information known to strangers. At the same time, the leakage of personal information may lead to frequent harassment and fraudulent calls as well as more deep-rooted social problems that will not be discussed in detail here [8][9].

In the process of integrating the Internet and real life, people hope that the Internet can become a platform to carry real life and thus serve real life. In this context, the Internet has become an excellent platform for people to communicate with each other. This has given rise to a number of communication platforms and software on the Internet. With the growing number of people with this need and the development of big data technology, people are no longer satisfied with just point-to-point communication and intra-group communication is becoming more convenient and faster. This led to the emergence of social media like Facebook and Twitter [10], which are able to organise topics of interest through advanced big data technology and group people with the same interests into groups [11]. Furthermore, the social media platforms can generate hot topics and gather related contents which are contributed by users under the topic [12].

Since this process involves users posting content on their own, there are inevitably privacy issues involved. Social platforms often encourage users to enter real personal information, such as email addresses and mobile phone numbers, in order to make their profiles more specific and analyse their preferences for targeted topics or in-site advertising. Such measures do improve the user experience to a certain extent, allowing them to receive a lot of content of interest to them, but the protection of user information is left entirely to the platform. A breach or loss of platform data could be catastrophic for these individual users but the security measures on these big platforms are generally better and it is difficult to have a massive data breach. So the risk of privacy breaches is mainly focused on what users post.

1.2 Problem Specification

In an environment where social platforms are relatively well protected for our privacy, the problem of privacy breaches is mainly focused on content created and shared by users. Users would spontaneously comment or express their opinions on social media

content after viewing them, and they also share their daily life and feelings on social media sites. They are able to create and share content that they find meaningful. They may also share content that involves other people (i.e. with @ before other people's screen name) or popular topics (i.e. with # before the hashtag topics).

However, the challenge here is that none of the major social media platforms currently offer an embedded sensitive information detection service. In other words, when a user wants to post a new piece of content, the system just passes it anyway regardless of whether it contains information that may be personal. The user themselves may have unwittingly created the content but it turns out to contain sensitive personal information that they do not wish to be published. Users need services to help them determine if the new content they create contains personal privacy information.

1.3 Goals

The main goal of the project is design and develop a tool to provide users with a detection service that will be enabled when they create new content on social platforms, helping them to detect whether the content contains sensitive information.

Due to the users' behaviour on Twitter are public by default meaning that their new tweet will be considered as public if they do not specifically set permissions, Twitter is more prone to privacy breaches so that our project will focus on privacy detection of tweets. Another reason for selecting Twitter as our main focus is that it have a wide range of application scenarios. Twitter has a great number of hot topics or trends every day and users can freely post new tweets which are visible to the public or the followers. This special binary visibility mechanism forces users to be careful about what they post, because on Twitter they cannot be visible only to themselves or only to their friends, in a sense their tweets are for everyone to see and the only difference is if the audience are followers.

In short, the final goal is to design and implement an intelligent privacy management tool for Twitter. The details of design and implementation will be fully discussed in the corresponding chapters.

1.4 Feasibility

The whole project follows the principles of agile development, and the design and implementation process were adjusted according to the actual situation in order to

achieve the goal of improving the completion of the project. After several feasibility discussions, we decided to split the plan for the all-in-one plugin into user interface and standalone application due to the time limit. The advantage of this is that we can achieve full functionality and provide a usable user interface, while saving time by avoiding the tricky problems that may be encountered when integrating front and back ends.

1.5 Dissertation Structure

The dissertation is divided into 7 chapters. The introduction chapter briefly introduced the origin of the project and provided a final goal of the project.

The background section will introduce the theoretical background knowledge involved in this project, which helps us to gain a deeper understanding of the design and implementation methodology.

Methodology will discuss the methods used in this project from a high level view.

Implementation part will fully presents the details of implementation for each part of the application and the final working process.

In the experiment and evaluation, we will discuss about user study approved by ethics team and evaluate the result.

The final part will conclude the findings and achievements of the whole project and discuss the limitations and future work.

Chapter 2

Background

2.1 Privacy Definition

Before designing a privacy management tool, we first need to clarify what privacy is and, more specifically, what it is on social media [13]. In the subjective view of the average person, one's privacy is the information about oneself that one would feel uncomfortable if the information known by others. From an objective point of view, privacy is recognized or legally enforced personal information that the person concerned does not wish to disclose, which may include various aspects such as physical and psychological information, etc. To make this concept more understandable, I would like to give an example here. When you make a friend online but you do not want to reveal your height, weight or even your looks because you are worried that your relationship will be affected by these appearance factors. You are totally free to keep this information because this information is personal privacy for you. However, for some other people, the body indicators such as height and weight are not their privacy because they do not care about if they will affect their life.

2.2 Privacy in Social Media

The above explanation of privacy makes sense on social media as well. Each user has a different definition of their own private information, resulting in privacy rules that cannot be measured on a uniform scale for each individual [14]. In brief, the most violation of privacy on social media is revealing unwanted sensitive information to the audience or even the public [15].

The general privacy leakage behaviour is related to access control meaning that

if the user can access the content from others [16]. This can be solved via semantic approaches proposed by R.Fogues [17].

Furthermore, there are two major ways to handle the privacy problem: prevention of privacy leakage [18] and detection of privacy leakage [19]. The former method is before the privacy breach and the latter one is during the privacy breach. In this project, we will adopt the prevention method to block the high-risk tweet. The reason is that the privacy model and user profile are always changing through time so that monitoring privacy behaviour in real-time is very difficult. Instead, stopping high-risk behaviour before it happens is clearly a better and more efficient solution.

There are more specific studies about privacy in social media. Thanks to big data technology, the personal profiles of certain groups of users can be searched through vast amounts of publicly available information [20]. A recommendation system with this kind of functionality can be a great help for the platform to push ads and content to groups with different interests. More specifically, a user's political preferences can also be detected through the content they have previously posted [21]. Recent research has shown that user privacy violations can be automatically detected and that this technology can further generate user privacy requirements only via the public information users provided before [19]. There are also some studies focus on finding the people who are likely to reveal their privacy. These high-risk groups can be targeted by graph-based and topic models [22]. An important metric in this model is the privacy score which can efficiently find the target group of users [23]. Furthermore, some methods such as the privacy wizard presented by Fang and LeFevre can provide suggestions about configuring the privacy settings for users [24].

2.3 Topic Models

In order to find out whether the content created and shared contains privacy information, we first need to know what the general subject matter of the content is so that we can better filter and determine sensitive information. In the text technology area, we usually define a text made up of a number of words as a corpus and a set of the corpus as a document. In a more understandable way, a corpus is a sentence and a document is a collection of sentences.

The first step of topic level analysis is dimensionality reduction transformation. The number of words would be transformed to the number of topics so that the dimension of raw data can be reduced (shown in Figure 2.1). This step is a preprocessing

operation to better fit the data into the topic model. Then, the topic modelling methods read the transformed data and output document-topic matrix and topic-word matrix [25]. In practice, these matrices can be displayed in a human-readable mode by calling encapsulated methods which will be discussed in the implementation part.

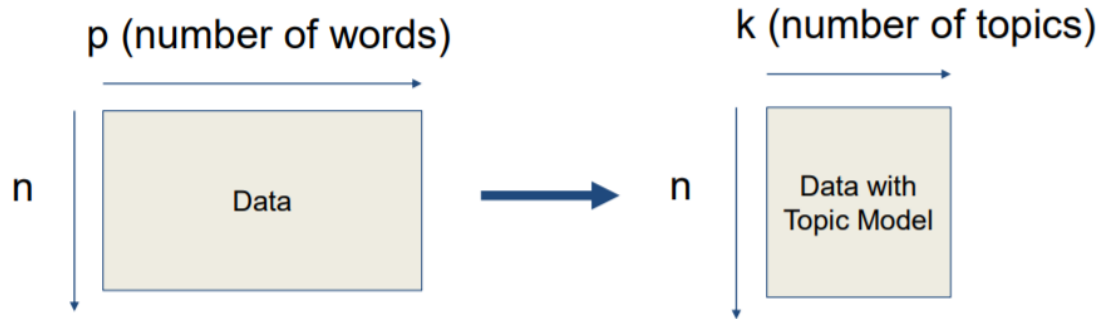


Figure 2.1: Dimensionality Reduction [26]

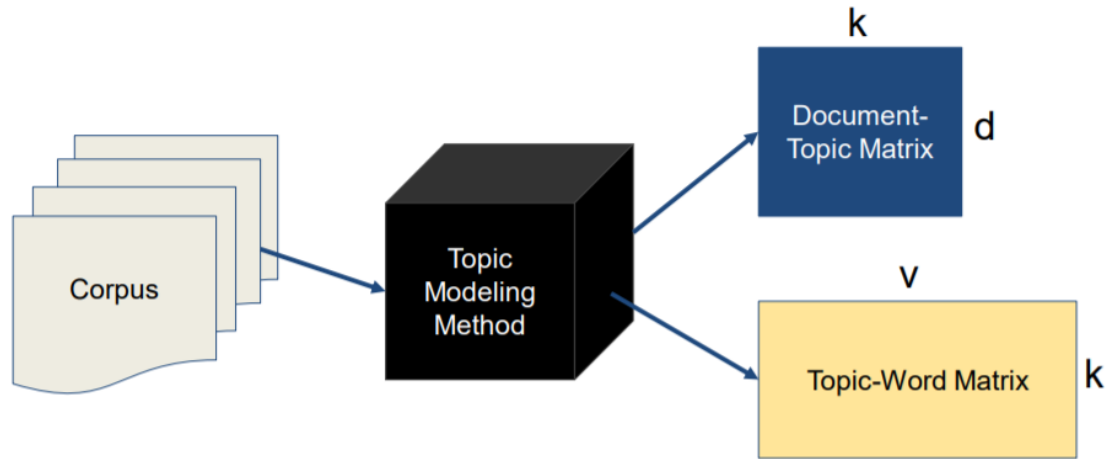


Figure 2.2: Topic Modelling Process [26]

One of the vector space model algorithms used in the project is Latent Dirichlet Allocation shortened as LDA, which is the most popular topic modelling method for text data so far [27]. LDA is a transformation from bag-of-words (i.e. a matrix contains words and their frequency) to topic space, and the topics generated by LDA can be considered as the probability distributions over words [28].

Another useful model is Latent Semantic Indexing shortened as LSI. As the focus of this project is on the development of the privacy management tool rather than on the underlying principles of certain natural language processing and machine learning models, these models will not be discussed in depth here. LSI can be understood simply as a model for transforming vectors (i.e. parts of the bag-of-words matrix) from

one space to another [29]. Each vector represents a tweet in this case, and all the vectors form the bag-of-words matrix. An important reason for choosing LSI is its ability to detect the similarity between the new text and each original corpus in the document. In other words, LSI model can measure the similarity between new tweets with each previous tweet and compute a score for each pair which represents the similarity [30]. The advantage of LSI is that it can be trained continuously, with only new data needed for each training session, and it can recognize patterns and relationships between words and topics [31].

Chapter 3

Methodology

This chapter shows the methods and approaches we used to push the project from a high level and abstract aspect, the details of the project will be fully discussed in the following chapters. Due to the project focus on development and implementation instead of theoretical research, this part will not be overly complex.

3.1 Privacy Model

In order to detect the potential privacy violation, a privacy model should be built first. The idea of building a privacy model was inspired by Nadin’s work [19] noting that the privacy model can denote the violation of privacy. The aim of building this model is to narrow down privacy from a big and abstract topic to a concrete and realizable solution. In this project, the privacy model is basically denoted by the words and tokens from the “word bank”.

The first part of the privacy model is the place and hashtag information. This basic part is based on the two directions we proposed in the project proposal: privacy topic protection and location information protection. This mechanism is similar to whitelisting [32] in the sense that the system stores some trusted information in advance so that the next time a user uses similar information, it will not be considered by the system as a high-risk behaviour. The result is that every time a user posts something similar to what already exists in the privacy model, the system will automatically pass it, and conversely, if a user posts something that does not exist in the privacy model, the system will determine that it is a high-risk behaviour and alert the user.

The second part of the privacy model is the Twitter app generated topics by LDA (Latent Dirichlet Allocation) model [33]. This kind of machine learning model can

generate several groups of words with the score of each word so that those topics can denote the more frequent and popular topics in the data and the corresponding keywords to some extent. The program can furthermore generate a keywords pool with the word frequency so that we can have a topic model and this topic model can be used to detect the high-risk behaviour.

The final part is the similarity measurement by LSI (Latent Semantic Analysis) model [34]. This model uses a similar data transformation method to that in LDA. More details will be discussed in the implementation chapter and the high-level aim of similarity measurement is to output the similarity score between new tweets and each existing tweet in the previous data. The final result outputs a rank from the highest score to the lowest and we can decide if the new tweet is similar to the previous tweets.

To sum up, the privacy model consists of extracted tokens (place and hashtag), potential topics (LDA) and similarity measurement (LSI). The new tweet will be put into these stages for analysis and the program will output suggestions about whether the new tweet is fine to be posted. For external users, this privacy model is well encapsulated and stores the frequently published information with the permission of the user. To simplify the abstract concept of the privacy model, the output of the privacy model can be simply considered as binary meaning that the system will suggest users post new tweets or not based on their previous tweets.

3.2 Project Management

The most important way in project management is weekly meetings. We organized weekly meetings from early June to mid-August. We discussed the progress and problems each week and had possible solutions for most of the issues from the design phase to the evaluation phase. Also, we spent much time on the ethics application since the project involves data collection from social media and individual participants.

Another useful approach is Notion¹ (i.e. a useful project management tool). This intelligent tool helped me to set up a relative comprehensive plan about when to do what. I can also add priority and status for each task and furthermore add some description and comments. The notes for most of the weekly meetings were also added to the Notion page including some ideas and solution descriptions. This tool helped me to have a clear idea in my mind to manage and progress the project and also provides an intuitive visual representation of the current status of the project to the supervisor.

¹The Notion page of this project is here: [Link to the Notion page of the project](#)

The screenshot of the current status of the Notion page is shown below.

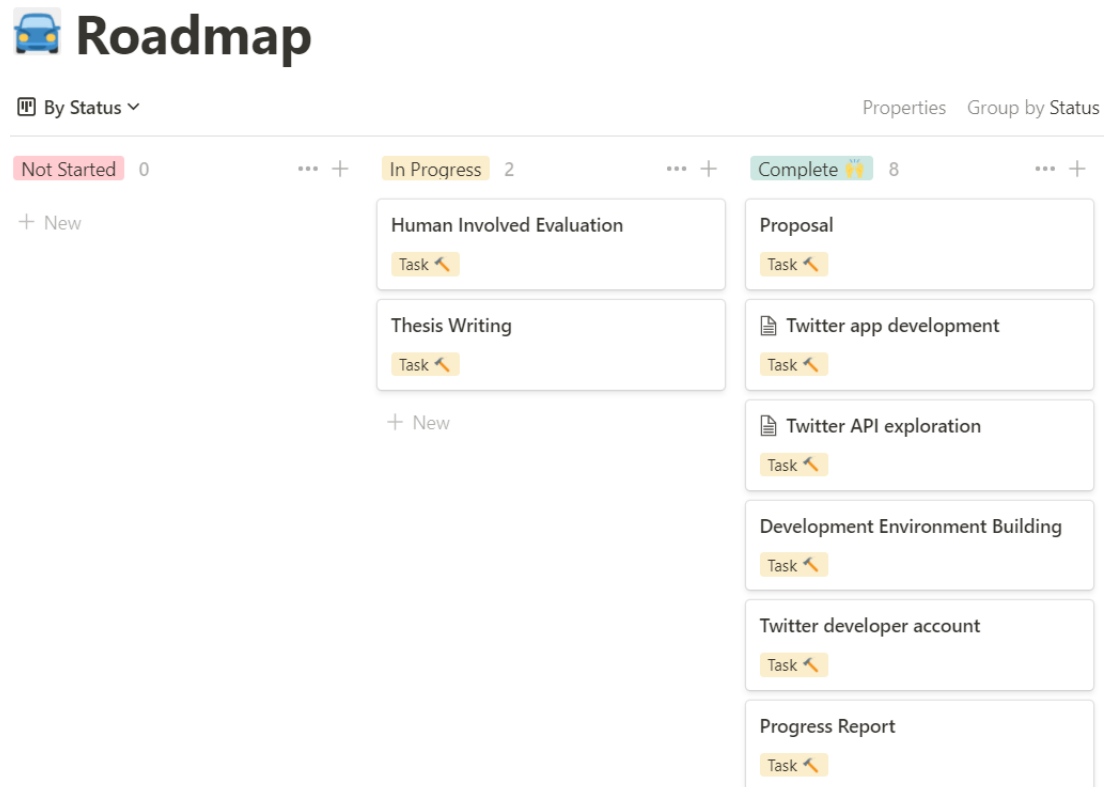


Figure 3.1: Notion page screenshot

The project management basically followed the plan in the proposal but somehow adjust it according to the real situation. Those changes were discussed in the weekly meeting with the supervisor and were good for the completion of the project.

3.3 Ethics

In order to progress the project, we need to collect data from Twitter. The data collection process must follow the terms and conditions of Twitter and GDPR. I applied for a new Twitter account only for this project and applied to transfer the account to a developer account. Appendix A includes more details about the information that the Twitter team asked me to provide.

Also, the ethics application contains similar information and declare that the data will only be stored locally. We have reorganised and answered a few questions from the ethics team. For collecting data from random public accounts, we will not display their raw data but only the summarized result in our analysis. For participant evaluation, we will not display their account name and raw data but anonymise the summarized result

of the experiment. These measures are effective in preventing potential damage to users from data breaches, and the ethics application have been successfully granted based on these explanations. The final participant information sheet can be seen in Appendix B.

Chapter 4

Design

This chapter shows the design process of the project including requirement analysis, user interface design and Twitter app design. The design starts from specifying requirements to designing an application to meet those requirements. The following design scheme is the final solution after several iterations.

4.1 Requirement Analysis

The requirement analysis is based on the problem of the project. The first step is to define use cases or scenarios.

For normal Twitter users, they may use their mobile phones or computers to browse Twitter. If they see something they are really interested in, they will comment on that tweet or post a new tweet by themselves. Or they just want to post some new tweet to describe their situation including record their daily life and express their feelings. Their tweets may contain information that is sensitive to them, such as political topics or location-related information. Some people do not care about these possible privacy breaches but there are also some people who take their privacy protection very seriously.

Their functional requirement is straightforward: to find an intelligent way to detect the presence of private information in new tweets and to passively accept the notification. The non-functional requirement can be an app on mobile phone devices or a browser extension on a computer.

For mobile devices, it is not easy to develop a third-party app and make it possible to access the Twitter app API in the background, because the background management mechanism of both IOS and Android is complicated, and it is too much of a hassle to

switch between apps on mobile if it is not a function that comes with Twitter. Although there is now a split-screen function on mobile, problems with background management such as the system's mechanism for killing the background because of insufficient voluntary memory will still have a greater impact on the stability and availability of the software. For example, a mobile phone user needs to open both the Twitter app and our tool app but he may find that the system will automatically kill the tool app in the background because the system does not have enough RAM (random access memory) to run both apps. Even if the system can run both apps at the same time, whether the two apps can interact with data in the background is still a serious problem. In summary, this issue relates to whether the Twitter API can allow access to personal data in the background on mobile, and the management of background application permissions and processes by different systems on mobile devices.

In order to solve the problem, an intelligent tool that runs in the browser can help users to detect the potential sensitive contents and alert them. Despite the portability and convenience of mobile devices, computers are still the main productivity tool. Users using Twitter on a computer will generally enter Twitter from the browser when most of the content on the entire display area of the screen is Twitter-related content, and opening a small plug-in at this time will not affect the user's perception of Twitter much, and the tool will not steal the thunder of Twitter as it does in the mobile app. The idea is that on mobile each app is independent of the other and each app requires the user to tap an icon on the main screen to access it, which makes each app feel the same importance to the user, whereas in the browser, the plug-in exists as a tool and does not take up space for the user to browse the web. In short, the tool in mobile will affect the usability of Twitter but the plugin will not affect much. The requirement is clear so far: designing a browser plugin that can detect if there are privacy sensitive information in the text. The privacy-sensitive information is defined by the privacy model, the program will pass the text if it is in the model otherwise block it.

4.2 User Interface Design

Once the design requirements have been defined, the next step is to design the mockup to have a rough simulation of the product functionality. Before that, a flowchart describing users' behaviour when they use the plugin should be designed.

4.2.1 Workflow Design

The basic user scenarios start from opening the browser. The user should first open the browser (Chrome in this case) and go to the Twitter page (www.twitter.com). Once the user logs in to their Twitter account by their own account name and password, they can freely browse the content in Twitter. If they would like to post some new tweet whether they comment on someone else's tweet or post their own feelings and thoughts, they may post some information that involves their privacy. At this time, they can launch the tool by clicking on the Twitter icon in the top right corner of their browser in the plug-in bar. They are required to input their Twitter account name which with @ as prefix and input the tweet that they would like to post. After they click the submit button, the program analyzes their new tweet for the risk of containing sensitive privacy information based on our privacy model. The plugin then output the alert information saying that it is fine to post the tweet or not. The flowchart is shown in Figure 4.1.

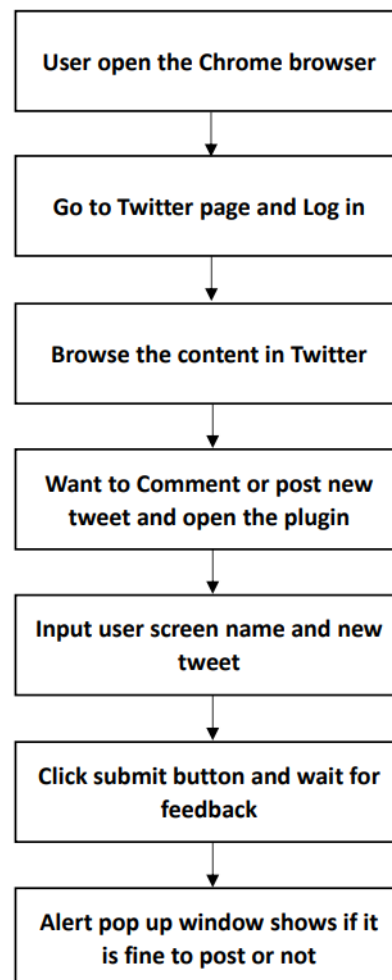


Figure 4.1: Design Flowchart

4.2.2 Mockup Design

According to the flowchart, the mockup was designed on a prototype design platform Balsamiq ¹ which was used several times in the previous HCI courses. It is worth mentioning that the Balsamiq project space ² is only valid for 30 days for free individual users. In order to clearly present the design mockup, the following parts will discuss each step of the mockup in detail.

Figure 4.2 shows the initial screen when a user first enters the Twitter site. The plug-in bar is on the top right, to the right of the web address bar and to the left of the search bar, and the Twitter bird is the plug-in icon.

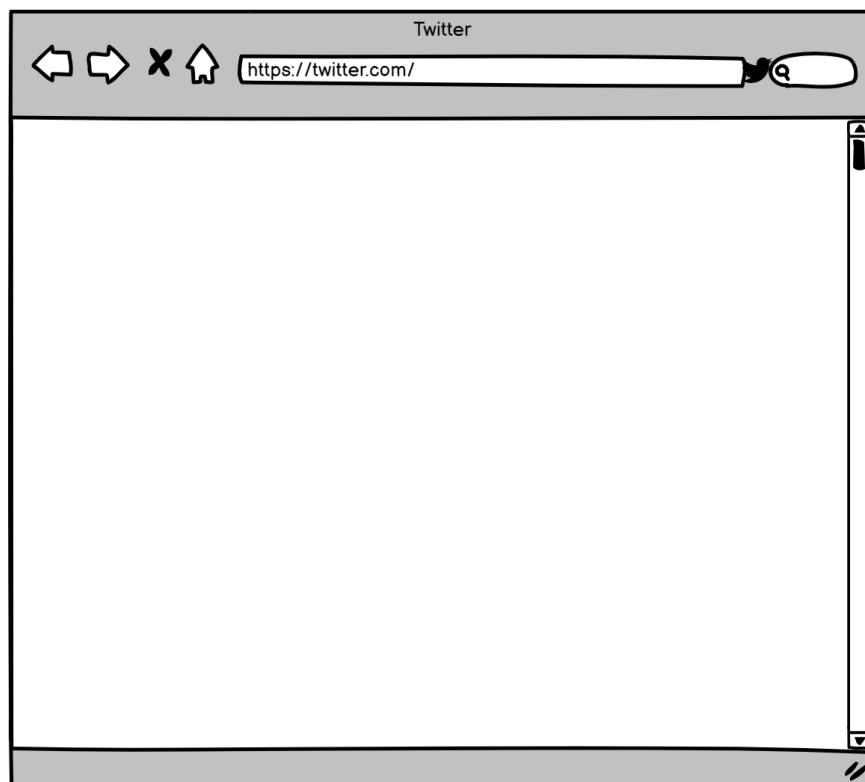


Figure 4.2: Twitter Page

When the user clicks the bird icon, the plugin tool is launched. Once the plugin has been initialised, the page is shown in Figure 4.3. The title of the graphic user interface indicates what the tool can do that is managing privacy. Although it does not directly manage privacy settings, it can go a long way towards helping users manage high-risk behaviours involving sensitive personal information in a timely manner so that we call it a “management tool”. This page also provides enough tips for users to input their

¹Balsamiq website: <https://balsamiq.cloud>

²The Balsamiq project link is here: <https://balsamiq.cloud/ssogf/pkuy01c>

information and the notice at the bottom tells users the function of this plugin. These are designed to show the user what the software does and how it works, so that even non-expert users who do not use a computer browser very often can easily understand how to use the plugin, thus improving the usability and ease of use of the software. This design concept follows Neilson's 10 heuristics rules [35] that the system helps users to use the software and maintain the visibility of the system status.

The diagram shows a web browser window with the title 'Twitter'. The address bar contains 'https://twitter.com/'. A search icon is visible in the top right corner of the browser. Overlaid on the browser is a 'Twitter Privacy Management Tool' window. This tool contains the following elements:

- A label: 'Twitter screen name (with @ as prefix):'
- An input field containing the placeholder text 'your screen name'.
- A label: 'Potential tweet:'
- A larger input field containing the placeholder text 'Please input a new tweet that you want to post here.'
- A 'Submit' button.
- A notice at the bottom: 'If you click submit, we will analyze your potential tweet to detect if there are any privacy risks and give you feedback.'

Figure 4.3: Plugin Initiation

After the user input screen name and new tweet then click the submit button, the program automatically analyzes the new tweet with the privacy model consisting of previous tweets and provides feedback. At this point, there are two different scenarios based on the feedback from the analysis.

The first case denoted in Figure 4.4 meaning that the new tweet may contain some sensitive privacy information such as place names or hashtag topics that did not appear in the previous tweets data. For this case, the system strongly suggests not to post this tweet and another alert window pops up (shown in figure 4.5) after the user clicks the 'No' button meaning that they do not want to post this tweet because of the risk of the privacy breach. The system praises the user saying that "good choice" since the system and the user work together to avoid possible privacy breaches. At this point, users can

just click 'OK' and back to the main page for starting a new tweet analysis.

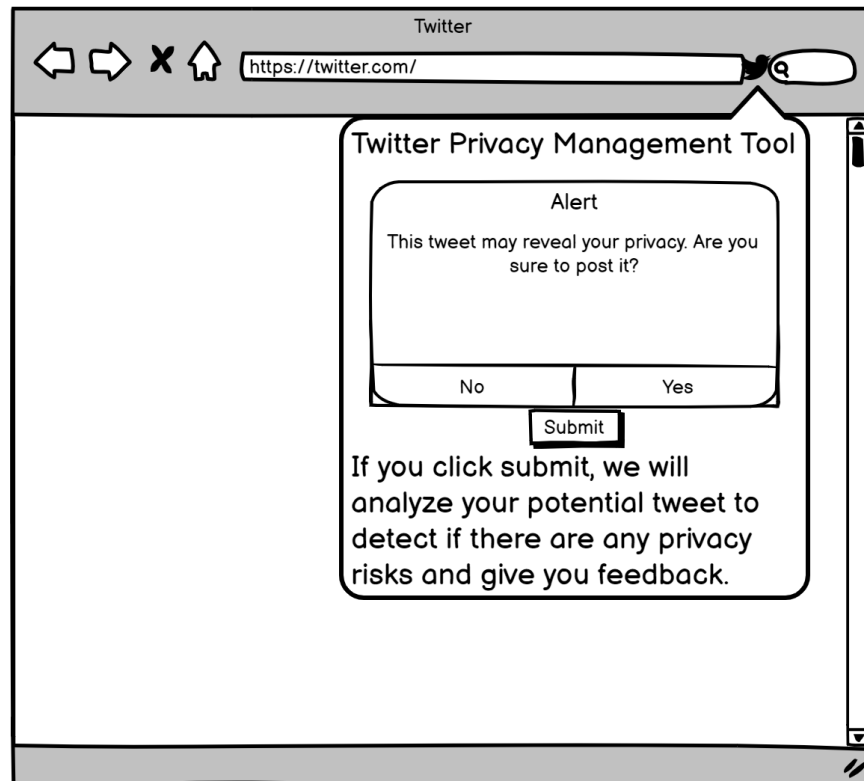


Figure 4.4: Clicked Submit

Another situation is that the new tweet does not contain any information that may leakage privacy information (shown in Figure 4.6). The alert window shows the information and users can click 'OK' to post it and back to the main page for analyzing another new tweet.

4.3 Twitter App Design

The Twitter app design is based on the privacy model and user interface design. The general phases for the program consist of collecting previous tweets data according to the screen name that users provided, building the privacy model, read the new tweet from the user and analyze the tweet based on the privacy model.

In the data collection phase, the program reads the screen name and get this user's timeline (i.e. a series of recent tweets posted by this user) from Twitter via Twitter API. The timeline from Twitter data fetched from the tweeters is not plain text, so we need to preprocess the data using some natural language processing technology. The location information and hashtag topics can be directly pulled out from the plain text

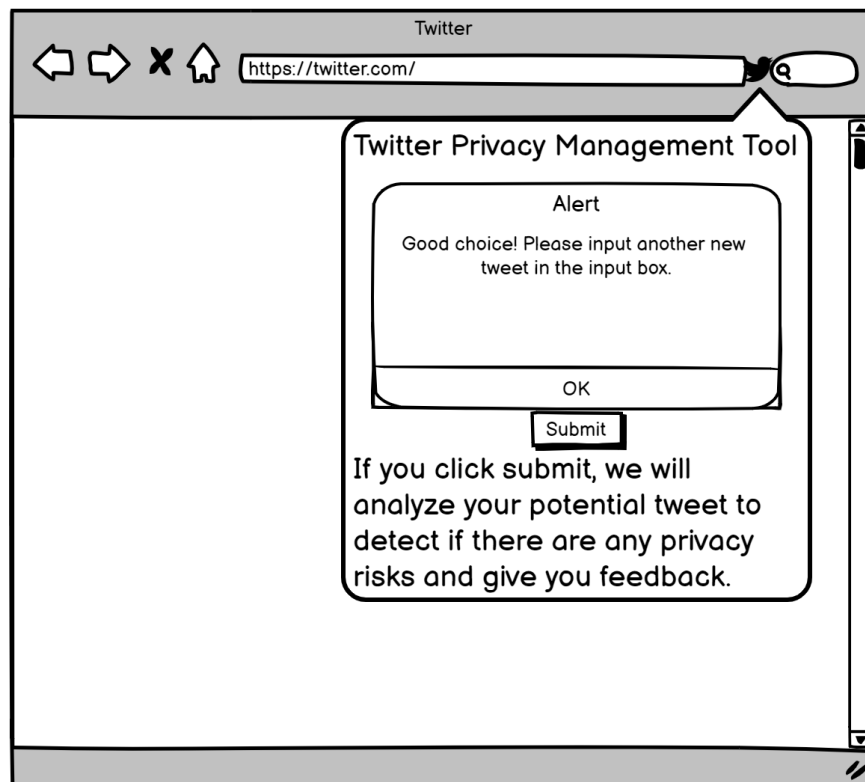


Figure 4.5: Clicked Submit + No

but the topics modelling and similarity measurement require more process about the data which will be discussed in the implementation chapter.

For this part, the flowchart of the final design scheme is shown in Figure 4.7. The program gets all the required data first and build a privacy model consists of hashtag, place name, topics modelling and similarity measurement. It will check the new tweet read from the user with the privacy model and reflect the result (fine to post or not). There are some differences between the initial design and the current scheme. The following scheme is from the initial design draft.

For audience setting:

1. Extract the hashtags or topics (if they do not have tags) based on Twitter API.
2. Generate 5-10 popular tags according to the extracted data and store these plugin generated tags in “tag bank” (these tags are different from the Twitter tags).
3. Add the corresponding audience setting (public or protected tweet/private) to the tag bank to form the “tag pairs” (e.g. COVID-public, politics-protected tweet). The whole bank is the privacy model.
4. Plugin detects the user creating a new tweet.

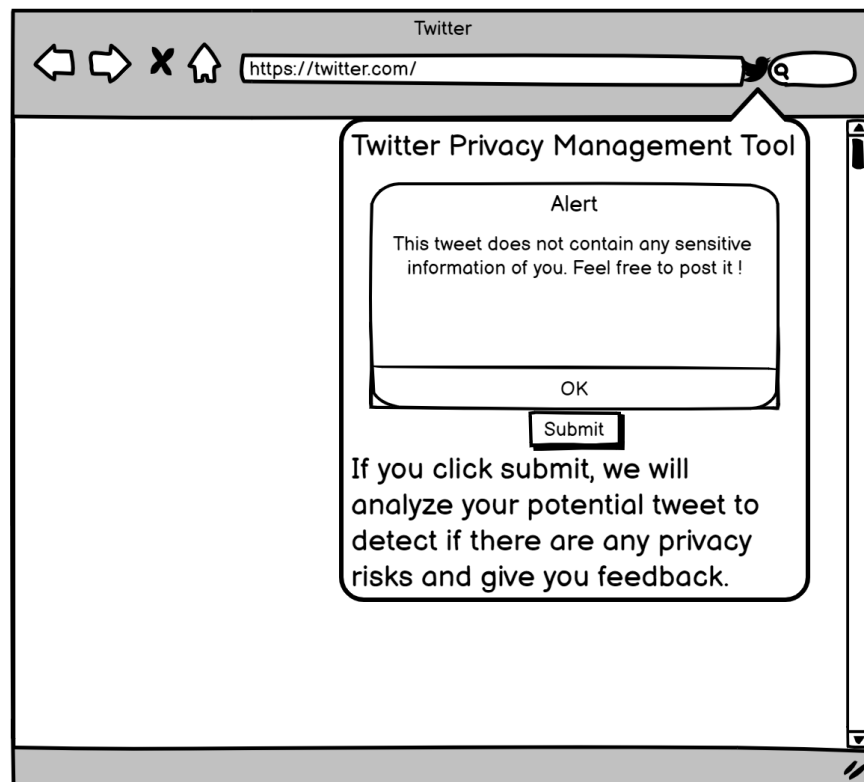


Figure 4.6: Clicked Submit Safe

5. Analyzing the tweet once the user stops typing for 5 seconds and extract the possible tags.
6. Give recommendations about audience setting based on privacy model.
7. Users select audience setting.
8. Plugin updates the privacy according to the users choice. (i.e. Users might change the audience setting for the same tag).

For location setting:

1. Extract the possible addresses and store them with the corresponding location setting to the “location bank”. The location settings include whether to display the address and if there are several addresses in the tweet which one to display.
2. Plugin detects the user creating a new tweet.
3. Analyzing the tweet once the user stops typing for 5 seconds and extract the possible addresses.
4. Give recommendations about location setting based on privacy model.

5. Users select address setting.
6. Plugin updates the privacy according to the users choice. (i.e. Users might change the location setting for the same address).

We have gone through several iterations of the design, refining and simplifying the original design to make it easier to implement. The most notable change is the merging of the audience and location settings, these two criteria should not be judged separately but jointly as they may both contain private information, so a new tweet must pass both checks before the following steps. Another improvement is removing the “tag pair” concept. In the original design plan, users need to set if the tag is public or not which is quite a lot of work for the user and not easy to implement. This plan can be interpreted as the user creating the privacy model manually, which obviously does not fit our idea of intelligence.

In summary, the design scheme is literally improved and adjusted in conjunction with the implementation scheme, which changes to varying degrees as the project progresses. This design-implementation model is based on the industry-established agile development model [36], and the end result proves to be useful and efficient for this project.

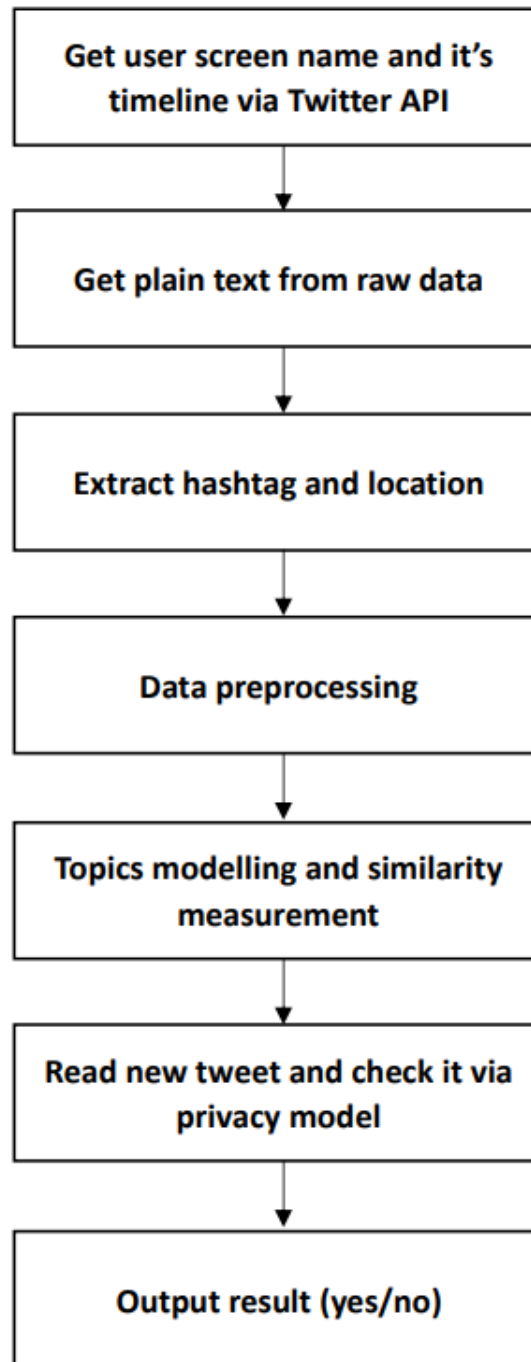


Figure 4.7: Twitter App Flowchart

Chapter 5

Implementation

This chapter presents the implementation details of the project. The implementation of the projects is divided into two parts: user interface development and standalone application development. The codes have been uploaded to GitHub repository [37].

5.1 User Interface

The implementation of UI was based on the design mockup and scheme, and also the coding work is referenced from Sxei's open source blog [38].

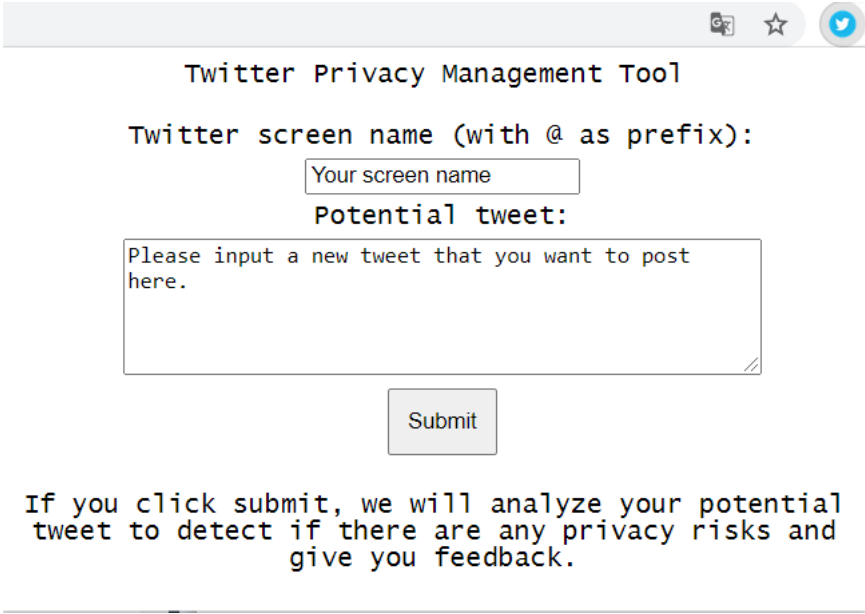
A chrome plugin will usually contain the following files.

- manifest.json : metadata about the plugin, as a chrome entry file
- HTML : page file with layout
- style : CSS file
- script : java script file
- image : plugin icon
- _locales: multi languages supported file

It is clear to see that a Chrome plugin is a web-like application that uses HTML, javascript and CSS to form a web app so this simple UI was mainly developed using basic front-end technology. The manifest file includes basic information of the plugin such as name, version, simple description and author. It also references the icon files in the same folder to render icons in the plug-in bar. The browser action also includes HTML files to render the page. Although the project folder contains the content-script

java scrip file, it is of no practical use in the user interface other than to print out some information in the console since this is only a user interface and the back end is a standalone application.

The final outcome is shown below. To run it, load the unpacked project file directly into Chrome's plugin management centre and the new plugin's icon will appear in the browser's plugin bar. Another way of running the project is to package the project folder into a special zip file with a .crx suffix, but this is more complicated to prepare (mainly because we need to prepare the private key file). This is usually used for publishing plugins, so here we present the results in the most straightforward way for convenience.



The screenshot shows a web browser window with a plugin bar at the top containing icons for Google, a star, and Twitter. The main content area has a title "Twitter Privacy Management Tool". Below the title is a label "Twitter screen name (with @ as prefix):" followed by a text input field containing "Your screen name". Underneath is a label "Potential tweet:" followed by a large text area containing the placeholder text "Please input a new tweet that you want to post here.". Below the text area is a "Submit" button. At the bottom, a message states: "If you click submit, we will analyze your potential tweet to detect if there are any privacy risks and give you feedback."

Figure 5.1: Plugin User Interface

5.2 Twitter App

This section presents each phase of the standalone Twitter application implementation in detail including data collection and preprocessing, EDA, privacy model building and the working process when a new tweet comes in.

5.2.1 Data Collection

In order to collect data from Twitter, the first step is to connect to Twitter API. Once we got the Twitter developer account, it automatically assigned the account to standard

product track [39].

Twitter provides two product tracks (i.e. standard product track and academic research product track) and three access tiers (standard, premium, enterprise). For the normal developers, we usually select standard product track and standard access tier since the functionality that the standard product provides is enough for our development. For the academic research product track, we initially intended to apply for it but the materials they asked for were too cumbersome such as the applicant's and institute's website and this product is probably for research staff, so in this project, we only used the standard product to connect to the Twitter API.

Twitter has a total of three special and secure authentication measures which include OAuth 1.0a, OAuth 2.0 Bearer Token and basic authentication [40]. Due to most of the endpoints of Twitter API are acted via OAuth 1.0a, we used the first authentication mode to access the API. OAuth authentication method requires four keys and tokens which were generated once we set up a new Twitter project and app [39]. Consumer key and secret are for Twitter to verify our developer identity, they can be simply understood as user name and password. Access token and secret are used to pull specific data.

Since our project requires a privacy model based on a user's previous tweets, we first need to pull the data from his/her previous tweets. Twitter API has a method that can get a user timeline meaning that a series of tweets previously sent by the user. The official document suggests beginners use Postman client with the user interface to send the request and receive a response since this is an intuitive way to understand the pattern of getting data from the endpoints of API. However, the response is JSON meaning that we need to parse it before we can use the text data. To get the data in a programmatic way, we decided to use the tweepy python library since it can automatically parse the response and return Tweet objects [41]. This special object contains the metadata of tweets such as full text and tweet ID, and those attributes can be easily pulled out.

The specific codes are shown in Figure 5.2 below. The first two lines are for passing the authentication by using the OAuth method. The third line is to call and instantiate API via the tweepy library. The latest tweets are pulled out by invoking the 'user_timeline' function with three important parameters explained below:

- screen_name: The user name with @ as prefix, provided by the user
- count: The number of tweets obtained in this request (200 is the upper limit that

Twitter allows)

- `tweet_mode`: If getting the full text of the tweet (extended meaning that getting the full text of the tweet)

```
# authorize twitter, initialize tweepy
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tweepy.API(auth)

# initialize a list to hold all the tweepy Tweets
alltweets = []

# make initial request for most recent tweets (200 is the maximum allowed count)
new_tweets = api.user_timeline(screen_name=screen_name, count=200, tweet_mode="extended")
```

Figure 5.2: Codes of Data Collection

After that, the program will keep pulling in 200 tweets until all tweets are fetched so that we can get all the tweets from the user. This process is usually time-consuming, so we also have an alternative way of pulling only the first two hundred tweets, which in practice improves the speed of the application launch but may be less accurate.

5.2.2 Privacy Model Building

5.2.2.1 Extract Place Information

The extraction of place information follows the object model of the data dictionary. Twitter API abstracts some common data dictionaries into objects such as Tweet object and User object. Each type of object has its specific attributes. For Tweet object, the place is a default attribute and we can directly get it from the Tweet objects that we stored before. Since we store all the Tweet objects in a list in the data collection phase, the place name can be pulled out from the Tweet object list one by one. It is true that some tweets do not have location information then we skip them and only extract the ones that contain place information. Also, in order to unify the structure of the location information, we only extract names with the place type city, other types such as country will be excluded. The final place name list should be made up of city names such as Edinburgh or London, etc.

5.2.2.2 Extract Hashtag Information

The hashtag extraction is similar to place extraction but the difference is that we cannot directly get the hashtag from the Tweet object. Unlike location information, hashtags cannot be extracted as one of the attributes of the tweet object. Instead, the hashtag information is stored in entities which are parsed from the text of the tweet. So what we did is to first get the entities from the Tweet objects and get the hashtag from the entities one by one.

Once we have extracted all the hashtags, our next step is to filter out some of the low-frequency ones. The reason is that we cannot treat a hashtag that appears only once or twice as the user's usual tag so that a threshold was set to filter the tags with low frequency. To count the word frequency, we call a function 'FreqDist()' in NLTK - a very common natural language processing python library - to quickly and easily calculate word frequency. The final step is to select hashtags that exceed the word frequency threshold and compose a list for subsequent use. The following Figure 5.3 shows the specific implementation details. It is worth mentioning that the threshold value was decided via some tests which will be discussed in the next chapter.

```
from nltk.probability import FreqDist
def getHashtags(alltweets):
    hashtags = []
    for tweet in alltweets:
        hashtaginfo = tweet.entities.get("hashtags")
        if len(hashtaginfo) != 0:
            hashtags.append(hashtaginfo[0].get("text").lower())
    tag_freq = FreqDist(hashtags)
    # print(tag_freq.most_common())
    # filter out the tags with high frequency
    tag2class = [tag[0] for tag in tag_freq.most_common() if tag[1] >= 5] # high possibility
    return tag2class
```

Figure 5.3: Hashtag Extraction

5.2.2.3 Data Preprocessing

We set the data pre-processing after extracting the location and hashtag information because we need to store the complete words of the place name and hashtag, once the data is pre-processed these words will be missing some prefixes and suffixes, and these two types of information need to be matched exactly, the data pre-processing will inevitably affect the accuracy of the match. For example, the place names are unique

and cannot be pruned so the place names should be fully matched. The hashtags in Twitter also represents unique topics and they cannot be obscured as similar meaning words.

This part contains a number of common data pre-processing steps :

1. Tokenization: Split the text into separate words (tokens) and store them in a list.
2. Casefolding: Convert all letters to lower case.
3. Stopwords_removing: Remove all stopwords that are not meaningful (i.e. The stopwords list here is pre-saved from the coursework material provided in Text Technology and Data Science course).
4. Stemming: Trim off prefixes and suffixes that do not affect the meaning of the word. Porter stemmer is a popular stemming python library.

```
# preprocess used for building inverted index including tokenisation, casefolding,
# removing stop words and porter stemming.
def preprocess(text):
    remove_url = re.sub(r"http\S+", "", text)
    # split non-letter characters and store them into a list.
    regEx = re.compile("\W").split(remove_url)
    token = [i for i in regEx if i != '']
    # all in lower case.
    casefolding = [s.lower() for s in token if isinstance(s, str) == True]
    # removing stop words.
    stopwords_removed = [w for w in casefolding if not w in stopwords]
    # Porter stemmer by using snowball stemmer lib.
    stemmer = snowballstemmer.stemmer('english')
    return stemmer.stemWords(stopwords_removed)
```

Figure 5.4: Data Preprocessing

5.2.2.4 Topic Modelling

The standard step before fitting the topic model is to convert the raw text data into a bag-of-words matrix [42]. In fitting the LDA model, we set the number of topics as 10 and the random state as 1 since we would like to fit the model with the default parameters and to avoid the influence of random seed which may lead to bias in the results of each fit [43].

These LDA model functions were provided by a popular third party topic modelling python library Gensim [44]. The result of the model can be shown by calling

```
# fitting lda model
tweets_lda = LdaModel(tweets_bow, num_topics=10, id2word=text_dict, random_state=1)
```

Figure 5.5: LDA Fitting

‘show_topics()’ function. Each topic is made up of keywords and a score for each keyword, the higher the score the higher the percentage of the topic the word is meant to be in. In other words, the meaning of the topic is close to the meaning of the high-frequency words. The final step is also to get all words from the topics and store them in a list for further use.

5.2.2.5 Similarity Measurement

The LSI model is also provided by Gensim [44]. The ‘MatrixSimilarity()’ function of the ‘similarities’ interface can generate a matrix that stores the similarity information for each tweet in the corpus. Then, the new tweet can be input to compare with the existing tweets. The reason why the index shown in figure 5.6 to be saved is that we should maintain the consistency of the index. If not saved then the index result may be very different each time.

```
from gensim import similarities
# transform corpus to LSI space and index it
index = similarities.MatrixSimilarity(lsi[corpus])
index.save('deerwester.index')
index = similarities.MatrixSimilarity.load('deerwester.index')
# perform a similarity query against the corpus
sims = index[vec_lsi]
```

Figure 5.6: Similarity Measurement

5.2.3 Working Process

The process of testing a new tweet has four phases and each of these stages corresponds to one of the stages in the model building process described above.

1. Recognize the city name in the new tweet.

This stage corresponds to place extraction above. The program gets the city name from the new tweet text via GeoText python library and compare them with

the previous place name list. If all the city names of new tweets are included in the previous place name list, then we pass it.

```
placename = GeoText(newtweet).cities
if len(placename) != 0 and set(placename) <= set(placeslist):
    return "This new tweet is fine to post ! place"
```

Figure 5.7: Get Place

An alternative way to check the place name is to manually set up a city name list but it is not that efficient in traversing the whole list so we finally decided to use the python library.

2. Get hashtags from the new tweet. If all the hashtags of the new tweet have appeared in previous tweets, then we decide to pass it.

The logic of getting hashtags is similar to that of getting place. The way we extract # is to check the first character of every word shown in figure 5.8. While detecting the hashtag, the second if logic also makes sure that the string after it is not empty.

```
# checking the first character of every word
if word[0] == '#':
    if word[1:] != "":
        # adding the word to the hashtag_list
        hashtag_list.append(word[1:].lower())
```

Figure 5.8: Check Hashtag

3. Determine if the number of key tokens in new tweets is over the threshold.

In this step, the program finds out the keywords from the new tweet. The criteria for determining whether a word is a keyword is whether it matches the word in the keyword list (i.e. 'alltopicWords' in the code shown in figure 5.9) which was generated by LDA model. In other words, the keywords in hot topics generated by LDA form a keywords pool and we would like to compare the number of the keywords in new tweets with the threshold value. The threshold value depends on the total number of keywords pool. For example, if the program finds 4 keywords in the new tweets and the threshold value is 3 then the program passes

it. The way we decided the threshold value also depended on the parameter tests that will be discussed later.

```
# 3. Determine if the number of key tokens in new tweets
#     is over the threshold
txt = preprocess(newtweet)
counter = 0
for token in txt:
    if token in alltopicWords:
        counter += 1
        print(token)
if counter >= round(len(alltopicWords) / 10):
    return "This tweet is fine to post! topics word"
```

Figure 5.9: Check Topic Model

4. Using LSI to measure the similarity of new tweets and the corpus.

The similarity query was performed in the last line of figure 5.6. This step is to sort the results (i.e. 'sims' in the code) according to similarity scores and the sorted result is a list shown the tweets from the highest score. The threshold here defines which tweets can be considered similar to previously existing tweets. Once the new tweet has a score higher than the threshold value, the program passes it.

```
sims = sorted(enumerate(sims), key=lambda item: -item[1])
for doc_position, doc_score in sims:
    if doc_score >= 0.97:
        return "This tweet is fine to post! similarity"
```

Figure 5.10: Check Similarity

In summary, the program first checks if the new tweet has the same place or hashtags, then check if it has enough keywords, finally check if it is similar to one of the previous tweets. Passing any of these steps will deem the new tweet to be privacy compliant, if all steps are not passed the program will indicate that the tweet has a high risk of the privacy breach. The reason for this implementation is that it is not possible to require users to tweet in strict compliance with every privacy requirement, this design looks loose but in fact, gives the decision back to the user.

Chapter 6

Experiment and Evaluation

This chapter discusses the full details of the experiment and evaluation. This study strictly follows the rules of Twitter and was approved by the ethics team of the School of Informatics at the University of Edinburgh with RT number 2021/699995. The experiment of this part is divided into a randomised experiment oriented towards determining certain important parameters and a user study oriented towards the participants.

6.1 Experiment Environment

All tests were run on a windows laptop with pycharm or jupyter notebook platforms installed python 3.8.10. The following tables show the hardware configuration and necessary third party python libraries.

CPU	Intel Core i7-10875H
RAM	16.0 GB (15.8 GB available)

Table 6.1: Hardware Configuration

Snowballstemmer	2.1.0
Tweepy	3.10.0
NLTK	3.6.2
Gensim	4.0.1
Geotext	0.4.0

Table 6.2: Python Libraries

6.2 Parameter Experiment

This phase is for determining some important threshold values mentioned before. The data collected from random Twitter users were also approved by both the Twitter team and the ethics team of the School of Informatics. According to the above chapters, there are three threshold values that need to be decided:

1. The value to determine if the hashtag is hot.
2. The value to determine if the new tweet has enough hot keywords.
3. The value to determine if the new tweet is highly similar to the previous tweets.

The test plan is to select three public official accounts and get their tweets data. The official accounts were chosen for testing because they have a large enough volume of data and the topics tweeted by each account are somewhat related, making it easier for the program to detect this high-risk privacy breach when they post new tweets that are not related to the majority of the tweets.

6.2.1 Hashtag Experiment

The threshold value for hashtag filtering is actually word frequency and the experiment here is for building the privacy model. If the hashtag word frequency in the previous tweets data is larger than this value, then we can put this hashtag word into the hashtag list which is also a part of the privacy model. We remove the five most recent tweets from each user's timeline data as test tweets and test five tweets in the previous tweets data. This testing scheme is also the same in the following topic words and similarity experiments. The ratio value is the experiment result meaning that how many tweets have passed through the program (e.g. 100% means five tweets are all passed and 40% means 2 tweets are passed out of 5). The expected outcome should be 100% meaning that all of the latest tweets should be passed and the higher the result of the experiment, the better performance of the threshold value. We set up three experimental groups for each public account as it is shown in table 6.3. It is clear that value 5 has the best performance. The implementation of this part was shown in figure 5.3.

6.2.2 Topic Words Experiment

The threshold value in this part is the minimum number of keywords in new tweets and this experiment is for user experiments. If the number of keywords in the new

Accounts	Threshold Value		
	5	10	20
Account 1	100%	80%	40%
Account 2	80%	80%	20%
Account 3	100%	60%	0%

Table 6.3: Hashtag Threshold Test

tweet is larger than the threshold value, then the program will pass it. The values in the table 6.4 are the divisor and the real threshold values are denoted as “the number of keywords list divided by the value in the table”. The expected outcome of this part is to choose a medium one. If the outcome is very high, it means that the program will pass the new tweet anyway and the threshold would be meaningless. Thus, the best one is divisor 10 and the implementation of this part was shown in figure 5.9.

Accounts	Threshold Value		
	/5	/10	/20
Account 1	100%	80%	40%
Account 2	100%	80%	20%
Account 3	100%	60%	0%

Table 6.4: Topic Words Threshold Test

6.2.3 Similarity Experiment

This experiment is also for the user test. The threshold value denotes the similarity score that the new tweet should have. Once the new tweet gets a higher score then the program will pass it. We also would like to choose a medium one as our final value. If the result is very high, it also means that the program may pass the tweet anyway and the value would be not meaningful. However, if the ratio is very low that means the program blocks most of the new tweets, it is also not a good choice. Thus, the final solution is 0.97 and the codes were shown in figure 5.10 which might be more intuitive and easy to understand.

Accounts	Threshold Value				
	0.8	0.9	0.95	0.97	0.99
Account 1	100%	100%	100%	80%	0%
Account 2	100%	100%	100%	80%	20%
Account 3	100%	100%	100%	100%	0%

Table 6.5: Similarity Threshold Test

6.3 User Study

6.3.1 Participant Information Survey

The participant information sheet shown in appendix B introduce all the information that the participants should know and it was approved by the ethics team. To make it easier for participants to know this information and collect data, we designed and distributed a survey using Microsoft form (shown in Appendix C) that includes all the information in the sheet and also the data we want to collect from participants. This survey briefly introduces the project including the researchers, purpose, risks, data protection and what we will do in the study. In order to get their consent efficiently, we set up a compulsory question to ask if they agree with this information. Once they select ‘yes’, we ask them to provide their Twitter screen name and the potential 5 tweets that they want to post. Then, we manually input their Twitter screen name and their potential tweets as new tweets to the program for test one by one. The result would be how many tweets are passed and how many are blocked. An additional option is that they can provide their email address so that we can send them our experiment result and ask them if they are happy with it.

6.3.2 User Feedback

Finally, we received feedback from 5 participants which mean that all of the participants give us feedback. All of them are studying computer science, some of them are master students at the University of Edinburgh and some are undergraduate students at the University of Liverpool. So the participants have a systematic UK education experience and a basic knowledge of computer science, and they are familiar with the process of projects in UK universities, especially the user study to varying degrees. This background and experience are beneficial to our user study. Due to the ethics

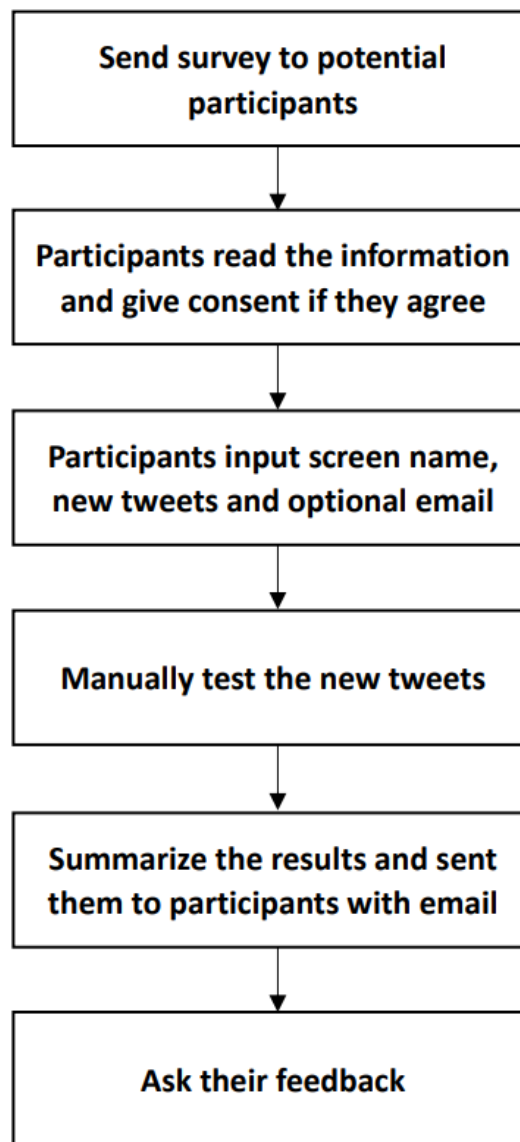


Figure 6.1: User Study Process

rules, we cannot show the raw text which may contain personal privacy information in the dissertation so that we only discuss the summary of the experiment result and their feedback.

The following table 6.6 shows the user study result with accuracy and the general topics they are interested in. It is worth noting that higher accuracy values are not better, and some people are able to accept predictions that are not 100% accurate because they provide new tweets that already contain privacy-sensitive information. Thus, the aim of the study is to make participants as happy as possible with the results of the experiment instead of improving the accuracy.

The following paragraphs will fully discuss the specific cases for each participant.

Participants	Accuracy	Topics
Participant 1	100%	Edinburgh, study
Participant 2	60%	video game
Participant 3	20%	travel
Participant 4	40%	workout
Participant 5	60%	video game

Table 6.6: User Study Result

We will choose some hypothetical tweets from participants shown here meaning that we will remove some privacy information in the tweets they provided and just take the meaning of the tweets out.

For participant 1, the experiment result shows that this person is concerned about the topic of University of Edinburgh, graduation and study. So the main contents of the previous tweets are about studies in Edinburgh. It also shows that this participant enjoys documenting his/her learning life. One of the hypothetical tweets of this participant is “Although the summer graduation ceremony was cancelled, there is still a graduation event in Edinburgh castle in late August. I REALLY want to attend it!!!”. This tweet follows the general topics of the participant with Edinburgh and study so that the result is very good. The five potential new tweets are all passed since they are all related to Edinburgh and study and this participant is satisfied with the result.

Participants 2 and 5 are most likely video game enthusiasts. The high-frequency words in their privacy model are roughly gaming laptops, play station 4 consoles, discord and some game specifics, etc. Their user profiles are quite similar so we put them together in this paragraph. The potential new tweets pass 3 out of 5 and all passed tweets are related to games unsurprisingly. They are generally happy with the result since the tweets that are blocked are about their daily life with some place names. For example, they have tweets such as “oh I really like the new version of the league of legends. The new champ with the new skin looks really nice” which is about video games and also “The weather is so nice in Edinburgh today that I want to go out and get some ice cream!” which is about daily life. The possible reasons are unrelated topics or place names that have not appeared in the game-related topics before such as the above latter example. The former example follows the topic in the privacy model that is video game but the weather and ice cream are not in the privacy model. So the model worked badly in this case which has multiple unrelated topics.

Participant 3 is a travel-lover with most of the tweets are about travel, hiking and climbing. The privacy model of this person also includes some famous tourist attraction names such as Isle of Skye, lake district, highland, etc. This data set is also the most uncertain of all participants since the tweets include place names and hashtags and most of the hashtags appeared only once. Most people do not travel to the same place frequently and those who travel a lot always change their destination. The result is that only one tweet pass the experiment and this participant clearly indicated dissatisfaction with the outcome. The reason why the model performed bad in this case is similar to that for participants 2 and 5 cases. Once the privacy context is complex, for example containing several different topics or many different place names or hashtags, the model would be ‘confused’. The possible solution is to adjust the threshold value according to the large volume of user tests since the larger threshold value has greater inclusivity for multiple topics and this improvement could be also effective for the above cases of participant 1,2 and 5.

Participant 4 probably loves fitness and workout. This person’s tweets are usually a record of training routine such as daily running distance, fat loss exercise hours, etc. According to the feedback, this person’s tweets are mainly pictures, with very little text, sometimes just a few words. This poses a major challenge for topic modelling such as LDA, where we are mostly unable to generate valid topics due to a lack of sufficient data, and instead identify privacy models by keyword. This participant is generally fine with the result that 2 of 5 with sufficient text are passed. One possible way to improve on this problem is to use some image tagger APIs or python libraries to collect the words in the image and add them to the privacy model via LDA topic modelling. Another challenging and powerful approach is to use some image recognition technology to extract the objects in the picture such as workout equipment, convert them to words and furthermore add them to the privacy model. The idea of this imaginative method is to convert from various formats (i.e. image, speech) to a uniform text format and build a text-based privacy model.

Chapter 7

Conclusions

In this project, we designed and implemented a Twitter privacy management tool to help users check if their potential new tweets include privacy sensitive information. This was implemented by building a privacy model and checking the similarity between new tweets and the model. If the similarity is high then pass it, otherwise block it.

In the process of the project, we extracted the place names which is direct privacy information and added them to the privacy model. We also pulled the hashtag out since the hashtag is actually the user-defined topic and it can represent the topic of the tweet. We also tried to understand the topics of the tweet without place names or hashtags via LDA topic model. The similarity measurement implemented by LSI model is also an important way to check the new tweet. There were also some experiments and evaluations about public official users and participants where the former tests were for determining the important parameters in the program and the latter tests were for the user study.

7.1 Limitations

There are some limitations that we found in the evaluation phase. This model is unfriendly to users who add different place names frequently (e.g. participant 3), and each time the new name is blocked by the program as sensitive information. Also, this system does not perform well on tweets that frequently change the subject (e.g. participant 2&5). For the tweets with little text, the system performs generally fine but still not good enough (e.g. participant 4). It is clear that the system performs very well for the users who focus on one or a few topics with sufficient text but lacks usability in

some complex and changing environments. In short, it performs well in a simple privacy environment but bad in a complicated privacy environment with multiple topics and other formats of information such as images.

7.2 Future Work

The first future work should be focusing on the integration of the front-end user interface and the back-end Twitter application. This project implemented the complete standalone application with only the user interface so the interaction of front and back-end data will be an important and challenging task. The issues mentioned in the limitation should also be fixed to improve the usability of the system and some of the possible future improvements are discussed in the user feedback section.

In general, adjusting the threshold value can improve the inclusivity for the complex multi topics privacy environment and the image tagger can provide more information for the privacy model. Also, there will be more approaches such as improving the model that can be explored in the future.

Bibliography

- [1] Richard A Posner. The right of privacy. *Ga. L. Rev.*, 12:393, 1977.
- [2] Alessandro Acquisti. Privacy and security of personal information. In *Economics of Information Security*, pages 179–186. Springer, 2004.
- [3] Julie E Cohen. What privacy is for. *Harv. L. Rev.*, 126:1904, 2012.
- [4] Fred H Cate et al. Privacy in the information age, 1997.
- [5] Neil Vidmar and David H Flaherty. Concern for personal privacy in an electronic age. *Journal of Communication*, 35(2):91–103, 1985.
- [6] Arthur R Miller. Personal privacy in the computer age: The challenge of a new technology in an information-oriented society. *Mich. L. Rev.*, 67:1089, 1968.
- [7] SE Kruck, Danny Gottovi, Farideh Moghadami, Ralph Broom, and Karen A Forcht. Protecting personal privacy on the internet. *Information Management & Computer Security*, 2002.
- [8] David F Linowes. Must personal privacy die in the computer age. *ABAJ*, 65:1180, 1979.
- [9] David Lyon and Elia Zureik. *Computers, surveillance, and privacy*. U of Minnesota Press, 1996.
- [10] Antony Mayfield. What is social media. 2008.
- [11] Liu Yahui, Zhang Tieying, Jin Xiaolong, and Cheng Xueqi. Personal privacy protection in the era of big data. *Journal of Computer Research and Development*, 52(1):229, 2015.

- [12] Marina Sokolova and Stan Matwin. Personal privacy protection in time of big data. In *Challenges in computational statistics and data mining*, pages 365–380. Springer, 2016.
- [13] Daniel J Solove. Understanding privacy. 2008.
- [14] Helen Nissenbaum. *Privacy in context*. Stanford University Press, 2020.
- [15] Michael S Bernstein, Eytan Bakshy, Moira Burke, and Brian Karrer. Quantifying the invisible audience in social networks. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 21–30, 2013.
- [16] Mainack Mondal, Peter Druschel, Krishna P Gummadi, and Alan Mislove. Beyond access control: Managing online privacy via exposure. In *Proceedings of the Workshop on Useable Security*, pages 1–6, 2014.
- [17] Ricard Fogues, Jose M Such, Agustin Espinosa, and Ana Garcia-Fornes. Open challenges in relationship-based privacy mechanisms for social network services. *International Journal of Human-Computer Interaction*, 31(5):350–370, 2015.
- [18] Raymond Heatherly, Murat Kantarcioglu, and Bhavani Thuraisingham. Preventing private information inference attacks on social networks. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1849–1862, 2012.
- [19] Nadin Kökciyan and Pınar Yolum. P r i g uard: A semantic approach to detect privacy violations in online social networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2724–2737, 2016.
- [20] Michelle X Zhou, Jeffrey Nichols, Tom Dignan, Steve Lohr, Jennifer Golbeck, and James W Pennebaker. Opportunities and risks of discovering personality traits from social media. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, pages 1081–1086. 2014.
- [21] Jennifer Golbeck and Derek Hansen. A method for computing political preference among twitter followers. *Social Networks*, 36:177–184, 2014.
- [22] Cuneyt Gurcan Akcora, Barbara Carminati, and Elena Ferrari. Risks of friendships on social networks. In *2012 IEEE 12th International Conference on Data Mining*, pages 810–815, 2012.

- [23] Kun Liu and Evimaria Terzi. A framework for computing the privacy scores of users in online social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(1):1–30, 2010.
- [24] Lujun Fang and Kristen LeFevre. Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web*, pages 351–360, 2010.
- [25] David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012.
- [26] Steve Wilson. Text technologies for data science infr11145 comparing text corpora. 2020.
- [27] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [28] Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, and Liang Zhao. Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, 78(11):15169–15211, 2019.
- [29] Roger B Bradford. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 153–162, 2008.
- [30] Radim Řehřek. Subspace tracking for latent semantic analysis. In *European Conference on Information Retrieval*, pages 289–300. Springer, 2011.
- [31] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. 2009.
- [32] Himanshu Pareek, Sandeep Romana, and PRL Eswari. Application whitelisting: approaches and challenges. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, 2(5):13–18, 2012.
- [33] David Alfred Ostrowski. Using latent dirichlet allocation for topic modelling in twitter. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pages 493–497. IEEE, 2015.

- [34] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
- [35] Jakob Nielsen. Ten usability heuristics, 2005.
- [36] James Shore et al. *The Art of Agile Development: Pragmatic guide to agile software development.* ” O’Reilly Media, Inc.”, 2007.
- [37] Github repository. <https://github.com/MingruiCai/Msc-Project>.
- [38] sxei. Chrome plugin (extension) development guide - good note’s blog. <http://blog.haoji.me/chrome-plugin-develop.html>.
- [39] Getting access to the twitter api. <https://developer.twitter.com/en/docs/twitter-api/getting-started/getting-access-to-the-twitter-api>.
- [40] OAuth 1.0a. <https://developer.twitter.com/en/docs/twitter-api/getting-started/getting-access-to-the-twitter-api>.
- [41] tweepy.api-twitter api v1.1. https://docs.tweepy.org/en/latest/api.html?highlight=timeline#tweepy.API.user_timeline.
- [42] Twitter topic modeling. using machine learning (gensim linear... — by amin azad — towards data science. <https://towardsdatascience.com/twitter-topic-modeling-e0e3315b12e2>.
- [43] Lda topic modeling with tweets. making sense of unstructured text — by rob zifchak — towards data science. <https://towardsdatascience.com/lda-topic-modeling-with-tweets-deff37c0e131>.
- [44] Topics and transformations — gensim. https://radimrehurek.com/gensim/auto_examples/core/run_topics_and_transformations.html#available-transformations.

Appendix A

Twitter Developer Application Email

Re: Twitter developer account application [ref:_00DA0K0A8._5004w2ASZ9z:ref]

CAI Mingrui <M.Cai-7@sms.ed.ac.uk>

Thu 17/06/2021 17:00

To: developer-accounts <developer-accounts@twitter.com>

Hi there,

Thanks for your reply.

The use of Twitter APIs is only for academic purpose.

The core use case is that when users create a new tweet and they want to post it, my Chrome plugin would analyze the text of the tweet and extract the topics of the tweet, and provide suggestions about whether this tweet should be public or private. My app will not analyze the whole bunch of data on Twitter, it will only use the data from the person who uses the app. The reason is that the privacy preference for every user is different so our suggestions should be mostly based on the user who uses the app. Furthermore, the plugin will generate several tags that might be related to the tweet and ask users whether they want to attach those tags to the tweet and whether they want to set those tags as private or public. Once the tags' privacy is set, we have a "tag bank" storing pairs of tags and corresponding privacy setting (private or public) also called the privacy model. Whenever the user creates a new tweet, the plugin will automatically analyze the tweet and find the key topics in the tag bank and give suggestions according to the privacy model. The privacy model can also extend to location information, analyzing the location information in the tweet and give suggestions about whether to share the current location or not. Not like in Facebook, privacy in Twitter is binary (public or private) most of the time, so that the classification tasks are binary.

For the analysis of the tweet, I would like to use the analysis tools of Twitter APIs to analyze the possible topics in the tweet. The NLP and ML techniques embedded in Twitter API is fair enough for my app.

For the interaction of the plugin and Twitter accounts, the app will not intervene too much in Twitter accounts. The plugin will only analyze the tweet and give suggestions but not force the users to set privacy settings. Users can freely control their accounts and my app is only for intelligent suggestions and help.

In most cases, the content will not display outside of Twitter because the plugin is just a tool to analyze tweets and give suggestions. In some rare cases, users might want to manage several tweets' privacy at the same time, then the content of the tweets might display in the plugin to remind users which suggestion corresponds to which tweet. If only suggestions are displayed, users may not be able to find the specific tweets corresponding to these suggestions, so the user experience will be greatly reduced. The plan of displaying the content will be to display the first few words or topics to remind the users of the pair of privacy setting and tweet.

I really hope you can approve my application, which is important for my academic career.

Thank you very much.

Best regards,
Mingrui

From: developer-accounts <developer-accounts@twitter.com>**Sent:** 17 June 2021 14:23

To: CAI Mingrui <M.Cai-7@sms.ed.ac.uk>

Subject: Twitter developer account application [ref:_00DA0K0A8._5004w2ASZ9z:ref]

This email was sent to you by someone outside the University.

You should only click on links or attachments if you are certain that the email is genuine and the content is safe.



Hello,

Thanks for your interest in building on Twitter.

Before we can finish our review of your developer account application, we need some more details about your use case.

The types of information that are valuable for our review include:

- The core use case, intent, or business purpose for your use of the Twitter APIs.
- If you intend to analyze Tweets, Twitter users, or their content, share details about the analyses you plan to conduct, and the methods or techniques.
- If your use involves Tweeting, Retweeting, or liking content, share how you'll interact with Twitter accounts, or their content.
- If you'll display Twitter content off of Twitter, explain how, and where, Tweets and Twitter content will be displayed with your product or service, including whether Tweets and Twitter content will be displayed at row level, or aggregated.

Just reply to this email with these details. Once we've received your response, we'll continue our review. We appreciate your help!

Thanks,

Twitter

Twitter, Inc. 1355 Market Street, Suite 900 San Francisco, CA 94103



ref:_00DA0K0A8._5004w2ASZ9z:ref

Appendix B

Participant Information Sheet

Participant Information Sheet

Project title:	A plugin that reminds you the privacy settings in online systems
Principal investigator:	Nadin Kokciyan
Researcher collecting data:	Mingrui Cai

This study was certified according to the Informatics Research Ethics Process, RT number 2021/699995. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers?

The research is being carried out as part of the MSc final project of Mingrui Cai at the University of Edinburgh. This project is supervised by Nadin Kokciyan.

What is the purpose of the study?

The purpose of this study is to design a tool that could help Twitter users to share content while preserving their privacy. The tool analyses the content of a tweet to be shared and compares this tweet to previous tweets of a user. If the tweet to be shared is about a topic that the user shares less frequently, the tool makes a recommendation to the user about not sharing it. Otherwise, it makes a sharing recommendation. We would like to measure the effectiveness of our tool in making personal recommendations for specific Twitter users.

Why have I been asked to take part?

We are looking for Twitter users with public accounts, who are interested in managing their privacy on Twitter. You have been asked to participate because we believe that you have this type of experience.

Do I have to take part?

No – participation in this study is entirely up to you. You can withdraw from the study at any time, without giving a reason. Your rights will not be affected. If you wish to withdraw, contact the PI. We will stop using your data in any publications or



presentations submitted after you have withdrawn consent. However, we will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part?

After you give consent for this study, you will be asked for your Twitter ID first. In order to understand what you share on Twitter (e.g. tweets about music), we will analyse your tweets shared publicly. We will then identify the topics that you are more willing to share. You will then be asked for 5-10 hypothetical tweets that you would be willing to share on Twitter. You can optionally provide your email address if you want to learn about your results. The survey will take 10-15 minutes to complete. Once you complete the survey, we will be running our tool to generate sharing recommendations and then compare them against your responses.

Are there any risks associated with taking part?

There are no significant risks associated with participation.

Are there any benefits associated with taking part?

Your participation can help us to understand the privacy expectations of Twitter users. This study will also inform privacy researchers to design tools to help social media users to preserve their privacy better.

What will happen to the results of this study?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of two years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team: Mingrui Cai and Nadin Kokciyan.

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk. Our tool will be run by the lead researcher on their local machine, and the anonymised data will be used during the experiments.

What are my data protection rights?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk. Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk.

Who can I contact?

If you have any further questions about the study, please contact the lead researcher, Mingrui Cai <s2085934@ed.ac.uk>.

If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk. When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on <http://web.inf.ed.ac.uk/infweb/research/study-updates>.



Consent

By proceeding with the study, I agree to all of the following statements:

- I have read and understood the above information.
- I understand that my participation is voluntary, and I can withdraw at any time.
- I consent to my anonymised data being used in academic publications and presentations.
- I allow my data to be used in future ethically approved research.



Appendix C

Participant Survey Sheet

Participant Intention Survey for Twitter Privacy Study

This question is for people who have intentions to participate Twitter privacy study which is a part of Mingrui's Msc project.

You should read and understand the following information before you take part in this study.

* Required

Participant Information

Project title:

A plugin that reminds you the privacy settings in online systems

Principal investigator:

Nadin Kokciyan

Researcher collecting data:

Mingrui Cai

This study was certified according to the Informatics Research Ethics Process, RTnumber 2021/699995. Please take time to read the following information carefully. You should keep this page for your records.

Who are the researchers ?

The research is being carried out as part of the Msc final project of Mingrui Cai at the University of Edinburgh. This project is supervised by Nadin Kokciyan.

What is the purpose of the study ?

The purpose of this study is to design a tool that could help Twitter users to share content while preserving their privacy. The tool analyses the content of a tweet to be shared and compares this tweet to previous tweets of a user. If the tweet to be shared is about a topic that the user shares less frequently, the tool makes a recommendation to the user about not sharing it. Otherwise, it makes a sharing recommendation. We would like to measure the effectiveness of our tool in making personal recommendations for specific Twitter users.

Why have I been asked to take part ?

We are looking for Twitter users with public accounts, who are interested in managing their privacy on Twitter. You have been asked to participate because we believe that you have this type of experience.

Do I have to take part ?

8/6/2021 No – participation in this study is entirely up to you. You can withdraw from the study at any time, without giving a reason. Your rights will not be affected. If you wish to withdraw, contact the PI. We will stop using

your data in any publications or presentations submitted after you have withdrawn consent. However, we will keep copies of your original consent, and of your withdrawal request.

What will happen if I decide to take part ?

After you give consent for this study, you will be asked for your Twitter ID first. In order to understand what you share on Twitter (e.g. tweets about music), we will analyse your tweets shared publicly. We will then identify the topics that you are more willing to share. You will then be asked for 5-10 hypothetical tweets that you would be willing to share on Twitter. You can optionally provide your email address if you want to learn about your results. The survey will take 10-15 minutes to complete. Once you complete the survey, we will be running our tool to generate sharing recommendations and then compare them against your responses.

Are there any risks associated with taking part ?

There are no significant risks associated with participation.

Are there any benefits associated with taking part ?

Your participation can help us to understand the privacy expectations of Twitter users. This study will also inform privacy researchers to design tools to help social media users to preserve their privacy better.

What will happen to the results of this study ?

The results of this study may be summarised in published articles, reports and presentations. Quotes or key findings will be anonymized: We will remove any information that could, in our assessment, allow anyone to identify you. With your consent, information can also be used for future research. Your data may be archived for a maximum of two years. All potentially identifiable data will be deleted within this timeframe if it has not already been deleted as part of anonymization.

Data protection and confidentiality.

Your data will be processed in accordance with Data Protection Law. All information collected about you will be kept strictly confidential. Your data will be referred to by a unique participant number rather than by name. Your data will only be viewed by the researcher/research team: Mingrui Cai and Nadin Kokciyan.

All electronic data will be stored on a password-protected encrypted computer, on the School of Informatics' secure file servers, or on the University's secure encrypted cloud storage services (DataShare, ownCloud, or Sharepoint) and all paper records will be stored in a locked filing cabinet in the PI's office. Your consent information will be kept separately from your responses in order to minimise risk. Our tool will be run by the lead researcher on their local machine, and the anonymised data will be used during the experiments.

What are my data protection rights ?

The University of Edinburgh is a Data Controller for the information you provide. You have the right to access information held about you. Your right of access can be exercised in accordance Data Protection Law. You also have other rights including rights of correction, erasure and objection. For more details, including the right to lodge a complaint with the Information Commissioner's Office, please visit www.ico.org.uk (<http://www.ico.org.uk>). Questions, comments and requests about your personal data can also be sent to the University Data Protection Officer at dpo@ed.ac.uk (<mailto:dpo@ed.ac.uk>).

Who can I contact ?

If you have any further questions about the study, please contact the lead researcher, Mingrui Cai <s2085934@ed.ac.uk (<mailto:s2085934@ed.ac.uk>)>.

If you wish to make a complaint about the study, please contact inf-ethics@inf.ed.ac.uk (<mailto:inf-ethics@inf.ed.ac.uk>). When you contact us, please provide the study title and detail the nature of your complaint.

Updated information.

If the research project changes in any way, an updated Participant Information Sheet will be made available on <http://web.inf.ed.ac.uk/infweb/research/study-updates> (<http://web.inf.ed.ac.uk/infweb/research/study-updates>).

Consent

By proceeding with the study, I agree to all of the following statements:

- I have read and understood the above information.
- I understand that my participation is voluntary, and I can withdraw at any time.
- I consent to my anonymised data being used in academic publications and presentations.
- I allow my data to be used in future ethically approved research.

1. Do you understand and agree with the information above ? *

☐ Yes

Data collection

This part will collect your Twitter username with "@" prefix (e.g. @EdinburghUni) and the potential new tweets that you want to post.

2. Please input your Twitter username (e.g. @EdinburghUni) *

3. Please input 5 tweets that your want to post.

P.S. Please distinguish each tweet in an appropriate way (e.g. one line per tweet, blank lines, numbers, etc)

*

4. Do you want to provide your email so that we can reflect our experiments results to you ?

If you want to hear the feedback, please fill out your email address below.