Towards the Understanding of Private Content – Content-based Privacy Assessment and Protection in Social Networks

Qiaozhi Wang

Submitted to the graduate degree program in Department of Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Committee members

Prof. Bo Luo, Chairperson

Prof. Fengjun Li

Prof. Heechul Yun

Prof. Guanghui Wang

Prof. Prajna Dhar

Date defended: March 2, 2020

The Dissertation Committee for Qiaozhi Wang certifies
that this is the approved version of the following dissertation :

Towards the Understanding of Private Content
– Content-based Privacy Assessment and Protection in Social Networks

Prof. Bo Luo, Chairperson

Date approved: March 2, 2020

Abstract

In the wake of the Facebook data breach scandal, users begin to realize how vulnerable their personal data is and how blindly they trust the online social networks (OSNs) by giving them an inordinate amount of private data that touch on unlimited areas of their lives. In particular, studies show that users sometimes reveal too much information or unintentionally release regretful messages, especially when they are careless, emotional, or unaware of privacy risks. Additionally, friends on social media platforms are also found to be adversarial and may leak one's private information. Threats from within users' friend networks – insider threats by human or bots – may be more concerning because they are much less likely to be mitigated through existing solutions, e.g., the use of privacy settings. Therefore, we argue that the key component of privacy protection in social networks is protecting sensitive/private *content*, i.e. privacy as having the ability to control dissemination of information. A mechanism to automatically identify potentially sensitive/private posts and alert users before they are posted is urgently needed.

In this dissertation, we propose a context-aware, text-based quantitative model for private information assessment, namely *PrivScore*, which is expected to serve as the foundation of a privacy leakage alerting mechanism. We first explicitly research and study topics that might contain private content. Based on this knowledge, we solicit diverse opinions on the sensitiveness of private information from crowdsourcing workers, and examine the responses to discover a perceptual model behind the consensuses and disagreements. We then develop a computational scheme using deep neural networks to compute a context-free PrivScore (i.e., the "consensus" privacy score among average users). Finally, we integrate tweet histories, topic preferences and social contexts to generate a personalized context-aware PrivScore. This privacy scoring mechanism could be employed to identify potentially-private messages and alert users to think again before posting them to OSNs. It could also benefit non-human users such as social media chatbots.

Acknowledgements

I would like to express my appreciation and thanks to my advisor, my committee members, my colleagues, and my family for their guidance, support and encouragement along my PhD journey.

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Bo Luo for his generous support, patient guidance, and constructive advice on both my PhD study and my personal life. When I started my PhD program, he passionately inspired me concentrate on the field of security. Dr. Bo Luo always encouraged me to explore the possibility of using the latest computer science techniques to solve security problems. This spirit of exploration expedited my growth in research. His enthusiasm, and his rigorous attitude towards academia throughout our discussion over the years, established a good working life model for me. His insightful advice has always been valuable for me. I will always be grateful for his persistent guidance and approach to my research inquiries. His actions in their entirety allowed me to grow as a PhD student.

I am also deeply thankful to my committee members: Dr. Fengjun Li, Dr. Heechul Yun, Dr. Guanghui Wang, and Dr. Prajna Dhar. Thank you for offering brilliant comments and suggestions, which had great impact on my research direction and depth.

I would next like to recognize the invaluable assistance that all my colleagues at the University of Kansas provided during my study. These include Dr. Lei Yang, Dr. Yuhao Yang, Dr. Abdulmalik Humayed, Dr. Chao Lan, Dr. Hao Xue, Sana Awan, Sohaib Kiani, and Xiaomeng Su. It was an honor to work with a group of such intelligent and delightful individuals. They selflessly provided useful advice and assistance on me, and provided me with comfort during the difficult times.

Last but not least, I want to thank my family. They have always given me endless love and encouragement throughout my life. They have supported me in every way shape or form, and without them, I would never have experienced so many adventures and opportunities in my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Abstract

With the increasing popularity of online social networks (OSNs) like Twitter, we have observed large amounts of potentially sensitive/private posts being published to OSNs inadvertently or voluntarily. The owners of these posts may become vulnerable to online stalkers or adversaries, and they often regret their postings later on. However, the definition of sensitive information is subjective and different from person to person. Therefore, in this chapter, we will first discuss what is private protection on OSNs, why it is critical for OSN users, how the current researches address this issue, and how our research contributes to solving this problem, from a high point of view.

1.1 Introduction

In the wake of the Facebook data breach scandal, users begin to realize how vulnerable their personal data is and how blindly they trust the online social networks (OSNs) by giving them an inordinate amount of private data that touch on unlimited areas of their lives. Furthermore, social networks fundamentally encourage users to share their privacy to improve their presence in the virtual world. According to a report administered by Twitter, 500 million tweets are sent each day [106]. Thus, boundless amounts of private information are buried in the massive amounts of text format post. Human stalkers or automated bots can navigate/crawl through historic posts to re-assemble scattered pieces of sensitive information.

According to surveys, many individuals regret previous post on their social media platforms [118, 91]. The consequences of these posts are often not realized until the damage is already done and too late to mend. For example, the singer Justin Bieber would have unlikely been aware of the ramifications of his purported racial tweet as a 15-year-old. [30]. This phenomenon is quite common among younger teens. According to [58], 81 percent of parents and 79 percent of online teens report that "teens are not careful enough about giving out their personal information online." Even the most privacy-savvy users are likely to post something aggressive or divulge too much information. Even worse, for most of the users, their posts are only intended to be shared with friends/followers. However, the audience of OSNs is significantly larger than users' expectation which includes advertisers, recruiters, search engine bots, etc. Though users may have predetermined notions about their audience before posting tweets, imagining is difficult, and these notions are often inconsistent with the actual audience, as examined by Vitak *et al.* [110]. Moreover, Luo *et al.* and Yang *et al.*, [67, 124] used information theory methods to examine users' identifiability and quantified the amount of information leaked through user attributes from seemingly little and harmless data. When personal data about individuals are collected, processed, stored and retrieved without their consent, their information security is under threat. Many people are unaware of the fact that their privacy has already been jeopardized, and do not take action to protect their personal information from being used by others [7]. Therefore, it is critical to automatically identify potentially sensitive posts and alert users before they are posted, i.e., #DontTweetThis.

Additionally, friends on social media platforms are also found to be adversarial and may leak one's private information. Threats from within users' friend networks – insider threats by human or bots – may be more concerning because they are much less likely to be mitigated through existing solutions, e.g., the use of privacy settings [51, 99, 123, 107]. Thus, a mechanism to distinguish potentially sensitive/private posts before they are sent is urgently needed. Such a mechanism could also benefit non-human users such as social media chatbots. For instance, Microsoft's Twitter bot, Tay, started to deliver racist and hateful content soon after it was launched in 2016. Tay "learned" from inappropriate messages it had received. Unfortunately, there did not exist a mechanism to

assess the sensitiveness of tweets before they were exposed to Tay or posted by Tay.

1.2 What is privacy?

In order to protect privacy, we need to understand the concept of privacy. Privacy can be defined from many different domains, including the rights of citizens, or political policies. The privacy law itself even defines privacy from several different aspects. In short, privacy is not just about hiding things, it is about self-possession, autonomy and integrity [31]. However, when referring to the area of social networks, the private versus public boundaries are quite ambiguous. The privacy paradox phenomenon is very common among social network users, which has been investigated by Tufekci [100]. Tufekci found there was no relationship between users' privacy concerns and their level of disclosure on social networks. Even the users who expressed many privacy concerns disclose large amounts of personal information. Therefore, the concept of sensitivity and privacy on social network is unclear among users.

Furthermore, the degree of sensitivity and privacy is a subjective perception which differs from person to person. For instance, some users are more conservative about health-related issues, while others might be more protective on work-related information. Even though some users feel that certain topics are sensitive, such as obscene content in posts, they may treat them to different degrees of sensitiveness. That said, in developing a privacy protection mechanism for online social networks, we cannot use a uniform measure of privacy for all users. Besides, users' perception of privacy varies from time to time. For example, although political attitude is thought to be private normally, there were one billion tweets about the election during the 2016 election season. Under this occasion, the political content would not be as sensitive when compared to a non-election season.

Therefore, we can see that to understand the perception of privacy from the online social network users' perspective, we should consider the different meanings of privacy from different users and consider to what extent it matters to them. The problem of privacy protection is dependent on users, and affected by the social context, which should be customized.

1.3 Existing Privacy Protection Methods

Conventional privacy protection mechanisms on data or OSN (e.g., *k*-anonymity [97], differential privacy [23]) mainly focus on the protection of individuals' identities. They are not suitable for social networks. First, most of the OSN data is unstructured text, while traditional methods are designed for structured data such as demographic and genetic datasets. More importantly, conventional privacy protection mechanisms mainly focus on protecting users' *identities* or *private attributes* [97, 23, 75, 13, 40, 68, 124]. However, according to a survey in [42], only 0.1% of users mentioned identifiable attributes such as email addresses or phone numbers in their tweets. Therefore, leaking identities or identifiable attributes during normal socialization is not the only privacy concern in OSNs. On the contrary, since the offline identities of OSN users are often known to their online friends, especially in strong-tie oriented OSNs such as Facebook, *sensitive or inappropriate content* is truly at risk due to careless or unintentional disclosure during socialization. The social network user is vulnerable with social media which has powerful *broadcastability*. It would be unknown who would access user's published content, or more importantly, what their intentions with this content would be. Thus, a *user-centered* protection mechanism is more meaningful for the social network privacy protection.

There are also many fine-grained tools or control schemes for protecting personal profile data in social media, including the "Privacy Wizards" [25] and the research project proposed by Lipford *et al.* [62]. These projects contribute a great deal in optimizing user privacy settings to enhance access control of profile data. However they all ignore an important factor – content which includes more personal information. As we know, examining posts over a long time interval from a single user could expose more habitual, personal information. Content-access control tools such as "Twitsper [89]" are also well-researched, but they all have the assumption that the user is aware of what is private or not. In fact, privacy is quite a blurred concept, and users themselves can have difficulties defining privacy as definition changes overtime [54].

Therefore, we argue that another key component in privacy protection in OSNs is protecting *sensitive/private content* beyond the protection of identities and profile attributes, i.e., *privacy as*

having the ability to control the dissemination of sensitive information. Here, the dissemination should not only regard the access to content, but also take into account whether such content would result in regrettable or negative outcomes for users in the future.

1.4 Contribution

The ultimate goal of this dissertation is to provide a user-tailored privacy score based on context, in which, the context depends on the sensitiveness of the topic at a specific time. Our research furthers the goal of designing a useful mechanism beginning with the question, *what content might be privacy related, and what defines private content from the users' perception*? We summarized 13 topics that might be privacy related and built a semantic model to identify the different topics, which benefited the process of personalization. Next, based on these 13 categories of content, a user study of common perception of privacy on OSNs was launched. With the understanding of the OSNs user study, we converted the human cognitive model into a computational model which can capture user's privacy perception in general terms. Considering the difference of human cognition, this general model was further adjusted by user's privacy preference and social context. The main contributions of this paper are the following four-fold.

- We explicitly research and study topics that might result in eventual user's regret while also analyzing private content that may be unbeknownst to the user. Based on this knowledge, vast amounts of tweets are extracted, processed and analyzed. We make the first attempt to classify potentially sensitive tweets into a comprehensive set of likely sensitive categories. Examples of these categories might include drugs and alcohol, family information, etc. The classification model is built with both semantic features and users' topic-preferences, which boosts the accuracy, comparing with the models purely based on the semantic features.
- We launch a crowd-sourcing survey on Amazon Mechanical Turk and collect the privacy perceptions from a diverse set of users. Through examining the consensuses in the responses of the sensitiveness of content, the survey gives us primary insight to the evaluation of the com-

mon perception towards content sensitiveness/privacy for average users in a neutral context. This is the foundation of developing our privacy protection mechanism.

- We make the first attempt to develop a computational model for quantitative assessment of content sensitiveness using deep neural networks. The context-free privacy score resembles the "consensus" perception of average users on the purely textual content. This will be shorted as *PrivScore* in this dissertation. *PrivScore*, to the best of our knowledge, is the first quantitative assessment for sensitive content. It has the potential to be utilized in various applications.
- We further integrate social contexts and topic-specific personal privacy attitudes to extend the predictive model to generate a context-aware score – adjusted based on the societal context and personalized privacy scores – adjusted based on personal preferences. The adjusted models make it possible for the user to protect sensitive content, to an extent, without affecting his/her normal socializing.

These results are published on [113, 114, 115].

1.5 Organization

The rest of the dissertation is organized as follows: Chapter 2 formally introduce the background knowledge and related works, to give readers a basic understanding of what we are doing. Then an overview of our proposed solution will be described in Chapter 3, which can make the previous abstract introduction becomes more specific. Chapter 4 focus on the topic classification of sensitive tweets. In Chapter 5, we will introduce our *PrivScore*, the context-aware, text-based quantitative model for privacy information assessment in detail. We finally conclude the dissertation in Chapter 6.

Chapter 2

Background and Related Work

Abstract

In this chapter, we will begin with the discussion of privacy in OSNs from three main angles: privacy modeling, protecting user identities, and preventing unauthorized access to private data. Then, based on our argument that *privacy as having the ability to control the dissemination of sensitive information*, the related work of regret tweet and content-based privacy protection are introduced. Meanwhile, since text classification and text understanding methods are intensively used in content-based protection, the background knowledge of natural language processing (NLP) is also described briefly in this Chapter.

2.1 Privacy on Online Social Networks.

Compared with traditional user data such as age and location, which can be easily qualified or quantified, social networks have more valuable and private information. This information lies in much more complicated data formats, such as text and photos, which are also much harder to extract and understand for computers. In addition, this content on social networks is always completely open to the Internet as a whole, especially if the user did not work on the user-unfriendly, implicit private settings. Therefore, the privacy protection of OSNs has been attracting more and more researchers across borders. Researchers analyze different angles to these problems and introduce many conceptual and technological methods to address users' privacy concerns. Existing research on social network privacy primarily focuses three directions: (1) privacy modeling, (2) protecting user identities and (3) preventing unauthorized access to private data.

2.1.1 Privacy Modeling

The purpose of privacy modeling is to understand the OSNs users' privacy perceptions, attitudes, behaviors and expectations. Fogel *et al.* [29] studying, among other things, risk taking, trust, and privacy concerns regarding social networking websites, found some interesting results. These findings include that men partake in greater risk with potential sensitive information than women, and general privacy concerns are of greater concern to women than men. Another important variable to consider is the highly studied "Privacy Paradox", which focuses on an individual's claim to care about their privacy while simultaneously providing a great deal of personal information through social media [39, 7]. In addition, a meaningful research is performed by Schomakers *et al.* [33]. They analyze the German users' perception of sensitivity and compare it to the results from US and Brazil. This research shows that there is a consensus on what constitutes sensitivity across nations, which provides theoretical basis for our research. However, many of these researches employ user studies to examine the factors that influence the privacy models, such as age, gender, culture, education level, etc [88, 63, 9, 59].

2.1.2 Protecting User Identity

Most previous research in user identity protection primarily focused on anonymization. Take *k*-anonymity [97] for example, which prevents joining attacks by suppressing portions of the released micro-data, so that there is no unique individual in at least *k* candidates. Yet, with some background knowledge or sensitive attributes, *k*-anonymity could be cracked. In succession, more sophisticated privacy-enhancing techniques methods, like *l*-diversity [69], *t*-closeness [60] and differential privacy [23] are developed to sanitize the dataset before publishing. However, they have also shown to be vulnerable against several re-identification attacks, e.g., [1, 37]. Meanwhile, these database anonymization methods are not applicable in online socialization in two instances. First, the data on social networks is not formatted or quantified as that in the database, and information is buried in more complicated ways like text or picture. Second, a portion of friends in a user's online social contacts are also personal friends offline. This means they have already known user's profile

information, and can access profile, posts, etc.

2.1.3 Preventing Unauthorized Access to Private Data

Preventing unauthorized access to private data is to anticipate and neutralize abuse before user's have future regret from posted data. Researchers in this area focus on privacy configuration and control for OSNs. Access control frameworks such as Persona [5] and EASiER [48] have been developed. Hummingbird [19], a privacy-enhanced variant of Twitter, encrypts pre-defined tweets in a way that only authorized followers can decrypt tweets of interest. Arcana [78] presents a fine-grained access control system for OSN content sharing through employing Ciphertext-Policy Attribute-based Encryption (CP-ABE). However, these systems are based on a key assumption that the private information is already known, which requires the user to explicitly define what is private and what type of protection is needed. Moreover, since different users may have different privacy definitions and expect different levels of protection, these approaches are too rigid to address the varying needs of different users. Liu *et al.* proposes computing privacy scores based on the uniqueness and visibility of information [65]. However, this work is still quite different from our approach both on input data and applied algorithms. We utilize only textual content instead of profile information and structure of social network. Since Liu's article, more researches about privacy scores have arisen. David Pergament *et al.* provides a system called FORPS [82] which calculates friends-oriented reputation privacy scores based on topics, object types and behavioral factors. These approaches take user profiles and network structures as input to learn private information types and protection requirements that are implicitly expressed by users. However, they still require the users to be "conscious" about what to protect.

Different from these schemes, our privacy scoring system aims to autonomously assess the scale of privacy or sensitiveness from the content shared by the user. The private content identified by our scheme and the corresponding privacy scores can be fed into privacy configurations and control systems discussed earlier which benefits their designs. Evaluating the sensitiveness of content, and letting users contemplate before posting or posting with access control will enhance user's

privacy awareness, reduce information abuse, and create a favorable social network environment.

2.2 Regret Post and Content-based Privacy Protection

The seemingly harmless and short posts can contain a large amount of information about users, especially through examining posts over long-time intervals. There are countless amounts of sensitive information that can be extracted from posts such as gender [66, 17], location [61, 44, 15, 14], home [70, 83], socio-demographic and socio-economic status [84, 111, 55], etc. Research conducted on private/sensitive post content on OSNs can be classified into two categories: private post modeling and private post identification and classification. We will introduce the related work of these two categories later in the paper. Most of this research utilizes the data from Twitter – tweets, due to the openness of Twitter. Twitter users can view anything, and everything posted from another Twitter user. This information can also be viewed from anywhere, and in present time. Twitter provides an easily accessible place for researchers to understand what users are discussing. Therefore, our research is also based on the data from Twitter.

2.2.1 Content-based Private Post Modeling

A significant body of research attempts to model (i) *regret tweets* and (ii) *privacy leakage* from various angles such as causes and cognitive models, cultural influences, and possible mitigation, etc [122, 126, 118, 77, 80]. For (i), large-scale user studies have been conducted to analyze the psychological and social perspectives of regret posts in OSNs [91, 118]. Through user studies, they find the types of regrets most shared by users, which include sensitive content, criticism, private information, etc. The cause of posting and regret are usually due to the negative emotions, underestimated consequences, unperceived audiences, etc. The regrettable posts could lead to some unexpected and unfavorable consequences, such as the ending of relationships or losing a job. Thus, they suggest building some content-based private post identification systems to alert users. However, they did not specify how such mechanisms can be developed. Similarly, for (ii), Mao *et*

al. [71] models three specific types of privacy leakage to identify the nature, cause, and influencing factors (culture) of such leaks. In both (i) and (ii), studies are followed up to model and examine private and/or regrettable posts and user perceptions [122, 121, 21] towards regrettable content. In the final part of these studies, the damage control of undesired disclosure of private/regrettable content is also discussed. For instance, [118, 91] identify repair strategies for regret posts such as deletion, apologise, excuse and justification. Presently, in the wake of topics concerning user's online privacy, more users realize that they have revealed too much information and have attempted to withdraw their tweets. This has triggered researchers' interest in investigating the privacy of deleted posts. For example, Mondal *et al.* [76] shows that a significant amount of information regarding deleted tweets, or even characteristics from the deleted accounts, could be recovered from the traces of residual activities, i.e., the replies of the deleted tweets or accounts. Age-based and inactivity-based deletion of data have been proposed in the literature [125, 76] and adopted in commercial OSN platforms, such as Snapchat, Dust, 4chan, WeChat. Recently, Minaei *et al.* [74] identifies the phenomena of tweet deletion attracting unwanted attention from stalkers. In defense, a temporary withdrawal mechanism is developed to offer deletion privacy, so that adversarial observers could not (confidently) identify users deleting their tweets.

2.2.2 Content-based Private Post Identification

The qualitative modeling of private/regrettable posts provides a theoretical basis for the automatic tweet assessment, and the automatic assessment of private/regrettable posts has drawn evermore attention from researchers. *Liu et al.* [65] proposes a framework to compute the privacy score of a user. They utlize tweet owners' attitudes, and rarity of information to calculate this privacy score. In [71], Mao *et al.* develops one of the first mechanism to identify potentially private tweets. In the paper, tweets are first filtered into three topics via keyword matching. Then, they design a classifier for private tweets using naive Bayes and SVM classifiers. Inevitably, the features and classifiers need to be fine-tuned for each category, and they only achieve approximately 80% accuracy in binary classifications. This mechanism demonstrates how the identification of sensitive tweets should

be based on different and varying topics, and these results correspond with our motives. However, we believe that by including more topics, this can have the potential of the program containing further amounts of sensitive content. Due to the varying lengths of posts on online social networks, including brief messages, and the heavy usage of slang, the information extracted, and the identification of posts can be far more difficult to obtain than longer documents, such as Wikipedia. To tackle the challenges in handling short text in Twitter, [47] aggregates tweets for each user, extracts topic matching, NER and sentiment features, and uses AdaBoost with Naive Bayes to classify each user into categories labeled as privacy score 1, 2 and 3. Recently, Zhou *et al.* proposes to examine the features of deleted tweets to predict the likelihood of a tweet being deleted [126]. They preselect ten topics that are commonly considered as sensitive (e.g., curses, drugs, etc.), and classify tweets as being sensitive or non-sensitive by analyzing if the tweet contains keywords from the sensitive topics. Existing private tweet identification/classification approaches employ term-based features (BoW, TF-IDF, sentiment) and simple supervised classifiers, which cannot capture semantic features, or accurately detect topics containing subtle yet sensitive content. Furthermore, the classification approaches only generate a binary notion on whether a tweet belongs to a predefined category. They also fail to quantitatively measure the levels of sensitiveness, which is another goal we want to achieve from this work.

Our project is primarily motivated by private post modeling, which calls for methods to assess private/sensitive tweet content, so that users can be alerted accordingly. While we have been inspired by existing methods in private post identification, our approach is novel in the following aspects: (1) We employ state-of-art content representation and classification algorithms (word embedding and RNN) to significantly improve the accuracy of assessing general tweet content. This approach does not rely on keyword matching, which significantly improves the efficiency through considering the semantic similarity among different words, whether these words are in the keywords list or not; (2) Instead of a binary notion of sensitive vs. nonsensitive, we generate a real score that provides more information on the *level of sensitiveness*, and enables personalization in setting alerting preferences. (3) We have developed a general purpose solution that works for a

broader scope of tweets, instead of only identifying specific types of private/sensitive tweets.

2.2.3 Topics Related to Regret Posts

In order to implement personalization in our mechanism, accurate topic classification is expected. To the best of our knowledge, our project makes the first attempt to automatically classify microblog messages into a comprehensive set of potentially sensitive categories. In this part, we assume that a set of potentially sensitive tweets, with controversial or private content, has already been identified. Our goal is to properly classify these sensitive tweets into 13 pre-defined categories: *Health & Medical, Work, Drugs & Alcohol, Obscenity, Religion, Politics, Racism, Family & Personal, Relationship, Sexual Orientation, Travel, School life, Entertainment*. The reason we did not use cluster to find these 13 topics is because the topics in tweets are broad and sparse, and the occurrence frequency of some extremely sensitive tweets can be low such as racist. To overcome these deficiencies, we manually list these topics. These 13 topics are selected because some of them are primary sensitive topics, such as drugs, racism and sexual orientation. Some of the topics might cause controversial discussions or thoughts on Twitter, like religion and political attitudes. Some seemingly harmless topics, such as family information, can also compromise a user's privacy. For example, the tweet about celebrating a user's mother's birthday might be associated with the security question of Twitter owner's bank/shopping accounts. Meanwhile, a user's travel plan, from another perspective, is a prediction of the user's absence from home. Sometimes, users also want to have access control functions for their tweets. For example, the tweets about going to a bar is not intended to be shared with their bosses or teachers. Chapter 4 will describe this part in detail.

2.3 Text Classification

Automatic text classification is the process of assigning a class label given a text document which is tweet in our research. Both the topic classification problem and privacy score algorithm can be

treated as text classification problems. To begin with the text classification problems, we should know that text modeling is messy, and machine learning algorithms cannot work with raw text directly. In order to enable machines to understand text, the text need to be converted into numbers, or vectors of numbers, which can be treated as feature extracting. In addition, the length of inputs and outputs should be fixed. The common and traditional methods for feature extracting of content includes the Bag-of-Words and TF-IDF, which makes the feature-space exceedingly sparse. Occasionally, principle component analysis (PCA) or single value decomposition (SVD) are used to decrease feature-dimension, though these algorithms themselves also suffer from the high dimension curse. In addition, these methods only consider the occurrence of words instead of the sentiment meaning which is the most important linguistic element in content understanding.

Deep learning has fundamentally changed the landscapes of text classification. The appearance of word embedding which uses condensed vectors to represent each word is a breakthrough in natural language processing (NLP). Pre-trained context-free word embedding methods, like Word2Vec and GLoVe, makes utilizing neural network for text classification feasible, and allows us to capture relationships in language that are very difficult to capture otherwise. Along with the combination of the neural network – Long Short Term Memory (LSTM) which has the ability to accumulate increasingly richer information as it goes through the sentence [79], this supervised deep learning method is widely adopted in text classification by both academia and industry. Recent research has increasingly focused on unsupervised learning, semi-supervised learning, and transfer learning to bridge the gap between the enormous amount of unannotated text on the web and the shortage of training data in task-specific datasets. The most successful research into transfer learning, BERT [20], is a deeply bidirectional, unsupervised language representation, contextually pre-trained using only a plain text corpus – Wikipedia. This pre-trained model can be fine-tuned on small-data NLP tasks, and obtain decent results when compared to training on datasets from scratch. These advantages make it extremely popular after its release.

In our work, for the topic classification portion, we utilize traditional feature extracting methods to obtain content features. These methods will be introduced in Chapter 4. The classifiers we

selected for this section will again be traditional ones, such as Naive Bayes. However, the problem of privacy assessment is a more complicated task, which is difficult to be solved by these traditional methods. Considering the advantage of deep learning neural networks, we leverage the word embedding method – GLoVe and Long Short Term Memory (LSTM) network to train a model for privacy assessment part. The evaluation metrics show the substantial improvements compared to the traditional methods. The further measurements on different kinds of datasets demonstrate the effectiveness of this method. We will subsequently explore the usage of state-of-the-art pre-trained model – BERT on our project, which will reveal the advantage of the model on solving the shortage of training data.

2.3.1 Related Work of Tweet Classification

Since Twitter is an ideal platform with numerous accessible tweets covering every aspect of daily life, tweets classification attracts lots of researchers to study. However, the classification of tweets is challenging due to the length of the tweet – up to 140 characters and the heavy usage of slang. Traditional text classifiers based on vector space model typically use BoW or TF-IDF features. In tweet classification, more features from the application domains are extracted to boost the classifier. For example, Lee *et al.* [57] classifies tweets about trending topics into 18 more general trending topics, such as sports, politics, and technology, which can make people understanding these topics easier and improve the performance of information retrieval. They integrate network information with text features to classify tweets into trending topics, which improves the accuracy of the result. Takemura *et al.* [98] separates tweets into three kinds: 'expired', 'going to be expired', 'would not be expired', in which 'expired' means the information value has vanished when users read them. To filter out "expired" tweets and saves users from outdated ones, some time-related features like the existence of bursty words and time expressions which are extracted from tweets to train the classifier. Sriram *et al.* [94] introduces eight features which extracted from the content of author's profile and tweet history. Their method outperforms the traditional BoW-based approaches on classifying generic topics–news, sports, etc. Vosoughi *et al.* [112] developed a sys-

tem to identify rumors on Twitter about real-world events. They first use syntactic and semantic features of tweets to determine if a tweet is an assertion or not. Then, if the tweet is an assertion, the hierarchical agglomerative clustering method is used to identify if an assertion is a rumor. Liu *et al.* [64] propose that user-posted content could implicitly reveal users' location context. And they adopted time-period and users' check-in history as boosting features to classify "check-in" tweets to different locations. Although inspired by these approaches, our purpose and application domain are completely different from them. In addition, our work focuses on tweets' sensitiveness, which is a very subjective perception whose cognitive process is not clearly understood yet.

2.3.2 Traditional Approach for Text Classification

2.3.2.1 Naive Bayes

Naive Bayes is a wildly used machine learning algorithm in text category [86]. The motivation of the algorithm can be described as, if each tweet is treated as a document d and d is composed of a bag of words $w_1, w_2, \ldots, w_n$, then the posterior probability that the tweets belongs to category c can be demonstrated as

$$p(c|d) \propto p(c) \prod_{1 \leq k \leq n_d} p(w_k|c)$$

In this expression, $p(c)$ is the prior probability of a tweet occurring in class c, defined as the number of tweets in category c divide the total number of tweets in training set. $p(w_k|c)$ is the conditional probability of words distribution in category c. The tweet is assigned to the best class determined by

$$\arg\max_{c \in C} p(c) \prod_{1 \leq k \leq n_d} p(w_k|c)$$

2.3.2.2 Bag-of-Words (BoW)

By using Bag-of-Words (BoW) model, a tweet can be treated as a bag containing all the words appearing in the tweet, disregarding grammar or order [3]. For example, both "John likes Mary" and "Mary likes John" can be represented as $\{"John'', "likes'', "Mary''\}$ in the BoW model. This

method simply uses all words as features to represent each tweet, which will make data set very sparse, and reduce the classification accuracy.

2.3.2.3 Term Frequency-Inverse Document Frequency (TF-IDF)

Comparing to BoW, tf-idf can reduce feature dimension effectively and distinguish the importance of different words [27]. TF-IDF is short for term frequency-inverse document frequency, which is intended to reflect the importance of a word to a document in a corpus. This scheme gives the word w in the document d the weight as

$$TF\text{-}IDF(w,d) = TermFreq(w,d) \cdot \log(N/DocFreq(w))$$

where $Weight(w,d)$ is the frequency of the word in the document, N is the number of all documents, and $DocFreq(w)$ is the number of documents containing the word w.

2.3.2.4 Principle Components Analysis (PCA)

Principle Components Analysis (PCA) is one of techniques for taking high-dimensional data, and reducing the dependencies between the variables, to represent it in a low-dimensional form, without loosing too much information [49]. PCA is a widely used simplest and robust dimensionality reduction approach.

We assume the PCA starts with p-dimensional feature vectors and we want to summarize them by projecting down into a q-dimensional subspace. To find the projection, one must minimize the correlation (redundancy) and maximize the variance. Thus, the computing process covers standard deviation, covariance, eigenvectors and eigenvalues, which will be described in more detail in the corresponding chapter.

2.3.3 Text Classification With Deep Learning

2.3.3.1 The GLoVe Model

The Global Vectors for Word Representation (GLoVe) [81] word embedding algorithm leverages the global word co-occurrence statistics in the training set, and the vector space semantic structure captured in Word2Vec. It represents an aggregated global word-word co-occurrence matrix as $\mathbf{X}$, in which the element X_{ij} denotes the number of times a word j occurs in the context of the word i. The soft constraints for each word pair is defined as:

$$w_i^T \tilde{w}_j + b_i + \tilde{b}_j = logX_{ij} \tag{2.1}$$

where w_i and $\tilde{w}_j$ are the main and context word vectors, and b_i and $\tilde{b}_j$ are scalar biases for main and context words. To avoid weighing all co-occurrences equally, GLoVe adopts a weighted least squares cost function:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 \tag{2.2}$$

where $f(X_{ij})$ is the weighting function in the form of:

$$f(X_{ij}) = \begin{cases} (X_{ij}/X_{max})^{\alpha} & if\ X_{ij} < X_{max} \\ 1 & otherwise \end{cases} \tag{2.3}$$

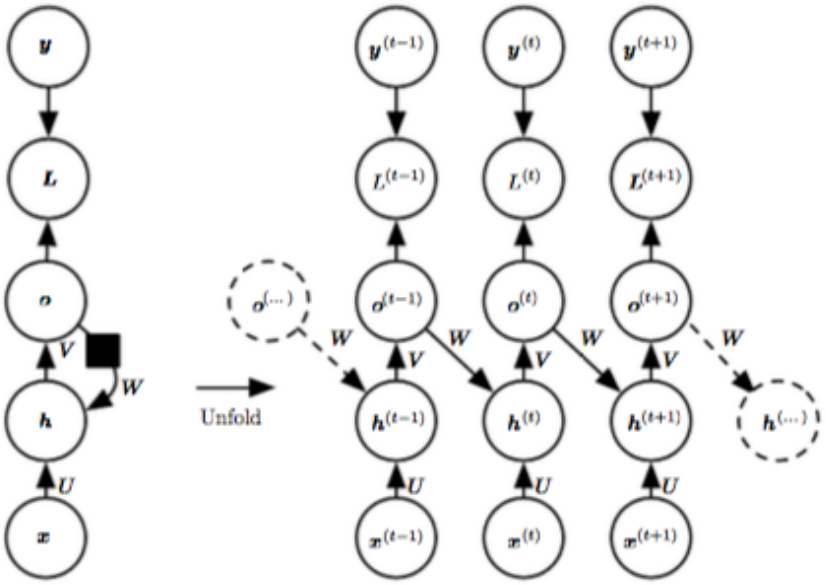
The model generates two sets of word vectors, $\mathbf{W}$ and $\tilde{\mathbf{W}}$. Since $\mathbf{X}$ is a symmetric matrix, $\mathbf{W}$ and $\tilde{\mathbf{W}}$ are equivalent and differ only as a result of their random initializations. Therefore, the sum $\mathbf{W} + \tilde{\mathbf{W}}$ is used as the word vectors to reduce overfitting.

2.3.3.2 Recurrent Neural Network (RNN)

The Recurrent Neural Network attracted much attention especially for natural language processing over the past few years, due to its advantage in processing sequential information. This is because the architecture of RNN is a class of neural network whose identical units connect together, like a

circle by itself, as shown in Figure 2.1. In the book "Deep Learning" [56], it gives us a deeper and clearer explanation, which says parameter sharing makes it possible to extend and apply the model to examples of different forms and generalize across them. If we had separated parameters for each value of the time index, we could not generalize to sequence lengths not seen during training, nor share statistical strength across different sequence lengths and across different positions in time.

Figure 2.1: Architecture of Recurrent Neural Network



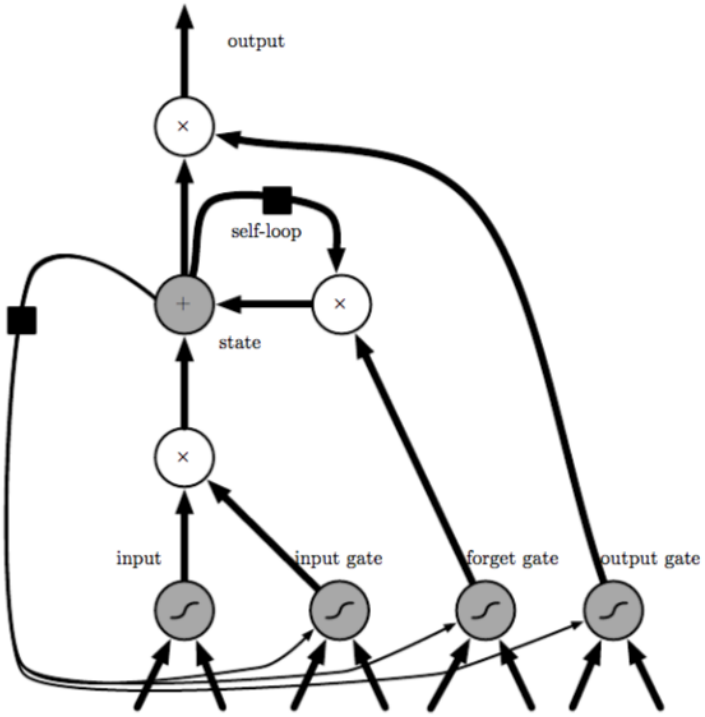
However, the repeated training of the same parameters also brings some problems. During the backpropagation, the gradient is passed back through many time steps; it tends to grow or vanish. To overcome this shortcoming, long short term memory (LSTM) is a widely used practical approach, which is a kind of gated RNN. The LSTM will be introduced in the following section.

2.3.3.3 Long Short-Term Memory (LSTM)

LSTM is introduced to overcome the issue that RNNs suffer from the problem of vanishing gradients, with the help of a special designed *memory cell* [8]. The structure of LSTM is shown in Figure 5.13. From the structure we can see, LSTM has several critical components compared with the RNN, including the states, and three gates – input gate, forget gate and output gate. These components control the flow of information. Therefore, LSTM can learn to memorize long time

dependency if necessary and can learn to forget the past information if needed, which can avoid the gradient vanishing/exploding and become more similar with the natural language processing procedure. Due to limitation of page number, the detailed forward-propagation and backward-propagation computing process will be introduced in Chapter 4.

Figure 2.2: Structure of LSTM



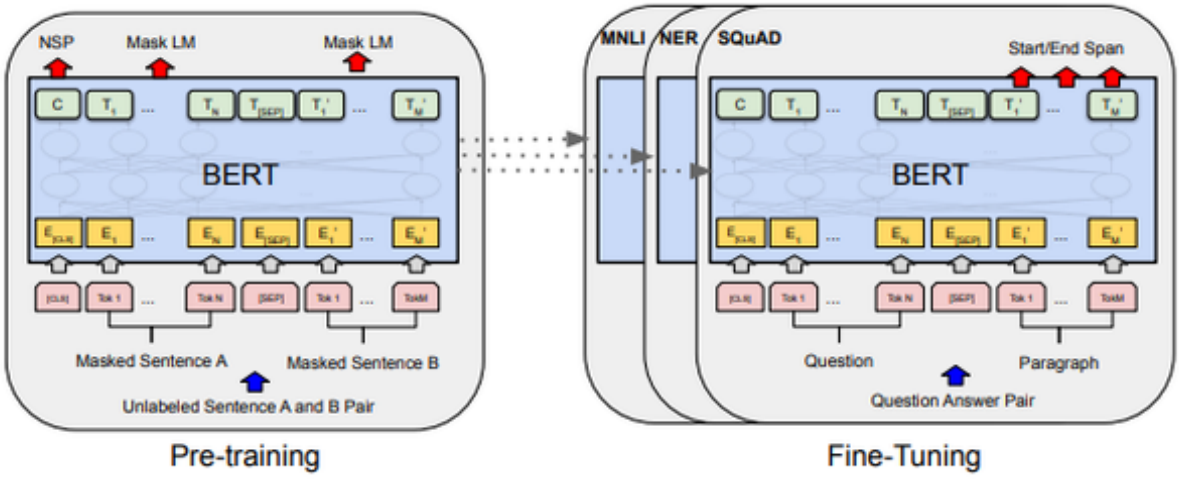
2.3.3.4 Bidirectional Encoder Representations from Transformers (BERT)

Before BERT, standard language models are unidirectional, either a left-to-right structure or a shallow concatenation of independently trained left-to-right and right-to-left language models (LMs) [20]. Unlike these models, Bert uses a "masked language model", which randomly masks some of the tokens from the plain text input, and outputs the prediction of the original vocabulary id of the masked word based only on its context. This mechanism makes the fusion of the left and the right context information possible.

The overall pre-training and fine-tuning procedures for BERT is shown in Figure 5.16. BERT's architecture is a multi-layer bidirectional self-attention Transformer encoder. The Transformer is

based on the original implementation described in [109]. The framework of Bert includes two steps: pre-training and fine-tuning. For the pre-training part, it's a unsupervised learning process, in which model is trained on enormous amount of unlabeled plain text. For fine-tuning, the BERT model is first initialized with the pre-trained parameters, then all of the parameters are fine-tuned/updated using labeled data from the downstream tasks. More details about utilizing Berts in our project will be described in Chapter 5.5.4.

Figure 2.3: The overall pre-training and fine-tuning procedures for BERT



Chapter 3

Overview of the Proposed Solution

Abstract

In this chapter, we propose a mechanism for protecting users' content privacy. We begin with an analysis of threat model and objectives of this mechanism. Then, based on these objectives, we propose a three-phrase privacy scoring model: context-free privacy scoring, context-aware privacy scoring, and personalized privacy scoring. These three elements are discussed briefly in this section. The expected outcome and the anticipated impacts of the proposed solution are also analyzed in this chapter.

3.1 The Problem and Challenges

A significant amount of research has been done on identity privacy. However, in social networks, offline identities are either known to friends/followers, or hidden to strangers through global privacy settings. Meanwhile, sensitive/identifiable attributes, such as SSN, phone numbers and emails, are seldom mentioned during online socialization. Therefore, before proposing the solution, the threat model and the objectives of our mechanism have been clarified below.

3.1.1 Threat Model.

In this work, we aim to protect social network users (and chatbots) from accidentally disseminating *any type of inappropriate content*, especially the private or sensitive information about themselves. We broadly classify the risk of inappropriate dissemination into two types of audiences: (1) fol-

lowers or friends (insiders), those who receive updates of the user's posts; (2) stalkers or strangers, those who peek into a target user's social network posts. Both types of audiences are likely to know the offline identity of the user. We do not focus on protecting identities or attributes (e.g., location), since they have been intensively studied in the literature. We do not block the user from publishing the (sensitive) content or block the receiver from viewing the content. Instead, we provide an alert to assist users' decision making. We assume the adversaries can browse the OSN through the user interface, or collect data using an automated crawler through the OSN's API, i.e., there is no hacking or phishing. Finally, we do not consider the retraction of previous posts. If user later regrets doing so, the damage has been done and it is almost impossible to completely erase the posts and the consequences.

3.1.2 Objectives.

The objective of this work is *to develop a computational model to quantitatively assess the level of privacy/sensitiveness of unstructured text content, and this will be further adjusted to reflect the impacts of societal contexts and personal privacy attitudes*. We make the first attempt to generate a *privacy score* (PrivScore) for each short text snippet, e.g., a tweet (limits to 280 characters), to reflect its level of sensitiveness within its societal context. The privacy scoring mechanism is expected to serve as the foundation for a comprehensive privacy monitoring and user alerting solution.

This research is exceedingly challenging due to several factors: (1) privacy or sensitiveness is a very subjective perception [22, 43]. Due to the peculiarity, complexity and diversity of human cognition, it is difficult to precisely capture the privacy calculus model for each individual, and generate a consensus privacy score that is unanimously agreed by all users. (2) Text understanding and natural language processing is still an area with active ongoing research. In particular, modeling and understanding of unstructured, short, and non-standard text snippets, such as microblogs, is exceedingly difficult. (3) The subjective perception of privacy is often motivated by many different aspects such as, personal privacy attitude, emotions, societal context, culture, etc. The complexity

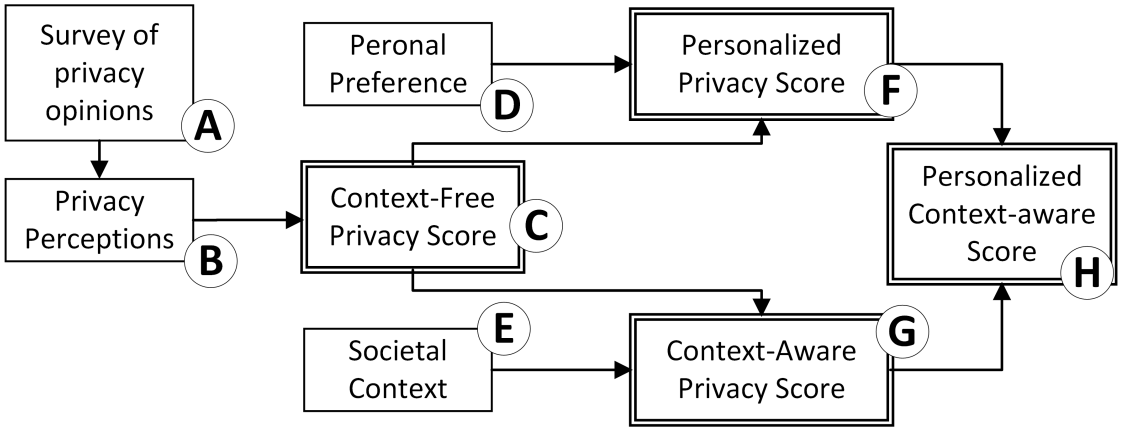


Figure 3.1: Key components of the three-phase privacy scoring model.

of modeling the influencing factors is also problematic.

3.2 Overview of the Proposed Solution

We categorize the personalized privacy score into two sub-questions – the identification of sensitive tweets and the classification of sensitive topics. The reason behind this categorization is that we consider *private tweet identification* (i.e., automatically identify if a tweet contains private information) and *private tweet classification* as dual-problems. Progress towards one category will eventually benefit the other. After this, we can obtain a topic classification model used for customized privacy protection, and a widely-accepted sensitive content identification model.

To determine the mechanism of the privacy score, we propose a three-phase privacy scoring model based on (Fig. 3.1): (1) context-free privacy scoring, (2) context-aware privacy scoring, and (3) personalized privacy scoring.

1. The *context-free PrivScore* is an autonomous assessment of the degree of sensitiveness of short, unstructured text snippets based purely on the textual content, i.e., free from its context. Our goal of the *PrivScore* is to identify sensitive tweets based on widely accepted opinions concerning sensitiveness. We do this by first collecting data regarding different users' opinions pertaining to the level of sensitiveness of potentially private content through a user survey on Amazon Mechanical Turk (MTurk) (Fig. 3.1 (A)). We then statistically analyze the responses to identify the consensus

of human perceptions (Fig. 3.1 (B)). Based on the results, we then develop a deep learning model which is realized by the cutting-edge NLP methods, like word embedding with long-short term memory, or Bert, to model text content to develop a privacy scoring mechanism (Fig. 3.1 (C)). Ultimately, the context-free privacy scoring mechanism takes potentially sensitive text snippets and generates a score that reflects the consensus of sensitiveness from the majority of users.

2. In *Context-aware privacy scoring*, we model the influence of the societal context and incorporate it into PrivScore. We had observed that privacy perceptions are dynamic and are influenced by societal contexts. Particularly, we notice that when popular topics trigger significant interests and/or discussions on OSNs, users became less concerned about its sensitiveness. For example, the political attitude is normally considered private. When one billion tweets were posted regarding the election during the midterm elections in 2018, the degree of sensitiveness of political content implicitly decreases from the non-election periods. The context-aware PriScore model measures the influence of societal context using the volume, duration, and relevance of trending topics (Fig. 3.1 (E)). We integrate this into context-free PrivScore to reflect the societal influence on privacy perceptions (Fig. 3.1 (G)).

3. The *Personalized privacy score* adjusts PrivScore for each user with a personalized topic-specific attitude. We recognize that the privacy perception is subjective, and differs for each user. Users have varying levels of tolerance on private information disclosure on different topics. Therefore, we treat the personalized privacy score as an additional step, combining the context-free privacy score with the classification of sensitive topics. This will be further explained in detail in Chapter 4. Progress towards *private tweet classification* will eventually benefit the scoring of personalized privacy. Moreover, individual privacy perception is also shaped by various psychological factors such as personality and emotion [77]. To provide privacy alerts that are customized for each user, we first analyze the activity history to discover the topic-based privacy attitude (Fig. 3.1 (D)). A personalized privacy scoring model is then developed to integrate personal attitudes into a context-free PrivScore (Fig. 3.1 (F)).

Sequentially, we develop a computational model for a *personalized context-aware PrivScore*

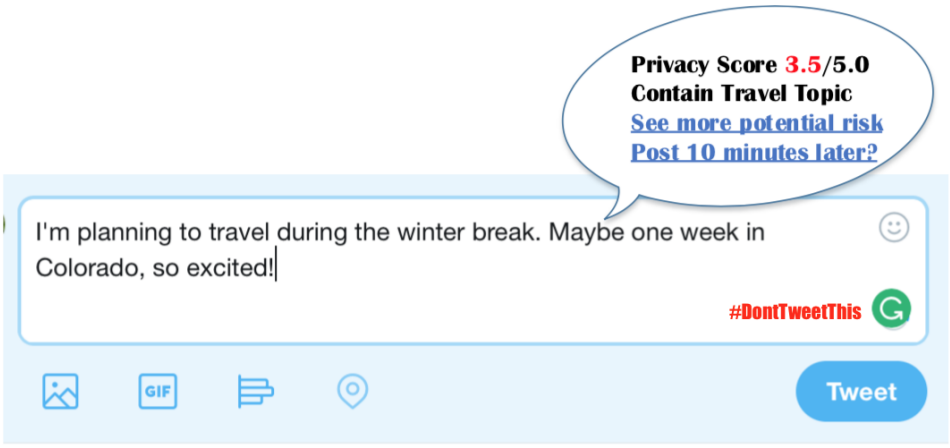
(Fig. 3.1 (H)). A PrivScore is generated for each social network post (such as a tweet) to reflect a quantitative assessment of the estimated sensitiveness. This scoring mechanism could be adopted for individual users or integrated with AI-based interactive bots. For instance, when a user attempts to post a tweet with sensitive content that is detected by the proposed mechanism, the user will be alerted that the content has the potential of becoming a future regrettable tweet, i.e. #DontTweet-This. This warning message intends to trigger self-censorship [90]. A delayed posting mechanism suggested in [118] could be invoked, especially for impulsive sensitive posts made by users.

3.3 The Expected Outcome

For this mechanism to be utilized in self-censorship, the privacy scoring model could be implemented as a browser plug-in that would include the pre-trained neural networks (w/o shipping the training data to user). When the plug-in would be used for the first time, it would be configured with the user's topic-specific privacy attitude. For example, some users would configure tweets concerning work related information as private information knowing their colleagues or employers are in their social network, while at the same time, the users may not consider information regarding family as sensitive information. This process acts as the preparation for the customized service. It monitors the socialization activities, for a personalized context-aware privacy score to be displayed to the user before they click the "Tweet" button. If the privacy score of a post surpasses a preset threshold, the user will be alerted that this post has the likelihood of becoming a regrettable tweet, i.e. #DontTweetThis. This warning message again intends to trigger self-censorship for the user [90]. A delayed posting mechanism suggested in [118] could be invoked in these instances.

During use, we will give a customized privacy score for each editing tweet before the user clicks the "Tweet" button, as shown in Figure 3.2. The advanced functions in our pop-up window includes the notification of whether this is a tweet containing sensitive information (*PrivScore*), which kind of sensitive topic the tweet might be related to (*private tweet classification*), what is the potential risk for this kind of topic, and whether the user should consider posting the content with regards to future regret. The privacy score might be higher for instance, if the user initially

Figure 3.2: UI for privacy protection



selected the topic of traveling as a sensitive privacy related issue.

3.4 Anticipated Impacts of The Proposed Solution

The meaningful use of our proposed privacy score mechanism is far more than what we have described above. The implementation of this privacy score mechanism into the OSN industry will benefit innumerable amounts of users. Horizontally, it can be utilized with all information sharing networks including Facebook and Youtube. This mechanism can also be applied with chatbots, which will be described in detail in 5.6.3. Vertically, it can be used as a filter for posting, and further as a filter for receiving. A user can then select the topics that interest them, and that they enjoy reading. Moreover, as society shows concerns regarding the online privacy of young teens including the sharing of inappropriate content such as misbehavior, this mechanism provides a self-examination platform for young users, and provides a monitorial approach for parents and schools. Further use might also include the usage of this mechanism in company intranets to detect information leakage. Overall, as a theoretical privacy protection framework, our work provides an early warning with respect to information leakage, which will benefit many kinds of different users.

Chapter 4

Content-Based Classification of Sensitive Tweets

Abstract

In this Chapter, we will demonstrate how we get these pre-defined private topics, how to classify tweets into these topics using text classification techniques, and how we improved the classification accuracy. Our baseline approach used most common methods in text mining and gave a relatively good performance. We further introduce two kinds of boosting features into our classification model: user topic preference and domain knowledge. Results showed the accuracy had improved notably.

4.1 Introduction

As the most popular open microblog platform, Twitter has 313 million monthly active users [103]. With this new socialization mechanism, users post tweets about every aspect of their daily life, ranging from professional and career development to personal and family updates, from entertainment to political opinions. Besides the diversity of users' topics, Twitter accounts are public by default, which makes Twitter an ideal source for advertisers and adversaries to collect personal information. At the same time, it makes users' private information exposed unexpectedly.

For most of the users, their tweets are intended for friends that follow them, called "followers" on Twitter. However, Twitter is in fact an open platform – all messages posted on Twitter are accessible to the public, including unregistered users and search engine bots. When users are emotional and/or careless and want to post something, few of them remember this fact. Therefore, users

sometimes become very vulnerable – private or sensitive information may be accidentally disclosed, even in tweets about trivial daily activities. Wang *et al.* [118] has shown that regret-tweets are very common. Most of them involve sensitive content and rich sentiments, such as alcohol and illegal drug use, sex, religion and politics, personal and family issues. However, the degree of sensitivity and privacy is a subjective perception, which differs from person to person. For instance, some users are more conservative about health-related issues, while some others might be more protective on work-related information. With that said, in developing a privacy protection mechanism for online social networks, we cannot use a uniform measure of privacy for all users. Classification of topics of private tweets is necessary for customized privacy protection – we can alert users for the pre-set types of private information they want to protect. Meanwhile, we consider *private tweet identification* (i.e., automatically identify if a tweet contains private information) and *private tweet classification* as dual-problems. Progress towards one of them will eventually benefit the other. Moreover, after the classification of different tweets, different strategies for evaluating degrees of sensitiveness can be applied appropriately.

In the rest of this Chapter, we will introduce the preliminary knowledge about the methods and algorithms we used in the paper will be described. The data collection and preprocessing will be followed. In this part, the process of data collection and labeling will be discussed in detail and preprocessing approaches will also be explained. The experiment results will be analyzed at the last part.

4.2 Approach

In this dissertation, we define 13 topics of privacy content, which is shown in Table 1. Our intension is to include topics that some twitter users may not want to share with everyone, i.e., tweet messages that the owners may want to hide from a specific group of followers. While some topics are usually considered as highly sensitive by most of the users, some other topics are only sensitive to a smaller population. For instance, category *Politics* may appear to be far less sensitive than topics such as *Drugs & alcohol*, however, it is not uncommon that some people do not want to

Table 4.1: Tweet categories

Category	Example
1. Health & Medical	Seriously starting to regret this surgery...
	I'm waiting for this cough syrup to go down
2. Work	Nothing like being at work at 6 am! #ineedanewjob
	Pretty sure @XXX and I spend more time snapchatting each other at work than actually working.
3. Drugs & alcohol	Nothin' beats whiskey & coke
	I'm thinking he gets pure Columbian cocaine Pseudo Prof. #druglord
4. Obscenity	I hate fucking a skinny bitch!!! #ineedbigass
	This spring break was kind of trash.
5. Religion	Be strong and take heart and wait for The Lord.
	If you put the Lord first, everything else will fall into place.
6. Politics	If Obama wins I'm becoming a communist!
	I wish I was at that debate to ask obama questions. #debate #tearhimapart #romneyryan
7. Racism	I hate black people and gay people as well
	Nowadays these niggas always caught up in they feelings
8. Family & Personal	Grandma and papa flying in tonight!!
	Drinkin beer with future father in law and shondas uncle #buzzed
9. Relationship	I have no problem flaunting my relationship.
	On a date with a pretty cute girl. Hope @XXX doesn't mind.
10. Sexual Orientation	Tayler just admitted to me that she is bisexual...
	One day I wanna convert a lesbo
11. Travel	I wish I could just leave and go on a long road trip
	4 more hours until a week of paid vacation
12. School life	3 hour class can suck my balls
	What's worse than immature freshman? Immature seniors.
13. Entertainment	Watching bad girls club while I wait for class #noshame
	Watched the series finale of Ally McBeal on #Netflix and now I'm all in my feels about everything in life ever #alldafeels

share political opinions with their supervisors, colleague, or clients. On the other hand, students may not want to share entertainment-related tweets (e.g., going to a party on a school day) with teachers, thus topic *Entertainment* is also considered sensitive for some people.

Our system consists of the following parts: data collection, labeling, data normalization, feature extraction and classification. The first three steps can be seen as preparation for data set, which will be introduced in the data collecting and preprocessing part. In the feature selection part, besides the content-based features, such as Bag-of-Words and TF-IDF, two kinds of boosting features are

explored in our system. They are users' topic preference and domain knowledge. To compare the performance of classifiers trained by different feature sets, Bag-of-Words method is used as the baseline, and another four methods are tested respectively: TF-IDF, TF-IDF with proposed boosting features – users' topic-preference, TF-IDF with proposed boosting features – domain knowledge, TF-IDF with users' topic-preference and domain knowledge which is called "All" for short in the rest of the proposal. The whole process of classification is shown in Figure 4.1. About the classification algorithm, the Naive Bayes model is selected, since it is commonly used in text classification and has good performance. The following part will describe these methods and models in detail.

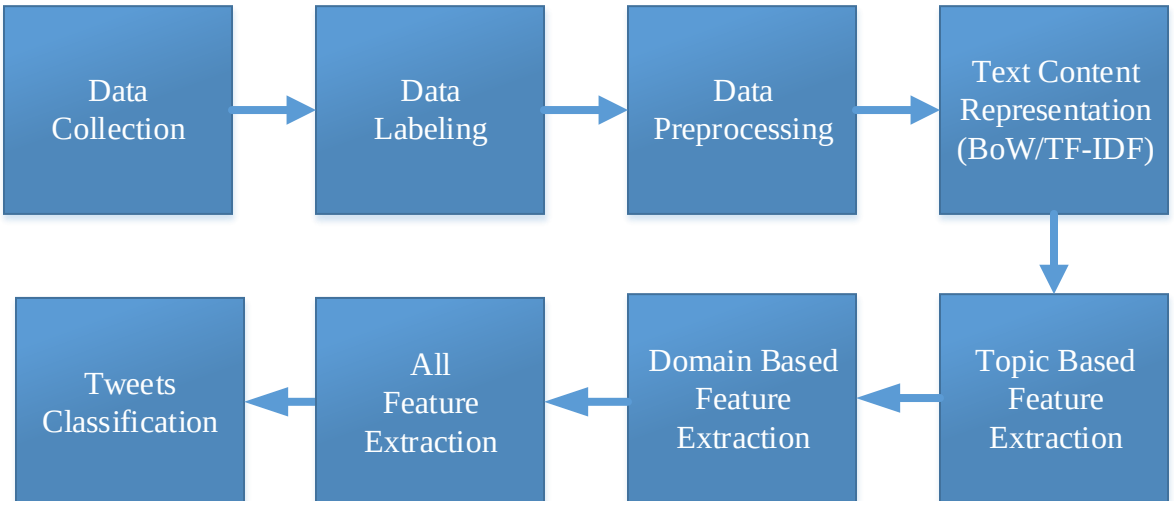


Figure 4.1: Diagram for classification process

4.2.1 Naive Bayes

Naive Bayes is a popular algorithm in text categorization. The motivation of the algorithm can be described as: if each tweet is treated as a document d and d is composed of a bag of words $w_1, w_2, \ldots, w_n$, then the posterior probability that the tweets belong to the topic c can be demon-

strated as

$$\begin{aligned} p(c|d) &= \frac{p(c)p(d|c)}{p(d)} \\ &= \frac{p(c)p(w_1,w_2,...,w_n|c)}{p(w_1,w_2,...,w_n)} \\ &\propto p(c)\prod_{i=1}^{n} p(w_i|c) \end{aligned} \tag{4.1}$$

In this expression, $p(c)$ is the prior probability of a tweet occurring in class c, defined as

$$p(c) = \frac{N_c}{N} \tag{4.2}$$

N_c is the number of tweets in the topic c, and N is the total number of tweets in the training set. $p(w_i|c)$ is the conditional probability of words distribution in category c, which can be calculated as

$$p(w_i|c) = \frac{N(w_i,c)}{\sum_{w_j \in V} N(w_j,c)} \tag{4.3}$$

where $N(w_i,c)$ is number of occurrences of word w_i from topic c, if Bag-of-Words method is applied. If TF-IDF is used, $N(w_i,c)$ will be the TF-IDF score in the certain topic which will be described in the following part. The tweet which is assigned to the best class c can be determined by

$$\arg\max_{c \in C} p(c) \prod_{1 \le k \le n_d} p(w_k|c) \tag{4.4}$$

4.2.2 Bag-of-Words

The Bag-of-Words model is a simplified representation used in a majority of information retrieval and natural language processing techniques. Essentially a text is represented as the bag of its words, disregarding grammar and even word order but keeping multiplicity. For example, both "John likes Mary" and "Mary likes John" can be represented as $\{"John'', "likes'', "Mary''\}$ in BoW model. The frequency of a term (TF), namely the number of times a term appears in the text is used as a feature for training a classifier. By using Bag-of-Words (BoW) model, a tweet can be treated as a set containing all the words appearing in the tweet.

However term frequencies are not the best representation for the text. Common words like 'a', 'the', 'to' are the terms with highest frequency in the text. Thus having a high raw count does not necessarily mean that the corresponding word is more important. Moreover, this method simply uses all words that have appeared in a topic as the feature set to represent each tweet in this certain topic, which will make data set sparse and large, and reduce the classification accuracy. Thus, we use this method as our baseline.

4.2.3 TF-IDF

To address the problems posed by the Bag-of-Words model, a widely used technique of normalizing the term frequencies (TF) is to weight a term by the inverse of document frequency (DF) or TF-IDF. TF-IDF can reduce feature dimension effectively, distinguish the importance of different words and reflect the importance of a word to a document in a corpus[26]. This scheme gives the word w in the document d the weight as

$$TF\text{-}IDF(w,d) = TermFreq(w,d) \cdot \log(N/DocFreq(w)) \tag{4.5}$$

where $TermFreq(w,d)$ is the frequency of the word in the document, N is the number of all documents, and $DocFreq(w)$ is the number of documents containing the word w.

In our system, we first remove stop words from tweets. Then, for each category, they are treated as a document, and the importance of each word in tweets belonging to a category can be calculated based on TF-IDF. Most frequent words and their TF-IDF weights are used to represent each tweet and build data set for classification [57].

4.2.4 Boosting with User Topic Preference

Because of the limitation of tweet-size (140 characters), each tweet contains very few features compared with all the word-features, which makes accurate classification hard. To improve the accuracy of a classifier, not only should semantic feature selection methods be used, such as TF-

IDF, but also features from other perspectives should be considered. In this proposal, we add 13 features, which represent users' topic preferences for 13 categories. The motivation behind introducing boosting features is that different users would have different posting preferences according to these 13 topics. It is a very intuitive assumption that a user who likes traveling, for example, more frequently posts tweets about Travel instead of Drugs and Alcohol. So by adding features about their topic-preferences will improve the accuracy. A user's preference for a topic is estimated by two steps.

Firstly, we define topic-related words as the words which scored more than 1.5 after TF-IDF calculation for each topic. The threshold is selected as 1.5 because during our experiments, this score can decrease the number of features dramatically and still have a good classification effectiveness. Then each user's own topic preference can be calculated through counting the occurrence of topic-related words in each topic and comparing with the occurrence of the whole words in the user's tweet history. The algorithm is stated in the following table 4.2.

Table 4.2: Algorithm for user's own topic preference

Algorithm for a User's Own Topic Preference
Input:
tweets: List of n tweets from a Twitter user *u*
wordList: Related words list of Certain Topic
Output:
ownPreferences: Topic preferences for Twitter user *u*

```
1: words = preProcess(tweets)
2: for topic from 1 to 13 do
3:     prob[topic] = 0
4:     for word in words do
5:         if word in wordList do
6:             prob[topic] += 1
7:     ownPreferences[topic] = prob[topic]/# of words in tweets
```

However, for some popular topics, such as 'Travel', lots of users have a high own topic preference in their tweet history. If we just utilize users' own topic preference as the topic preference, the always low score topics, like 'Racism' or 'Sexual Orientation' might be influenced. To avoid the influence of this evaluation method, the relative topic preference should be considered. The

estimation of relative topic preference is our second step for users' topic preference features' extraction. In this step, all users' own topic preferences for a certain topic are sorted. We treat users whose own topic preferences are among top 50% as having a preference for this certain topic.

Now the classification problem becomes trying to maximize the conditional probability of a tweet belonging to a certain topic given its content and owner's topic preference, which can be defined as

$$\arg\max_{c \in C} p(c|d,t) \tag{4.6}$$

Consider a user with certain topic preferences. When she wants to post a tweet, it is probably that the content related to her topic preferences. Thus, the conditional independence is presumed in this situation. And it can be formally defined as follows in Naive Bayes.

$$p(c|d,t) = p(t)p(c|t)p(d|c) \tag{4.7}$$

For a tweet, its owner's topic preference can always be estimated. Therefore, the conditional probability can be found by calculating

$$p(c|d,t) \propto p(c|t)p(d|c) \propto p(c|t)\prod_{i=1}^{n} p(w_i|c) \tag{4.8}$$

where $p(c|t)$ can be calculated as

$$p(c|t) = \frac{N_{ct}}{N_t} \tag{4.9}$$

4.2.5 Boosting with Domain Specific Features

Besides the content of tweets which are the important features for tweet classification, the background knowledge of the information can also play a role for accurate prediction. Therefore, we leverage the background knowledge of four specific topics – entertainment, work, religion and drugs – as four extra features for each tweet. The judgement of these four features are described in detail below.

4.2.5.1 Entertainment

The Entertainment topic can contain varied information. The tweets might have the information about users' preference for a particular kind of movie, music or artist. If a preliminary knowledge about entertainment has been obtained and implemented in the judgement of tweets, it will be helpful for the classification. Therefore, a script has been wirtten to look at individual tokens in a tweet and check whether any of them reference to entertainment content. IMDB's database, IMDB's API and the Google's API are used to identify whether tweets have any reference to entertainment features. For example, "Rachel is making me celebrate World Oceans Day with her by watching Finding Nemo. No complaints. #wildlifeconservation". In this tweet, the user mentions the movie "Finding Nemo", which can be found in IMDB's database.

4.2.5.2 Work

A common observation for the tweets in the work category includes their work load, salaries and professions. Therefore, a list of professions is created, which covers close to 1,100 job titles. Moreover, regular expressions are written to identify tweets where a monetary income is being discussed e.g. $10,000. And for work load regular expressions are written to identify description of a time like 5am, 6pm, 3 days etc. Take the following tweet as an example, "Act for 4 hours, then work for 7.. Tomorrows gonna kill me" contains a commonly used sentence "work for + number (hours)", which can be checked by our regular expression.

4.2.5.3 Religion

A lot of tweets that were categorized as religion had the name of a chapter in a religious text and the verse number. For example, "I praise you, for I am fearfully and wonderfully made. Wonderful are your works; my soul knows it very well. Psalm 139:14". If a chapter name and a verse number are detected, it's definitely a Religion topic tweet.

4.2.5.4 Drugs and Alcohol

Tweets categorized as drugs had a mention of a drug or an alcoholic drink. So we created two lists. One list for all possible drugs that were illegal or used for recreational purposes and the other list for all kinds of alcoholic drinks. These information is sourced from Wikipedia. For example, in tweet "This song makes me want to do copious amounts of MDMA and cocaine", MDMA (methylenedioxy-n-methylamphetamine) and cocaine are on our list.

4.3 Data Collection, Labeling and Preprocessing

Before training classifier, the data set used for classification should be collected from Twitter. Once the tweets are crawled, randomly selecting and labeling them to 13 pre-defined topics related to privacy content is called data labeling. After these processes, tweets are still merely strings of text. To make the classifier understand the document, there is a need to represent the document in a more structured manner. Hence, the preprocessing of the data set and the text representation is necessary for the experiment. The above process is shown in Figure 4.1.

4.3.1 Data Collection

Data Collection is the process of collecting data that is relevant to our project. The data collected will then be preprocessed and used to make predictions and evaluate outcomes; thus, it is one of the most important steps. The better the data for training the classifier is, the better prediction results there will be. Therefore, some constraints are set during collection, which is demonstrated in the following part of this section.

In this part, we randomly selected a user as the valid seed user and the crawler began from a valid "seed user" by using Twitter Rest API, which provides programmatic access to read and write Twitter data [101]. For a valid user, the following constraints are applied: (1) less than 500 followers or following count, (2) user's account language should be English.

We think a user with more than 500 followers or followings means the user is extremely active,

and their behaviors on the social network are quite different from "normal user". Our research does not target celebrities or public accounts, which are over-active and containing few private information. The seed user's recent 3,000 tweets (according to Twitter's limitation), seed user's followers' and followings' accounts are recorded during crawling. Then we check the seed user's followers and friends to find more potential seed users. If the seed user's friend or follower fulfills the criteria of a seed user mentioned above, then we can start crawling this new seed user's tweets. This method is called snowball crawling. We repeated the snowball crawling method twice using the Twitter Rest API.

More than 29,000 user accounts were crawled in our experiment, from March 10th to March 31th, 2016. From tweets obtained, we deleted tweets containing the term "RT@" or URLs for better quality tweets, since most of them contained less personal information.

After obtaining tweets, the next logical step was to find the tweets most relevant to the project. In order to get tweets that were relevant, a rough list of keywords for each category is created. For example, keywords like hospital and surgery would be added to keyword list of 'Health & Medicine', keywords like vacation and road trip would be added to keyword list of 'Travel', so on so forth. To obtain these keywords, a seed word relevant to the category is selected and then fed the Urban Dictionary [urban] which is an Internet dictionary containing lots of slang and shortenings. Then the 20 most related words of each seed word on the website are extracted. For each related word we found its related words. This process was repeated twice. After populating and proper cleaning, keywords of each category are obtained. These keywords were then used to fetch relevant tweets from the raw tweet data we collected.

4.3.2 Data Labeling

Once the tweets are filtered by keyword sets, the next step is to manually label the tweets into 13 pre-defined topics which might contain private information. The description and the criterion for judging are listed below. Examples of each category are shown in Table 1.

- Health & Medicine: Tweets that describe users' injury, pain, disease, medicine, surgeries or

anything related to hospital visits, etc., are included in this topic category.

- Work: Tweets in this topic category pertain to users' work or employment. Tweets contain users' feeling towards his profession, the work environment, colleagues, work hours or wages, career, job hunt, etc.
- Drugs & Alcohol: Tweets are related to substance abuse or alcoholic consumption.
- Obscene & Abusive: Tweets that contain male and female intimate body parts, sex or porn related text are included in this category. Also, tweets where people use obscene language or complain about something fall in this topic category too.
- Religious: All tweets that indicate religious inclination or contain verses from holy books or talk about faith in God are included in this topic category.
- Politics: Tweets talk about a country's government, policies, elections, etc.
- Discrimination: Tweets that contain text related to discrimination against someone based on cast, color, creed, religion, sexuality, etc.
- Family & Personal Life: Tweets that tell us about the users' personal life. They include birthday tweets, anniversary tweets, marriage or engagement tweets, or pregnancy tweets about the user or his family members.
- Relationship: Tweets related personal relationship.
- Sexual orientation: Tweets that describes a person's sexual orientation.
- Travel: It contains tweets where a user talks about taking a vacation.
- School Life: Tweets containing text related to school like homework, assignments, grades, graduation etc., are included in this category.
- Entertainment: Tweets talk about movies, TV shows, books or music.

For each category, the distribution of tweets is different. For example, it's easy to find a tweet about work, while hard to find one about illegal drug use. To make sure our classifier can distinguish different topics correctly, we select around 500 tweets for each category to make the data set balanced. The distribution of tweets for each category is shown in Figure 4.2.

During labeling, one annotator first labels almost equal number of tweets for each category.

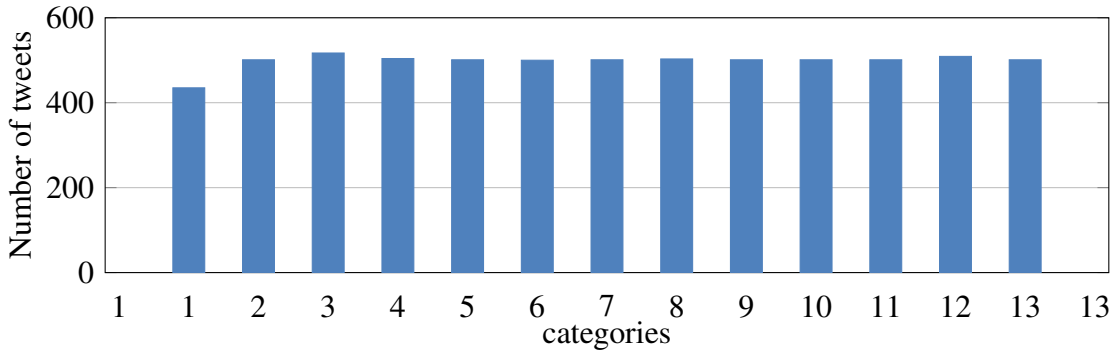


Figure 4.2: Number of tweets in each category

Then the second annotator checks whether the labelings are correct. Only tweets agreed upon by both annotators remain in the data set. In case a tweet belongs to more than one category, the tweet is saved in all relevant documents. For example, tweet like “anonymous yoo baby how’s that sexy ass of yours? Just sitting here thinking about it while I’m working.” is about both work and obscenity. Finally, there are 6,475 labeled tweets in our data set. Among these labeled tweets, 6,345 are distinguished and owned by 3,694 users. We also extract all the tweets of these 3,694 users to analyze their topic preferences, which will be used in the feature selection part of this study.

4.3.3 Tweet Preprocessing

Tweets have the traits of shortness, full of slang and shortenings, and widely usage of hashtags, which makes it hard to understand for computers if we don’t normalize it. Before doing natural language processing for these tweets, a widely used text analyzing tool GATE is used to normalize them. Gate is open source free software for many types of computational task involving human language [GATE].

In GATE, there is a pipeline specifically developed to handle tweets, called TwitIE [11] which includes the components used for recognizing the words in long hashtags and changing normal shortenings to complete words. For example, “#lifeisbeautiful” will become “# life is beautiful” and “lol” will become “laugh out loud”. This process is important, since hashtags usually contain very important words for classification.

After dealing with the hashtags and shortenings by GATE, further processing is also needed. Tweets are tokenized to words through using python library nltk. For each word token, tokens starting with @ are removed and tokens containing non ascii elements like emojis are also deleted.

4.4 Experiment Results

The experiments are performed using the popular machine learning tool – Weka [38]. Weka supports many machine learning algorithms for data categorization, clustering, and feature selection. In our experiments, we implement the Naive Bayes model in Weka for five different feature-sets extracting from the labeled data set. The first feature-set consists of labeled tweets processed by Bag-of-Words model. The second one is data set processed by TF-IDF. And only words with a TF-IDF score more than 1.5 are selected. The third one is based on the second one but adding features of users' topic-preferences. The fourth consists of the features in the second one and domain-knowledge features. The last one combines the TF-IDF features and two kinds of boosting features: users' topic preference and domain-knowledge, which is called "All" in our experiment for short. After classification, each tweet in the data set will be in only one category. We utilize 5-fold cross validation to evaluate the classification accuracy, and the final results are averaged over the five folds. The performances of these five different feature-sets are analyzed one by one in the following part. And the experiment results show that compared with the Bag-of-Words, the effectiveness of TF-IDF is obvious and after introducing boosting features, the accuracy of classification improved from 85.4% to 89.2%.

Table 4.8 presents the comparison of classifiers' accuracy, precision, recall and F-measure of the five different feature-sets. The classifiers' performances in each category are evaluated by F-measure and shown in Table 4.9. To visualize the Table 4.9 and observe the result more directly, Figure **??** is drawn.

As the baseline, the Bag-of-Words method achieves the accuracy of 78.8%. Table 4.3 shows the Confusion matrix of Bag-of-Words approach. From the table we can see, the mis-prediction happens in every category and distributes evenly compared with the other four methods. This is due

to the mechanism of Bag-of-Words, which only counts the frequency of words, regardless of order and without distinguishing the importance between words. For example, some common words like "people" and "girls" will count in every category. Moreover, this method can not decrease the feature-dimension effectively, which leads to the sparsity of the feature-matrix. A more advanced representing method is needed for content-based feature extraction. Therefore, TF-IDF is used in our second experiment.

Table 4.3: Confusion Matrix with Bag-of-Words

Truth \ Predict	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	334	18	5	20	3	0	1	15	9	2	10	10	8
C2	10	401	4	21	3	2	1	7	12	3	16	8	12
C3	12	14	381	22	2	5	6	13	6	7	7	10	16
C4	25	31	10	307	3	0	14	20	36	13	8	20	16
C5	3	6	0	8	463	1	1	1	8	1	2	1	6
C6	6	18	3	13	5	412	12	6	7	5	2	5	7
C7	7	5	3	17	3	9	390	7	13	36	3	5	3
C8	11	14	7	8	2	2	1	390	28	2	16	9	19
C9	5	15	3	23	5	0	5	16	394	4	3	7	21
C10	9	8	7	31	6	2	26	12	18	365	2	8	7
C11	4	6	1	9	3	1	2	9	9	3	455	9	6
C12	10	6	3	13	7	1	6	3	7	5	12	409	12
C13	8	15	2	14	2	1	3	20	17	3	11	5	400

In the second experiment, each category is treated as a document, and each tweet is a sentence in this document. Then, for each category, TF-IDF is utilized to dress the important words in each topic. Finally, we select 1.5 as the threshold of TF-IDF score, which decreases the number of features from a huge one – 10,107 – to an acceptable one – 2,369 – and increases the accuracy of classification from 78.8% to 85.4% effectively. Table 4.4 shows the Confusion Matrix of the classification results using the feature-set extracted by TF-IDF. We can see the results are quite different from that of the Bag-of-Words method. Most of the wrong predicted tweets are categorized as topic 13. This is because entertainment contains varied content. Except for the specific words like 'Netflix', 'cinema' and 'movie' with extremely high scores, words like 'club' and 'doctor' which might appear in other categories also have a score more than 1.5. For example, tweet labeled as work –"When a parent tells me I was a huge impact on their daughter's life when I

worked at the boys and girls club" is hard to judge for the classifier and wrongly categorized as topic 13. Although the overall accuracy of TF-IDF improved a lot compared with Bag-of-Words, in some specific topics, such as 'Entertainment' and 'Obscenity' which share parts of important words with other categories, this method cannot work very well. For example, on the topics of 'Sexual Orientation' and 'Drug and Alcohol', dirty words are common. One of the mis-prediction shows this problem, "Anonymous is a lesbian but she had sex with hale so the makes her bisexual. This notch got it down". This tweet is labeled as topic 9 – sexual orientation, but with little dirty meaning.

Table 4.4: Confusion Matrix with TF-IDF

Truth \ Predict	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	372	0	1	0	0	0	0	2	0	0	0	1	59
C2	0	427	3	2	0	0	0	2	1	0	17	2	46
C3	2	0	455	6	0	0	1	0	0	0	0	1	36
C4	1	0	9	312	0	0	0	3	4	0	4	0	170
C5	0	0	0	0	487	0	0	0	3	0	0	0	11
C6	0	0	2	0	6	450	1	0	0	1	5	5	31
C7	0	0	0	1	1	4	420	0	2	16	0	3	72
C8	1	0	0	0	3	1	0	463	0	0	16	5	20
C9	0	0	0	7	3	0	0	2	436	0	0	0	53
C10	0	0	0	11	0	0	7	4	1	374	0	0	104
C11	0	3	0	1	0	0	0	1	0	0	412	3	97
C12	1	0	0	0	0	0	1	0	0	0	20	458	24
C13	2	0	1	0	1	0	0	3	0	0	5	6	483

Therefore, only using content-based features are not good enough for accurate classification. In the following part, two kinds of boosting features are added to the TF-IDF feature-set respectively. And finally, except for the Bag-of-Words, all methods are combined together, which improves accuracy more than 10% compared with the baseline.

The very beginning intuition behind adding users' topic preferences is that users are more likely to post the tweets about their interested topics. However, for some extremely sensitive topics such as racism or sexual orientation, though a few users might have a higher preference than that of the other users, these topics are still seldom mentioned, compared with some other 'popular' topics like 'work' or 'travel'. Thus, the user's own topic preference is not accurate enough for classification.

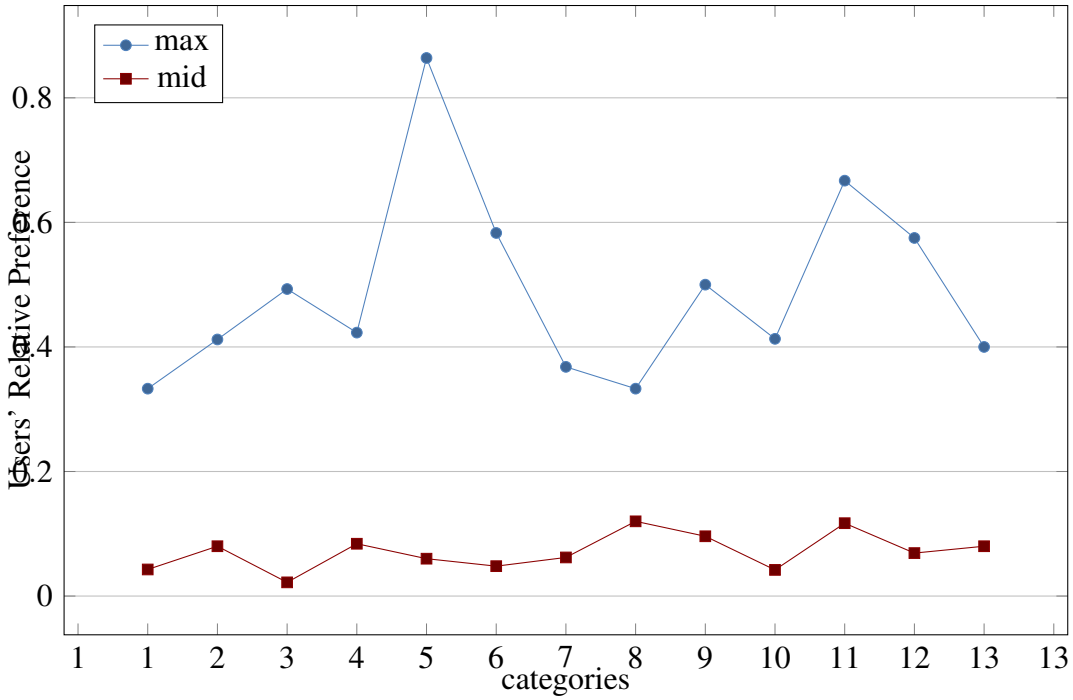


Figure 4.3: Users' Relative Topic Preferences

The relative topic preferences are needed, which has been described clearly in section 3.4.

The Figure 4.3 shows the maximum of a user's own topic preference and the medium of a user's own topic preference respectively. Combined with Table 4.9, we find, this method has an obvious effect on category 4, 7, 10, 11 and 13. This is partial because, topic 4, 7 and 10 are seldom talked about on Twitter. But for the topic active users, they still have more tweets on these topics compared with the most users. For category 11 – travel, it is almost the most favorite topic for every user in our sample set. Then the relative preference makes more sense for the extreme popular topic. The improvement of topics 13 – 'Entertainment' is due to the decrease of mis-prediction. The Confusion Matrix in Table 4.5 can confirm the demonstration of improvement for various categories.

As mentioned before, through using domain knowledge, more accurate topic-related words will be grasped. By observing the Confusion Matrix in Table 4.6, we can see improvement in category 'Work' is obvious, due to the application of domain knowledge. So does the category 'Entertainment'. Compared with TF-IDF, the accuracy of domain knowledge method is not im-

Table 4.5: Confusion Matrix with TF-IDF + topic

Truth \ Predict	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	374	0	15	1	7	0	0	2	0	0	0	8	28
C2	1	424	9	4	1	1	0	2	1	0	20	8	29
C3	2	2	460	7	1	1	1	0	0	0	0	5	22
C4	7	0	67	336	2	0	0	3	4	0	4	24	56
C5	1	0	0	0	486	0	0	0	3	0	0	0	11
C6	0	0	6	1	4	458	1	0	0	0	4	10	17
C7	0	0	18	5	7	5	414	0	1	15	0	7	29
C8	1	0	6	0	5	1	0	464	0	0	16	7	9
C9	2	1	17	10	2	0	1	2	435	0	0	4	27
C10	0	0	36	18	3	0	9	4	1	393	0	6	31
C11	0	3	17	0	3	0	0	1	0	0	430	6	57
C12	2	0	3	0	1	1	1	0	0	0	19	465	12
C13	2	0	5	0	0	0	0	3	0	0	5	9	477

proved in category ‘Religion’ and ‘Drug and Alcohol’. This phenomenon verifies that the topic related words in these two categories are very distinct among topics and ubiquitous in the specific topic, which makes TF-IDF a more powerful approach.

Table 4.6: Confusion Matrix with TF-IDF + domain knowledge

Truth \ Predict	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	394	0	5	0	0	0	0	2	0	0	0	33	1
C2	0	454	2	1	0	0	0	3	0	0	8	3	29
C3	2	17	472	5	0	0	1	0	0	0	0	4	0
C4	5	0	38	318	0	0	0	3	5	0	4	130	0
C5	1	0	1	0	486	0	0	1	3	0	0	5	4
C6	0	0	2	0	9	465	1	0	0	1	5	18	0
C7	0	0	1	1	1	4	413	0	2	15	0	64	0
C8	0	0	2	0	3	1	0	463	0	0	16	24	0
C9	0	0	2	8	3	0	1	2	446	0	0	38	1
C10	0	1	4	12	0	0	7	4	1	388	0	82	2
C11	3	2	2	1	0	0	0	1	0	0	451	53	4
C12	1	1	3	0	0	0	1	0	0	0	19	475	4
C13	2	0	1	0	0	0	0	2	0	0	5	20	471

After combining TF-IDF, users’ topic preferences and domain-knowledge together, we get the classification result as shown in Table 4.7. Compared with the previous results, fewer tweets are mis-predicted as category 13, due to the usage of domain knowledge. However, at the same time,

the mis-prediction on some topics increased. This is because our domain-knowledge features just stress the characters on the four specific categories. For some categories with unobvious content-based traits, such as 'School life', the accuracy will be compromised.

Table 4.7: Confusion Matrix with All

Truth \ Predict	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13
C1	389	0	12	5	1	1	0	4	1	0	0	21	1
C2	0	457	4	0	0	0	0	2	0	0	8	2	27
C3	3	16	475	3	0	0	1	0	0	0	0	3	0
C4	16	0	40	373	1	0	0	4	4	0	4	59	2
C5	1	0	1	0	486	1	0	2	3	0	0	1	6
C6	4	0	2	4	4	464	2	0	0	0	4	14	3
C7	5	0	8	7	1	9	424	1	1	15	0	27	3
C8	6	0	0	3	2	2	0	468	0	0	16	12	0
C9	12	0	3	22	1	0	1	2	445	0	0	13	2
C10	11	0	10	33	0	6	9	4	1	402	0	23	2
C11	9	2	7	3	0	1	1	2	0	0	451	36	5
C12	4	1	3	0	0	1	1	0	0	0	19	471	4
C13	4	0	2	1	0	1	0	2	0	0	5	14	472

To make the comparison more clearly, Table 4.8 has shown the accuracy, precision, recall and F-Measure for five different feature-sets respectively. And Table 4.9 lists the F-measure score for each category under five conditions. From the results, we can see that TF-IDF is an effective content-based feature extraction method, with significant improvement compared with the Bag-of-Words. However, for some categories whose topic-related words are also parts of other categories' topic-related words, this method performs bad, such as topic 'Obscenity' and 'Entertainment'. To make up this disadvantage, boosting features (i.e., users' topic preferences and domain knowledge) are introduced. After combining TF-IDF and boosting features together, the classification accuracy has a notable improvement.

To make sure that our approach can be implemented in a real-time system, we also used Weka to evaluate the time for building model on training data and time taken to test each tweet based on five different classifiers, shown in Table 4.10. From the results we can see, BoW takes the most time both of training and testing. This is because, the feature set of BoW contains more than 10,000 words, which is five times that of the other four methods. The module building time for the other

Table 4.8: Comparison of Different Model

Models	Accuracy	Precision	Recall	F-Measure
BoW	0.788	0.793	0.788	0.789
TF-IDF	0.854	**0.915**	0.854	0.870
TF-IDF+topic	0.867	0.891	0.867	0.871
TF-IDF+domain	0.879	0.911	0.880	0.886
TF-IDF+topic+domain	**0.892**	0.902	**0.892**	**0.894**

Table 4.9: F-measure Score of Each Category

Category	BoW	TF-IDF	topic	domain	All
1. Health & Medical	0.760	0.914	0.904	**0.935**	0.865
2. Work	0.752	0.918	0.912	0.931	**0.936**
3. Drugs & alcohol	0.819	**0.936**	0.793	0.911	0.890
4. Obscenity	0.609	0.740	0.759	0.749	**0.780**
5. Religion	0.919	0.972	0.950	0.969	**0.975**
6. Politics	0.879	0.941	0.946	**0.958**	0.940
7. Racism	0.805	0.881	0.892	0.893	**0.902**
8. Family & Personal Info	0.759	0.936	**0.937**	0.935	0.936
9. Relationship	0.740	0.920	0.920	**0.931**	**0.931**
10. Sexual Orientation	0.768	0.839	0.865	0.857	**0.876**
11. Travel	0.855	0.827	0.847	0.880	**0.881**
12. School life	0.810	**0.927**	0.875	0.654	0.785
13. Entertainment	0.774	0.566	0.730	**0.926**	0.918

four methods are almost the same. The testing time on each tweet decides whether our proposed method is a real-time solution since the final goal of the project is to check users' posts in the real time and recommend users before they want to post something sensitive. The results in our table shows, except for BoW, the others' time consuming are around 0.01s, which is fast enough for real-time realization.

Table 4.10: Training and Testing Time for five Different Models

Models	Module Build Time (second)	Testing Time on each tweet (second)
BoW	10.13	0.0395
TF-IDF	2.7	0.0096
TF-IDF+topic	2.6	0.0091
TF-IDF+domain	2.36	0.0096
TF-IDF+topic+domain	2.07	0.0093

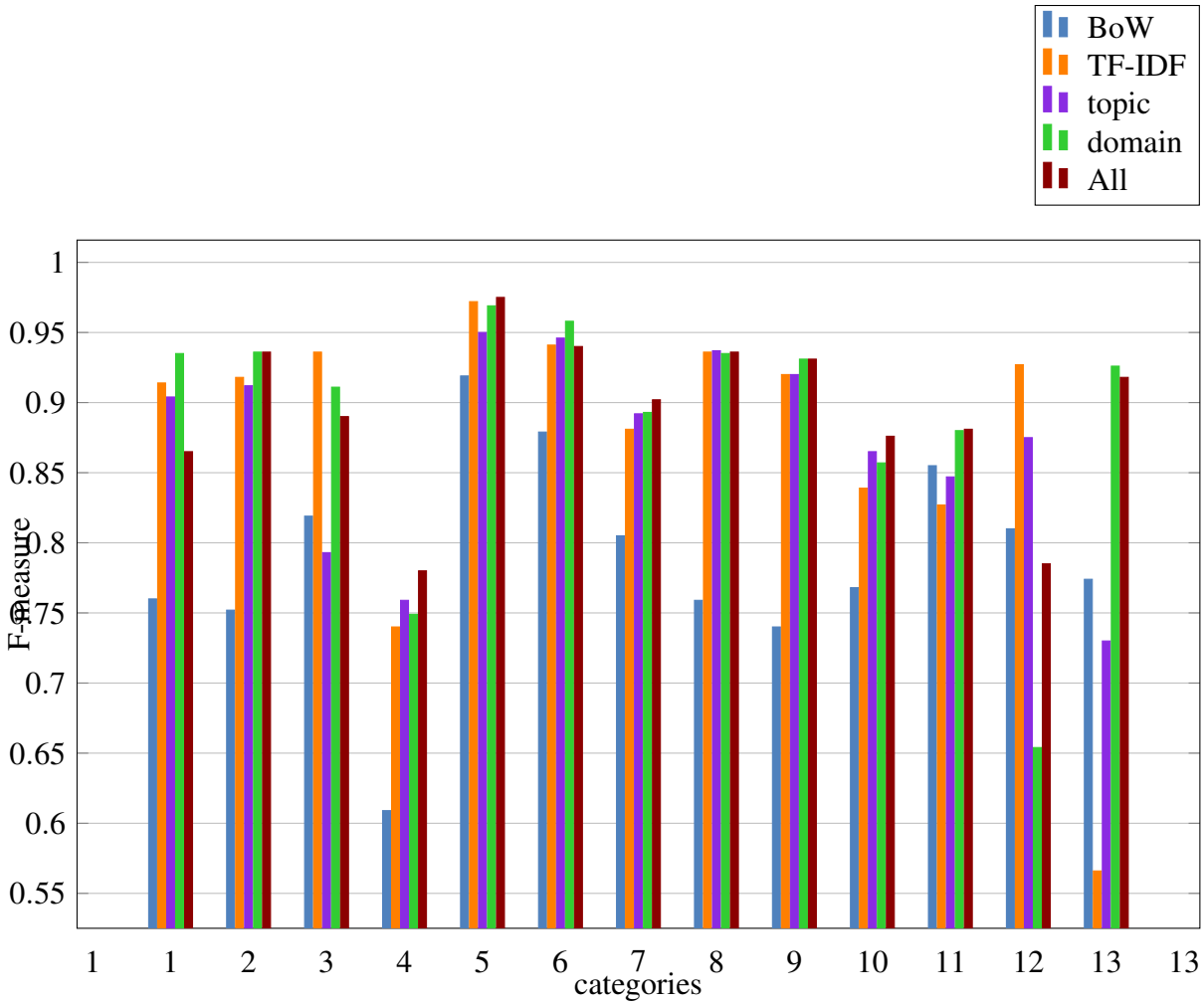


Figure 4.4: F-measure Score of Each Category

4.5 Analysis and Discussion

The previous experiment results show that our motivation of adding boosting features has an impact on the accuracy of classifiers. However, for the results, there are still several phenomena need to be explained in the following part of this section.

From the results, we can see that even the simple bag of words model produces accuracy higher than 70%, which is relatively high, especially considering that there are 13 categories. This is partly due to the existence of bias in the dataset caused by the labeling process. The first annotator quickly scans through a large number of tweets and labels tweets into a category when certain keywords are spotted. For example, when the annotator sees terms like “drunk”, “intoxicated”, the

tweet is labeled as *Drugs & alcohol*. If a tweet contains terms that are weakly associated with this category, e.g. "a cup of beer before dinner", the tweet is labeled as "not sensitive", and eliminated from the dataset. As a result, each category only contains tweets with strong indicator words. That is, to some extent, inadvertent word filtering is made during the human cognitive process in data labeling. In our future work, we will include a significantly larger amount of data labeled through crowdsourcing platforms.

When compared with the result of TF-IDF, we might notice that the improvement after introducing users' topic preferences seems not ideal enough. This is due to the size of the data set. As we randomly selected the tweets from users' tweet history pool, the chance of getting several tweets from the same user is low. After checking, we find 30% of users have more than 1 tweets in our data set. And the chance that these same user's tweets belong to this user's preferred topics are even lower. With including more tweets in our data set and targeted on the specific users in the future work, we believe the impact of this method will have a significant improvement.

4.6 Summary

In this Chapter, we study the problem of classifying private tweets into 13 different potentially sensitive topics based on the common TF-IDF method and boosting features – users' topic-preferences. The experiment results show that with users' topic-preferences and domain-knowledge, the accuracy of classification will increase notably. Our boosting features effectively boost the classification performance of each category, especially for the ones that BoW and TF-IDF are most inaccurate. This work was published on [113, 114].

Chapter 5

Scoring Private Information in Social Networks

Abstract

In this chapter, we will describe our context-aware, text-based quantitative model for private information assessment, namely *PrivScore*, in detail. This model serves as the foundation of our privacy protection mechanism. We start from understanding diverse opinions on the sensitiveness of private information from crowdsourcing workers, and discovering a perceptual model behind the consensuses and disagreements. Then, a computational scheme using deep neural networks is implemented to calculate a context-free PrivScore (i.e., the "consensus" privacy score among average users). Finally, we integrate tweet histories, topic preferences and social contexts to generate a personalized context-aware PrivScore.

5.1 Introduction

As we talked in the previous chapters, we argue that another key component in privacy protection in OSNs is protecting *sensitive/private content*, beyond the protection of identities and profile attributes, i.e., *privacy as having the ability to control the dissemination of sensitive information*. Meanwhile, friends may leak one's private information. Threats from within users' friend networks – insider threats by human or bots – may be more concerning because they are much less likely to be mitigated through existing solutions, e.g., the use of privacy settings [51, 99, 123, 107]. Therefore, it is critical to automatically identify potentially sensitive posts and alert users before they are posted, i.e., #DontTweetThis, and a mechanism to distinguish potentially sensitive/private posts before they are sent is urgently needed. Such a mechanism could also benefit non-human

users such as social media chatbots. For instance, Microsoft's Twitter bot, Tay, started to deliver racist and hateful content soon after it was launched in 2016. Tay "learned" from inappropriate messages it had received. Unfortunately, there did not exist a mechanism to assess the sensitiveness of tweets before they were exposed to Tay or posted by Tay.

In this Chapter, we present the first quantitative model for private information assessment, which generates a PrivScore that indicates the level of sensitiveness of text content. PrivScore is expected to serve as the foundation for a comprehensive user-centered privacy protection solution. In this project, we examine users' opinions on the levels of sensitiveness of content, and then build a semantic model that comprehends the opinions to generate a *context-free PrivScore*. The model learns the sensitiveness of the content from text features (e.g., word embeddings) and sentiment features using a Recurrent Neural Network (RNN). Then the advanced model – Bert is leveraged in this project to explore the benefits of the transfer learning model. To further personalize PrivScore and make it aware of the societal context, we integrate the topic-based personal attitudes and the trending topics into privacy scoring, to generate the *personalized PrivScore* and the *context-aware PrivScore*, respectively. With intensive experiments[1], we show that PrivScores are consistent with users' privacy perceptions.

PrivScore, to the best of our knowledge, is the first quantitative assessment for sensitive content. It has the potential to be utilized in various applications: (1) It could be adopted by individual users for self-censorship and parental controls, to prevent highly sensitive content from being posted to online social networks, especially when the users are careless or emotional. (2) PrivScore could be integrated with AI-based interactive agents, especially the ones with learning capabilities, such as social media chatbots (Twitterbots, Dominator) and virtual assistants (Siri, Alexa, Cortana), to evaluate the content before delivering to users. (3) PrivScore could be aggregated over a large population (across demographic groups, friend circles, users in an organization, etc.) to examine privacy attitudes from a statistical perspective. This method and the results could be used for research purposes, assisting policy making, or privacy education/training.

[1]Since the experiments involve human subjects, we have obtained an IRB approval.

The rest of the Chapter is organized as follows: Section 5.2 explains the data collection and annotation processes, followed by the context-free, context-aware and personalized PrivScore models in Sections 5.6.1, 5.7, and . We present the security analysis and discuss the performance, usability, and limitations in Section 5.6.2. We then summarize the literature in Section 8 and finally conclude the paper in Section 9.

5.2 Data Collection and Labeling

Before training classifier, the data set used for classification should be prepared first. Since there is no previous work same with us, which means we do not have an off the shelf data set to analyze, we need to get the dataset from scratch. This process includes data collection, data labeling and data normalization. In the data labeling part, we utilized widely-accepted online crowd-sourcing approach – Amazon Mechanical Turk. And the design of the survey will be describe in detail on portion 5.2.2.

5.2.1 Data Collection

We selected Twitter as the OSN platform because of its openness and popularity. While it is comparably easy to crawl large amount of data from Twitter, a number of privacy risks have been identified due to this easy data harvesting. We performed a snowball crawling process in March 2016 for about a month and collected 31,495,500 tweets from 29,293 users. We eliminated non-English speaking users, and tweets beginning with "RT @", since forwarded articles and re-tweets do not contain private information of the forwarder.

It is impractical to ask annotators to label 31 million tweets. Meanwhile, since the dataset is highly imbalanced, if we randomly sampled tweets for labeling, the majority of the samples would be non-sensitive. Therefore, we selected potentially sensitive tweets as candidates for labeling to save labor and cost. Note that our goal is not to develop an accurate classifier in this step. Instead, we aim to construct a balanced dataset by eliminating most of the clearly non-private tweets.

First, we referred to [47, 126, 113, 71] to identify potentially private topics, such as *Health & Medical, Drugs & Alcohol, Obscenity, Politics, Racism, Family & Personal*. For each topic, we selected a root set of "seed terms" and expanded the set using *Urban Dictionary*, an Internet dictionary containing slang words and abbreviations (frequently used in Twitter). For each seed word, we expand it to 20 most relevant words.[2]. After proper cleaning, we collected more than 100 terms for each topic. Terms are available at `http://bit.ly/privscore`. By selecting a relatively large set of keywords, we aim to increase recall, i.e., to include a majority of potentially private tweets. We then filtered all the tweets with the candidate terms. In total, we extracted 6,917,044 candidate tweets (i.e., 21.9% of the crawled tweets) that contained at least one of the terms. To confirm the recall of the filtering mechanism, we randomly sampled 500 tweets from the non-candidate set. Close examination showed that only two of them appeared to be slightly sensitive. Eventually, we selected more than 10,000 distinct tweets to be labeled in MTurk, among which 9,936 tweets were included in the annotated dataset.

Collection for Analyzing Trending Topics: In order to collect testing samples that are irrelevant to the training data and add trending topic information (for societal context modeling) into the dataset, we performed a second crawl in March 2018. We monitored the *trends* at a 15-minute interval, and recorded the corresponding *tweet_volume* [102]. In total, 1,130 trending topics with volume larger than 10,000 were collected. We also collected 8,079 new tweets from the same set of users that we crawled in 2016, during the same time period when we crawl the trending topics. This new dataset is used later to evaluate our privacy scoring approaches.

5.2.2 Data Labeling

Keyword spotting could be used to coarsely identify potentially sensitive tweets. However, it is too simple to mimic the human cognition of sensitiveness. In particular, our preliminary examination revealed that a significant portion ($> 50\%$) of the candidate tweet set were indeed not sensitive to us. To annotate sensitive tweets, we collected opinions from a large number of users through a

[2]We recently noticed that the function was removed by Urban Dictionary

crowd-sourcing platform Amazon Mechanical Turk.

We sampled from 6M potentially sensitive tweets to generate questionnaires of 20 tweets each. The number of tweets containing each keyword conforms to Zipf's distribution [52]. To ensure that less frequent terms still get represented in the labeling set, we used a biased sampling process (i.e., using a biased die) that gives higher probabilities to rarer terms.

Turkers (English speakers in the US, with 95%+ approval rates) were asked to annotate each tweet as: [1:Very sensitive]; [2:Sensitive]; [3:Little Sensitive]; [4:Maybe]; [5:Nonsensitive]. That is, a score $s_t \in \{1, ..., 5\}$ is assigned to each tweet by a Turker. Note that we did not use the standard 5-level Likert: [2:very-sensitive][1:Sensitive][0:neutral/undecided] [-1:nonsensitive][-2:very-nonsensitive], because it is hard to judge between [-1] and [-2] in the Likert scale, i.e., to tell if a tweet is "more non-sensitive" than another.

Each Turker was paid $0.45 per questionnaire. For attention check, we embedded two non-random questions in each questionnaire, which were selected from two very small sets of clearly non-sensitive or very sensitive tweets, e.g. Q16 (non-sensitive): *Btw if you're my friend, I love you* and Q17 (sensitive): *Wild crazy strip cloths off at club the. Forgot this morning where I parked /: drank way to f–king much!!! #gayboyproblem*. The screenshot of these two anchored questions is shown in Figure 5.1. We thought if the score of question 17 less than that of 16 more than 1, the worker treated it seriously enough and the questionnaire would be used for analysis. We discarded questionnaires answering $s_{16} \leqslant s_{17}$ and re-posted the tasks to MTurk. Tasks passing the attention check were completed in 140 to 647 seconds, with a median of 249 seconds. Each Turker was limited to answer only one questionnaire and each questionnaire was answered by three Turkers. Eventually, we collected 552 qualified questionnaires from 1,656 Turkers. After eliminating the attention-check tweets, our final dataset contains 9,936 distinct tweets and 29,808 scores.

We also impose a question to survey the Turker's self-reported attitude towards OSN privacy. We define six levels of attitudes as: (1) "I am not concerned about my privacy. I would post just anything on Twitter."; (2) "I am a little concerned about my privacy. I would not post private information, but I do not carefully check each tweet before I post it."; (3) "I am concerned about

Figure 5.1: Control Questions of the Questionnaire

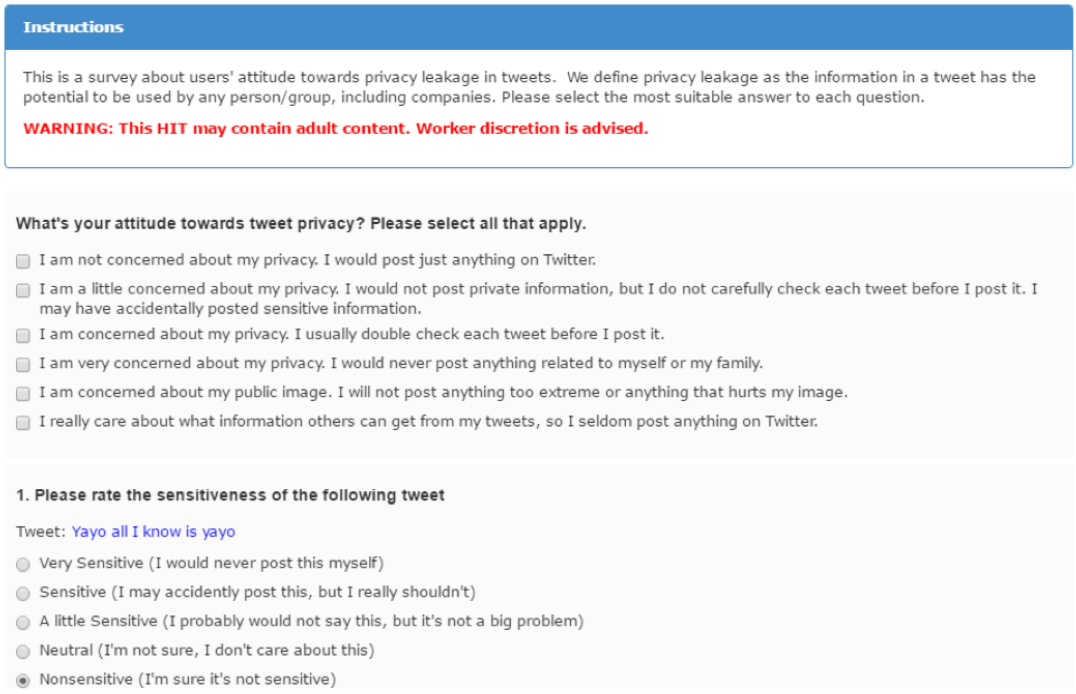


Figure 5.2: An Example of the Questionnaire

my privacy. I usually double check each tweet before I post it."; (4) "I am concerned about my public image. I will not post anything too extreme or anything that hurts my image."; (5) "I am very concerned about my privacy. I would never post anything related to myself or my family."; (6) "I seldom post anything on Twitter." Since the last three options ask for users' attitudes from different aspects, Turkers can select multiple options in this question. The beginning of the questionnaire is shown in Figure 5.2.

5.3 User Privacy Perception

We eventually collected 552 qualified questionnaires from 1,656 Turkers, with 9,936 distinct tweets and 29,808 scores. Our first task is to model users' privacy perceptions by analyzing their attitudes, inter-rater agreement (IRA), and topic-specific attitudes. These initial analysis gave us insights to questions such as "What are the consensuses for sensitive/private tweets? What are the common attitudes among users? Are there consistencies between users' self-reported attitudes and their judgments on real tweets?" Such insights help us tune the collected dataset and design the computational model for privacy scoring.

5.3.1 Analysis of the User Privacy Attitudes

Since in the attitude part Turks can select several options, for each user, we calculate the self-reported privacy attitude score as the mean score of the selected options, where 1 represents "do not care" and 6 represents "really care". The number of responses in each option is: [1]:42, [2]:429, [3]:767, [4]:336, [5]:542, [6]:359 (multiple selections are allowed). From the distribution, we can see that the self-reported privacy attitude is relatively diverse.

When we further examine their opinions on the levels of privacy of the candidate tweets, as shown in Figure 5.3. For each question number, we have 1652 different Turks labeled 552 random selected potentially sensitive tweets, except for question 16 and 17 (control questions). We find that the sensitiveness distributed evenly among different questionnaires. Among the responses, 12.44%

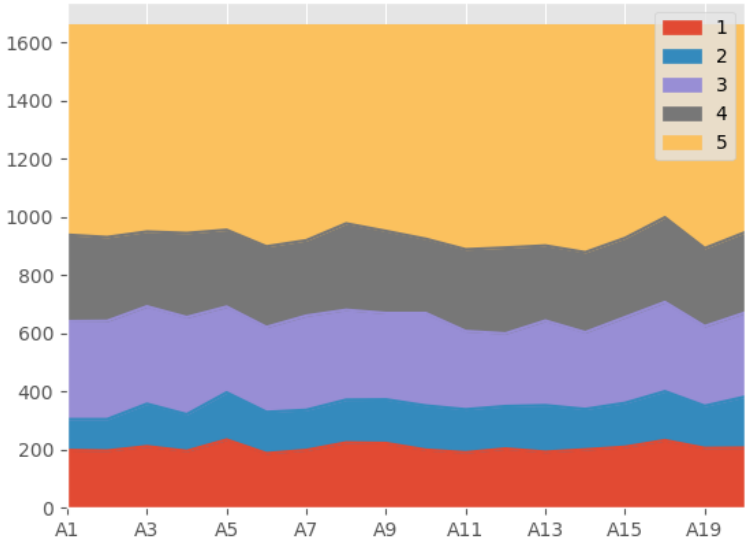


Figure 5.3: Score Distribution

of the annotated scores are [1: extremely sensitive]; 8.73% of the scores are [2: sensitive]; 18.17% of the scores are [3: little sensitive]; 16.66% of the scores are [4: maybe]; and 44.01% of the scores are [5: Nonsensitive].

Next, we analyze if there exists any correlation between Turkers' self-reported attitudes and their privacy scoring. Each questionnaire is answered by three Turkers. Hence, we examine the consistency among each three-Turker group. In particular, we calculate the variance of self-reported privacy attitudes in group i as $v_{a,i}$, where $i \in [1,552]$. We also calculate the average score annotated by each user, and denote the variance of group i as $v_{s,i}$. The normalized average variance of self-reported attitudes $\bar{v}_a$ is 0.34, while the normalized average variance of annotated scores $\bar{v}_s$ is only 0.21. Moreover, we randomly selected 80 questionnaires to demonstrate normalized $v_{a,i}$ and $v_{s,i}$ in Figure 5.4. From the figure, we can see that although different Turkers report relatively diverse privacy attitudes, their annotations of sensitiveness are more consistent. Moreover, we have not observed a strong correlation between the self-reported attitudes and the annotated privacy scores. Therefore, we choose not to include the self-reported privacy attitude in the predictive model for privacy scoring. This also demonstrates the feasibility of a commonly-accepted sensitiveness/pri-

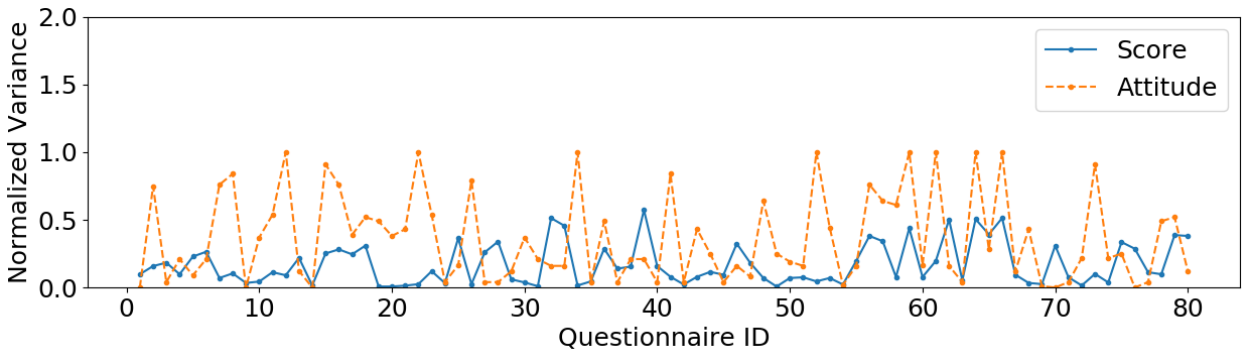


Figure 5.4: Normalized variances of Turkers' self-reported privacy attitudes (dashed orange line) and normalized variances of Turkers' annotated privacy scores (solid blue line).

vacy score mechanism.

5.3.2 Inter-Rater Agreement (IRA)

To examine the consistency across multiple annotators for each questionnaire, in this section, we introduce three approaches that are used to assess Inter-rater Agreement (IRA): *Fleiss' Kappa* measures the agreements between raters on categorical labels, *Pearson Correlation* measures the linear dependency between two variables, and *Spearman Correlation* measures the strength of monotonic (but not necessarily linear) relationship between two variables.

Fleiss Kappa. To statistically measure the agreement between two raters on categorical labels, *Cohen's Kappa* was introduced as a more reliable indicator than calculating percentage of agreements. *Fleiss' Kappa* extended Cohen's Kappa to measure the agreement between more than two raters. The Kappa, k, is defined as:

$$k = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{5.1}$$

In this formula, the denominator denotes the *agreement by chance*, i.e., the degree of agreement among multiple raters that is attainable above chance. The numerator denotes the *observed agreement*, i.e., the degree of agreement that is achieved by these raters. That is, *Fleiss' Kappa* quantitatively measures the actual degree of agreement in comparing with completely random raters, i.e., the level of agreement when the raters' selections are completely random [28]. A smaller k

(e.g., $k < 0$) indicates poor agreement among raters, while a larger K (e.g., $k \to 1$) indicates good agreement.

Pearson Correlation. Fleiss' Kappa was designed for categorical data, therefore, it treats each label as an independent category. In our experiments, when two raters label a tweet as [1 Very sensitive] and [2 Sensitive], while another two raters label two tweets as [1 Very sensitive] and [5 Nonsensitive], they are considered as equally inconsistent by Fleiss' Kappa. But in reality, 1 and 2 are significantly more consistent than 1 and 5. To better handle numerical data, *Pearson Correlation* was designed to capture the linear dependency between two variables X and Y, which is denoted as:

$$r = (\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}))/(\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}) \tag{5.2}$$

where x_i and y_i are indexed samples from two variables, $\bar{x}$ and $\bar{y}$ denotes the sample mean. The numerator captures the covariance of the two variables, while the denominator denotes the standard deviations of X and Y. For two non-negative variables, $r = 0$ indicates that there is no linear correlation between X and Y, while $r = 1$ indicates a perfectly linear relationship between X and Y.

Spearman's Rank Order Correlation. Last, the Spearman Correlation captures the agreement between two annotators in terms of the correlation between the ranks of their labels. It is very similar to Pearson, but it considers the relationships between the ranks of X and Y, instead of directly on X and Y. It is specified as:

$$r_s = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X}\sigma_{rg_Y}} \tag{5.3}$$

where rg_X and rg_Y denotes the rank order of X and Y. The correlation coefficient is interpreted similarly as Pearson correlation, where 0 indicates no correlation in rank orders and 1 indicates perfect correlation between rank orders of X and Y.

The results are shown in Table 5.1. Note that k in Fleiss' Kappa and r in Person/Spearman

Table 5.1: Interrater Agreement based on Fleiss' Kappa, Pearson, and Spearman (P:Poor; Sl:Slight; F:Fair; M+:Moderate+; VW:Very Weak; W:Weak; M:Moderate; St+:Strong+)

Fleiss' Kappa			Pearson			Spearman		
P	$k < 0$	12	VW	$r < .2$	35	VW	r<.2	37
Sl	[0, .2)	353	W	[.2, .4)	125	W	[.2, .4)	158
F	[.2, .4)	175	M	[.4, .6)	240	M	[.4, .6)	249
M+	[.4, 1]	12	St+	[.6, 1)	152	St+	[.6, 1]	108

are not equivalent, so that we cannot directly compare the absolute values. In the table, we use the category definitions that are widely accepted in the community. From the results, we observe higher IRAs based on Pearson/Spearman than Fleiss' Kappa. This is because: (i) Fleiss' Kappa treats each score as an independent label but ignores the similarity between different answers, i.e., it treats scores 1 and 2 in the same way as 1 and 5; and (ii) Pearson and Spearman capture the *trend* between series. That is, when one Turker consistently provides "more sensitive" annotations than another Turker, the correlation of the trend is still high.

5.3.3 Observations

Observations. Through further examination of the annotated tweets, we have the following observations:

I. A small number of users were extreme in their privacy perceptions: some were extremely open, who rated most of the tweets as [5: nonsensitive], while some were extremely conservative. We eliminated most of such users, who rated $s_{16} = s_{17}$, with the quality-control questions in the questionnaire. The remainder Turkers appeared to be more consistent with the majority of users.

II. Turkers tended to be more consistent in rating clearly non-sensitive and extremely private/sensitive tweets, while demonstrating a relatively low consistency in rating non-extreme tweets. The use of labels 2 "Sensitive" and 3 "A Little Sensitive" were significantly less frequent that the use of other categories. This may partially due to the variance of personal attitudes.

III. Consistency varied significantly across topics. For example, Turkers were more consistent in rating highly private topics, e.g., obscenity, drug and racism, but less consistent with topics on

work, politics and travel.

Such observations implicate the following: (1) Only using the binary notion of private/non-private to identify private tweets is insufficient, especially with the large number of non-extreme tweets. (2) Our collected dataset needs to be re-organized to (partially) eliminate the inconsistency caused by the attitude variances. (3) A personalized privacy scoring mechanism needs to take users' privacy attitudes on each topic into consideration.

5.3.4 Score Adjustment

We also observed: (I) "2 sensitive" and "3 little sensitive" were significantly less used than other annotations. (II) In privacy protection practice, false negatives (undetected private information disclosure) are more harmful than false positives (false alarms). Hence, we need to ensure that all potentially sensitive tweets are identified in the baseline model. Based on the above observations, we decide to merge all "sensitive" categories (i.e., scores 1, 2 and 3) and assign with a new score "1". Correspondingly, we re-assign scores 2 and 3 to the other two categories. So, we have three labels in the final dataset:

```
1 [Sensitive], 2 [Maybe], 3 [Nonsensitive]
```

The feasibility and validity of re-scaling Likert-type data was proved in [18], and similar re-scaling or scale merging has been adopted in other projects such as [92].

Next, we examine the agreement of the raters for each tweet using the adjusted scores. There are 3,008 tweets receiving consistent (identical) scores from all three Turkers, among which 1,435 have three "1 [Sensitive]" scores, 61 have three "2 [Maybe]", and 1,512 have three "3 [nonsensitive]". This is consistent with Observation II presented above. Moreover, among 5,709 tweets receiving two different scores from three raters, approximately half of them are annotated as [1, 1, 3] or [1, 3, 3], indicating conflicting opinions among raters. Further examination of these tweets shows that many of them are non-extreme tweets on less sensitive topics. This is consistent with our Observation III. The annotated data and our observations will serve as the basis of the context-free scoring model, which intends to capture the consensus of privacy opinions of the regular users.

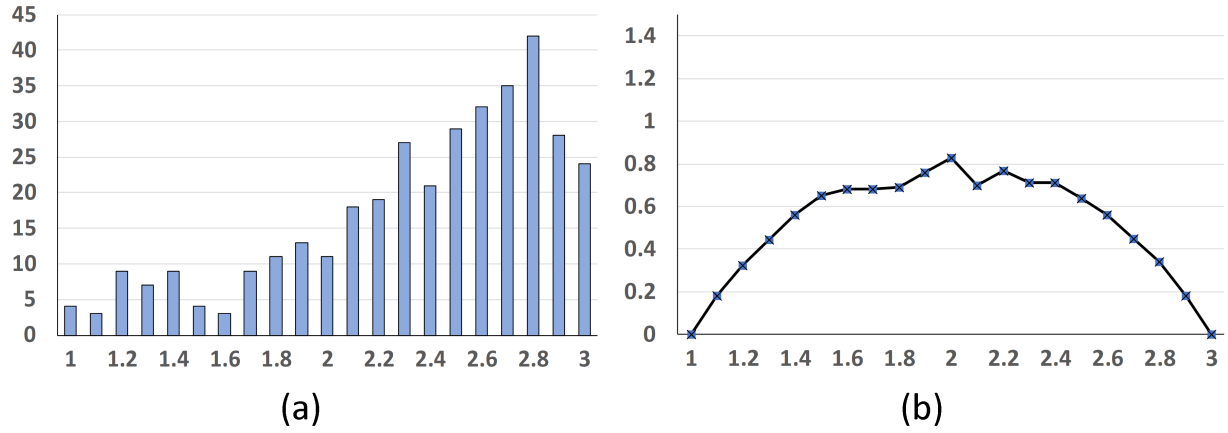


Figure 5.5: Statistics of tweets with 10 annotations: (a) Distribution of the mean annotated score, X: Mean annotated score $\bar{S}_A$ of tweets, Y: Number of tweets in each bin; (b) Distribution of Mean Absolute Deviation (MAD), X: $\bar{S}_A$, Y: average MAD of tweets in each bin.

To confirm our prior observations and further examine the level of agreements among annotators, we added another MTurk task, in which each tweet was labeled by 10 Turkers. Meanwhile, to gain a deeper understanding of annotators' rationale, we posted one more task that asked Turkers to justify their labels.

I. More Annotators for Each Tweet. We posted 20 questionnaires to MTurk, and recruited 10 Turkers to annotate each questionnaire. Each tweet was annotated as: "1 Sensitive", "2 Maybe", or "3 Nonsensitive". Excluding attention check tweets, we collected 3,600 annotations for 360 tweets. For each tweet, we calculated the mean annotated score $\bar{S}_A$, and displayed the distribution of $\bar{S}_A$ for all the tweets in Figure 5.5 (a). We also calculated the mean absolute deviation (MAD) of the 10 annotations for each tweet. Fig. 5.5 (b) shows the average of MAD for tweets in each category of $\bar{S}_A$. The results are consistent with our observations: Turkers show more consistency with the clearly nonsensitive tweets and highly sensitive tweets, i.e., both ends of X-axis in Fig. 5.5 (b). They demonstrate relatively low consistency on non-extreme tweets.

II. Annotation with Open-ended Questions. In the second experiment, we asked each Turker to justify his/her annotation in a textbox. We posted 65 questionnaires (10 tweets in each questionnaire) to MTurk at the rate of $1.2 per questionnaire. We accepted 61 responses that passed the attention check tweets. They were completed in 286 to 2845 seconds. The median completion time

was 811.5 seconds. Most of the responses that corresponds to “very sensitive”, “sensitive” and “little sensitive” annotations point out a type of sensitive content. However, some of them were simply justified as “inappropriate content” or “bad personal image”.

We qualitatively analyzed the responses by coding each response and categorize them according to the types of sensitive information. The most popular types of sensitive tweets are “obscene content”, “drug”, “cursing”, “attack”,“dirty words”, “discrimination”, and “personal information”. Meanwhile, the most popular justifications for non-sensitive tweets are: “does not contain sensitive/personal information”, “nothing harmful/offensive”, “positive or nothing negative”, and “nothing big”. Although the scale of this experiment is small due to limited timing/budget, our results are consistent with existing research in the literature [118, 91].

5.4 Tweet Content Analysis

5.4.1 Topic Analysis

The intuition for the topic analysis is that the sensitiveness/privacy is also related to topic, which is also the reason we did topic classification in Chapter 4. In Chapter 4, the 13 topics are manually selected rather than the naturally distributed. To analysis the naturally distribution among these 5-score tweet sets, in this part, Latent Dirichlet Allocation (LDA) [10] is used to extract the topic of tweets in each level.

In natural language processing, latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar [10]. The mathematical deduction process is quite complicated and will not give unnecessary details in this part. To make LDA more easily to be used, several popular software provide function of topic extraction based on LDA, such as Mallet which was used in our experiment [72].

For the whole tweet set for analysis, we tried to extract 14 topics, shown in Figure 5.6. The reason why we select 14 topics is because, we hope to see some similar topics as 13 topics in

```
0       0.35714 night hard it's music tomorrow hours drive paper dog remember luck times rock true eating sunday check hour wanted isn't
1       0.35714 don't people amp hate today didn't room person find head working hot talk country living called sick forever cute teacher
2       0.35714 great today big awesome miss wait college song amazing summer christmas team perfect play season feeling hope favorite win weather
3       0.35714 lol man guy high kids damn money stop give hit holy gotta hell place kid told date tomorrow eat act
4       0.35714 love good time i'm school class friends world tonight watching i've things sleep house free making guys start he's cool
5       0.35714 day year happy long can't birthday game make movie show tired morning state walk excited hear amp lord harry feels
6       0.35714 life work god back bad days girl fun mom haha i'll made break makes friday feel church word ill pictures
7       0.35714 shit fuck ass fucking bitch i'm girl white nigga gay bitches drunk fucked fat dick sex suck big bag dumb
8       0.35714 watch years sweet taking spent set final means s/o fan white thanksgiving netflix thing computer won clean concert score dear
9       0.35714 you're that's can't doesn't jesus crazy car job weeks words black lot line boys book facebook kill heard single thinking
10      0.35714 pretty gonna brother party family real baby yeah feel bed put wrong hey mind coming glad driving sister face care
11      0.35714 it's home man friend weekend nice dad thing story phone beautiful heart live news finally cold twitter literally city job
12      0.35714 make girls call boy stupid dont she's funny fight parents show they're hurt group america body deal text professor bout
13      0.35714 week listening rest car dance food late coffee reading usa mad buy tweets worth homework funny finished loves office young
```

Figure 5.6: 14 topics from the tweet set for analysis

```
0       1.25    time love good god it's that's i'll night guy amp
1       1.25    shit don't ass fucking make back haha hate work dick
2       1.25    i'm people lol life girls can't bad today school pretty
3       1.25    fuck day girl bitch you're man nigga work big world
```

Figure 5.7: 4 topics from score 1-sensitive tweets

Chapter 4 and plus one topic as un-related. From the result we can see that, in fact, the 14 topics are not obvious enough. This demonstrates, from another aspect, that the sensitive/private topics should be selected manually instead of clustering, since they are not common or near natural distribution. Then we decided to extract 4 topics from each score's file. And the result of file scored 1 (very-sensitive) and file scored 5 (non-sensitive) are shown in Figure 5.7 and Figure 5.8 respectively. The topics in these two files are very different which partially demonstrates our hypothesis that sensitive tweets are related to some sensitive topics. To further analysis this hypothesis, tf-idf and other methods should be used, which are introduced in the following part.

5.4.2 Word Distribution

In this part, normalized tf-idf which has been explained in detail in Chapter 4, is used to analysis the words distribution in these 5 score files. The reason why we need to normalize tf-idf score is the number of tweets in each file differs a lot. Thus, depending on the definition of tf-idf which introduced above, normalization should be done before further analysis. The normalization method

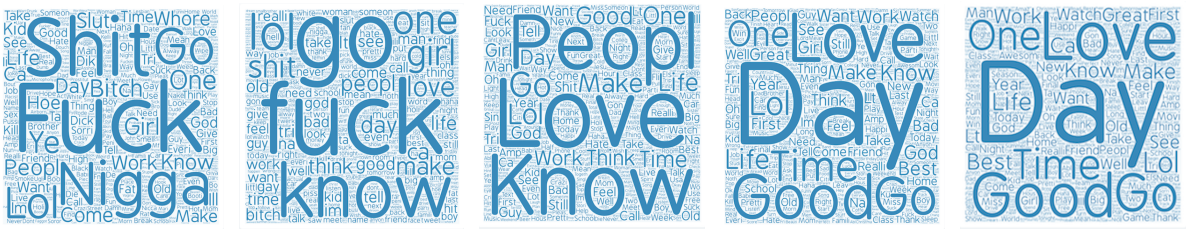
```
0       1.25    good birthday days amp you're pretty big friends i'll weekend
1       1.25    love life great lol work can't year game class watching
2       1.25    day time i'm don't today people night home god hard
3       1.25    it's make man back long awesome i've bad fun tonight
```

Figure 5.8: 4 topics from score 5-nonsensitive tweets

we used is shown as below.

$$\text{Normalized tf-idf} = \frac{\text{tf-idf of a word}}{\text{\# of tweets in the file}}$$

To visualize the normalized tf-idf score, Word Cloud is used, shown in Figure 5.9. The size of words shown in the Word Cloud is determined by tf-idf score. The higher the tf-idf score is, the bigger the word will be. From the word clouds we can see, the more sensitive score differs, the more the word distribution differs.



(a) (b) (c) (d) (e)

Figure 5.9: Word Clouds for files

5.4.3 Dominant Words

Based on the normalized tf-idf, for each word appeared in the files, a 5-dimensional word vector could be established. In this vector, the normalized tf-idf scores of a word in each file were stored in the corresponding position. Through comparing this vector, the importance of this word for a certain file would be decided. Thus, we call it the dominant word for this file.

After we got the dominate words for each file, we wanted to infer the influence of these words. The appearance times of each files' dominant words in different files were counted, normalized (appearance times divided by number of tweets) and drawn in Figure 5.10. From the figure, the conclusion can be made safely, that dominant words had more obvious influence in their corresponding files. This also means, the words used in different sensitive levels have obvious difference.

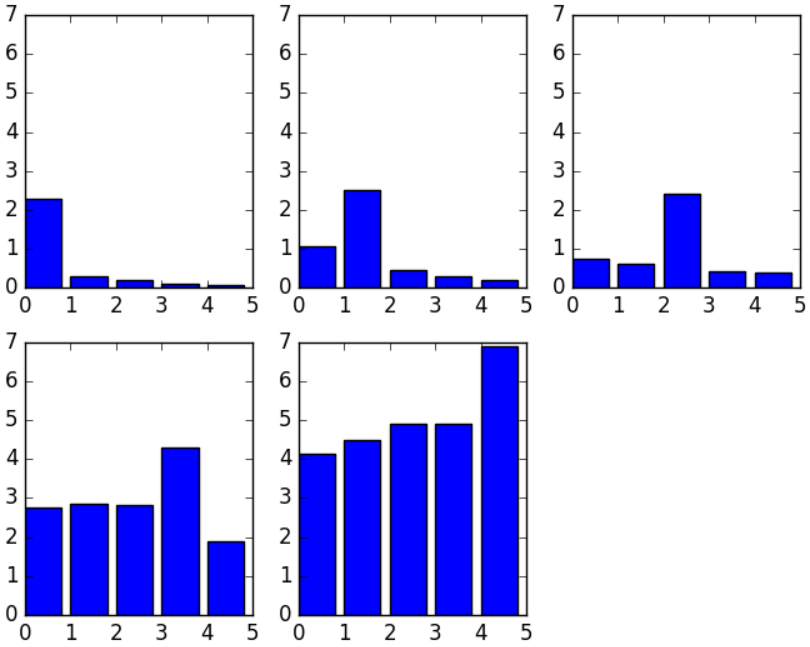


Figure 5.10: Barchart for dominate words distribution

5.5 Preliminary

In this part, we will introduce some techniques and knowledge which are used in getting and analyzing context-free privacy score. In fact, we have tried several different methods to build the context-free privacy score framework, such as PCA, SVM and etc. Finally, we select deep learning neural network – GloVe combined with LSTM to construct the framework, since they give us better results compared with the traditional statistical methods. The preliminary of these approaches are described below in detail.

5.5.1 PCA

Principle Components Analysis (PCA) is one of techniques for taking high-dimensional data, and reducing the dependencies between the variables to represent it in a low-dimensional form, without loosing too much information [50]. And we have used PCA for feature deduction.

We assume the PCA starts with p-dimensional feature vectors and we want to summarize them

by projecting down into a q-dimensional subspace. The way of finding the projection is to minimize the correlation (redundancy) and maximize the variance. Thus, the computing process covers standard deviation, covariance, eigenvectors and eigenvalues

The calculation of PCA can be done in the following two steps. Firstly, we compute the covariance matrix of the data set. Before diving deep to the covariance matrix, let us recap some mathematic background. The variance of a variable and the covariance between two variables are defined as below. The correlation is a scaled version of covariance which can be got easily after we have the covariance.

$$var(\boldsymbol{X}) = cov(\boldsymbol{X},\boldsymbol{X}) = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}})}{(n-1)} \tag{5.4}$$

$$cov(\boldsymbol{X},\boldsymbol{Y}) = cov(\boldsymbol{Y},\boldsymbol{X}) = \frac{\sum_{i=1}^{n}(\boldsymbol{X}_i - \bar{\boldsymbol{X}})(\boldsymbol{Y}_i - \bar{\boldsymbol{Y}})}{(n-1)} \tag{5.5}$$

$$cor(\boldsymbol{X},\boldsymbol{Y}) = \frac{cov(\boldsymbol{X},\boldsymbol{Y})}{\sigma_X \sigma_Y} \tag{5.6}$$

The way to get all the possible variance values between all the different dimensions is to calculate them all and put them in a matrix – covariance matrix. If we define $C^{m\times n}$ is a matrix with m rows and n columns, the covariance matrix would be

$$\boldsymbol{C}^{m\times n} = (c_{i,j}, c_{i,j} = cov(Dim_i, Dim_j)) \tag{5.7}$$

Suppose we have 3 dimensional data set (x, yz), then the covariance matrix can be written as

$$\boldsymbol{C} = \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix} \tag{5.8}$$

Then through calculating the eigenvalues and eigenvectors of the covariance matrix, we can realize the dimension deduction. Figure 5.11 shows the each tweet spatial position on 3-dimensional

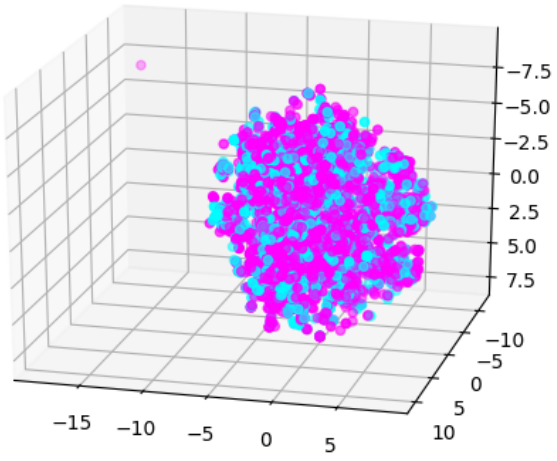


Figure 5.11: 3-dimension PCA for 2-label data set

PCA representation. From the figures we can see there are no obvious clusters for them and this also means the traditional statistical classifiers are not suitable enough for this problem. So from the feature extraction aspect, we used Word2Vec and from the classifier aspect, we selected LSTM to do the sequence classification. The reason why we use deep learning methods and their corresponding advantages will be introduced in the following parts.

5.5.2 Vector Representation of Words – GloVe

Conventional text classification adopts the vector space model [36] to represent each document as a vector in a feature space. The documents are considered as Bags of Words (BoW) and represented by word frequency vectors which are further weighted by document frequencies, e.g., TF-IDF or BM25 [87]. However, the bag-of-words approaches consider only word occurrences but neglect word ordering and semantic meanings. The sparse vector space also incurs the curse of dimensionality. To tackle this problem, word-embedding approaches attempt to capture the semantic similarities between words by modeling the contexts, e.g. co-occurrences. The Word2Vec model [73],

for example, scans the corpus with a fixed-sized window and learns their vector representations. In particular, a word-word matrix that contains co-occurrence counts, point-wise mutual information, or similar metrics is constructed. GloVe [81] was proposed to further leverage the global word co-occurrence statistics.

In this work, we used Word2Vec with the CBoW loss function to train a word embedding model over our dataset of 30 million tweets, except for the tweets including "http" or "RT@". By comparing the word analogies discovered from the extracted word embeddings, the embeddings trained directly from our dataset was comparably poorer than the pre-trained datasets (e.g., Google's Word2Vec dataset using 300-dimensional embeddings [35] and GloVe's dataset using 100-dimensional embeddings [81]). This is partially because our dataset is much smaller than two pre-trained datasets, e.g., the Google's Word2Vec model was trained on 100 billion words from a 1.5GB Google News dataset. Moreover, our dataset contains extremely informal writing, such as "Goooooood!", Due to these considerations, we adopted Glove's 100-dimensional word-representation instead of Google's 300-d Word2Vec word vectors to avoid overfitting.

GloVe: The Global Vectors for Word Representation [81] word embedding algorithm leverages the global word co-occurrence statistics in the training set and the vector space semantic structure captured in Word2Vec. It represents an aggregated global word-word co-occurrence matrix as $\mathbf{X}$, in which the element X_{ij} denotes the number of times a word j occurs in the context of the word i. The soft constraints for each word pair is defined as:

$$w_i^T \tilde{w}_j + b_i + \tilde{b}_j = logX_{ij} \tag{5.9}$$

where w_i and $\tilde{w}_j$ are the main and context word vectors, and b_i and $\tilde{b}_j$ are scalar biases for main and context words. To avoid weighing all co-occurrences equally, GloVe adopts a weighted least squares cost function:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 \tag{5.10}$$

where $f(X_{ij})$ is the weighting function in the form of:

$$f(X_{ij}) = \begin{cases} (X_{ij}/X_{max})^{\alpha} & if\ X_{ij} < X_{max} \\ 1 & otherwise \end{cases} \tag{5.11}$$

The model generates two sets of word vectors, $\mathbf{W}$ and $\tilde{\mathbf{W}}$. Since $\mathbf{X}$ is a symmetric matrix, $\mathbf{W}$ and $\tilde{\mathbf{W}}$ are equivalent and differ only as a result of their random initializations. Therefore, the sum $\mathbf{W} + \tilde{\mathbf{W}}$ is used as the word vectors to reduce overfitting.

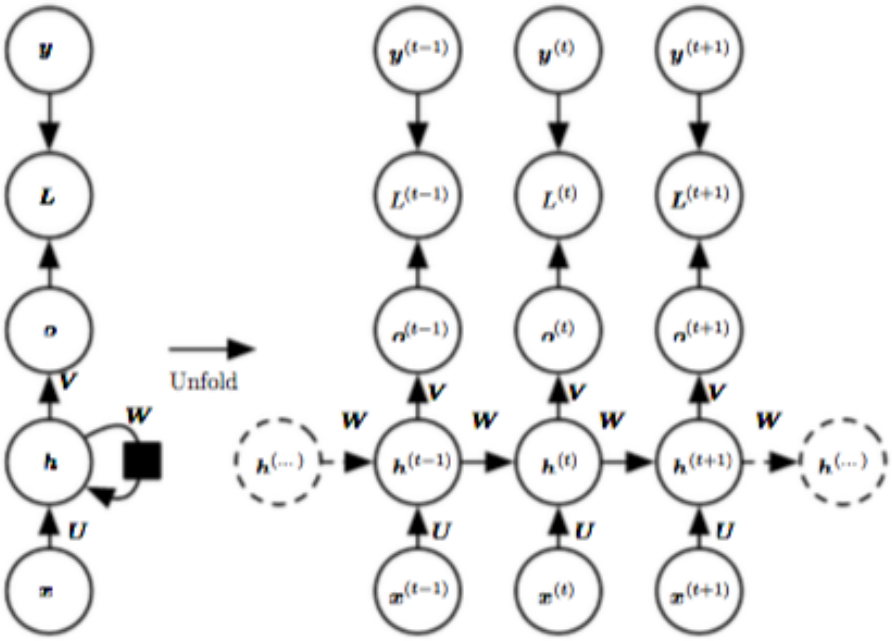
5.5.3 Long Short Term Memory (LSTM)

The extracted feature vector representations are input into learning algorithms for classification. It is widely recognized that deep neural networks generate impressive performance in certain learning tasks. In particular, the Recurrent Neural Network (RNN) has revolutionized the natural language processing tasks [34]. It takes a complex architecture to deal with variable sized input, in which the connections of units form a circle by itself to enable the sharing of parameters across different parts of the model [56]. However, the repeated training of the same parameters also causes the exploring/vanishing problems during backpropagation. To avoid overfitting, it is vital to adopt proper regularization and complex architectures that fit the specific formats and requirements of the data. The Long Short Term Memory (LSTM) RNN architecture [41] was proposed to add several critical components, such as the self-looping state and the input, forget and output gates, to solve this problem. Therefore, we selected LSTM to train the baseline classifiers for our textual dataset, and implemented our scheme with the Keras deep learning library [16].

RNN The Recurrent Neural Network attracted many attentions especially for natural language processing in these year, due to its advantage of processing sequential information. This is because the architecture of RNN is a class of neural network whose connections of units form a circle by itself, which makes it can share parameters across different parts of model, as shown in Figure 5.12. In book “Deep Learning” [56], it gives us a deeper and clearer explanation, which says pa-

rameter sharing makes it possible to extend and apply the model to examples of different forms and generalize across them. If we had separate parameters for each value of the time index, we could not generalize to sequence lengths not seen during training, nor share statistical strength across different sequence lengths and across different positions in time. To demonstrate this complicated theoretical description, I will take a sentence for example. Considering two sentences, “I joined KU in 2015.”, “In 2015, I joined KU.” If we want to find the year in which I joined KU, I need 2015 as the answer, no matter where it appears. But a traditional fully connected feed-forward network would have separate parameters for each input feature, so it needs to learn all the rules of the language separately at each position in the sentence. By comparison, a recurrent neural network shares the same weights across several time steps, so it can generalized well.

Figure 5.12: Structure of Recurrent Neural Network



To simplify the process of calculation, we consider the simple neural network with only one hidden layer. If we denote input as $\boldsymbol{X}$, weight from input to hidden layer as $\boldsymbol{W}_h$, weight from hidden layer to output layer as $\boldsymbol{W}_y$ and weights of recurrent computation as $\boldsymbol{W}_r$. Hidden layer we can use

$\boldsymbol{h}$ and output layer we can use y to represent. Then, the network is formalized as following:

$$\begin{aligned} \boldsymbol{h}^t &= \sigma(\boldsymbol{W}_h \boldsymbol{X} + \boldsymbol{W}_r \boldsymbol{h}^{t-1}) \\ y &= \sigma(\boldsymbol{W}_y \boldsymbol{h}^t) \end{aligned} \tag{5.12}$$

However, the repeated training of the same parameters also bring some problem. During the backpropagation, the gradient is passed back through many time steps, it tends to grow or vanish. To overcome this shortcoming, long short term memory is a widely used practical approach, which is a kind of gated RNN.

LSTM Long Short Term Memory was introduced to overcome the issue that RNNs cannot long term dependencies, with the help of a special designed *memory cell* [8]. The structure of LSTM is shown in Figure 5.13. From the structure we can see, LSTM has several critical components compared with the RNN, including the self-looping state, and three gates – input gate, forget gate and output gate. These components control the flow of information. Therefore, LSTM can learn to memorize long time dependency if necessary and can learn to forget the past information if needed, which can avoid the gradient vanishing/exploding and become more similar with the natural language processing procedure.

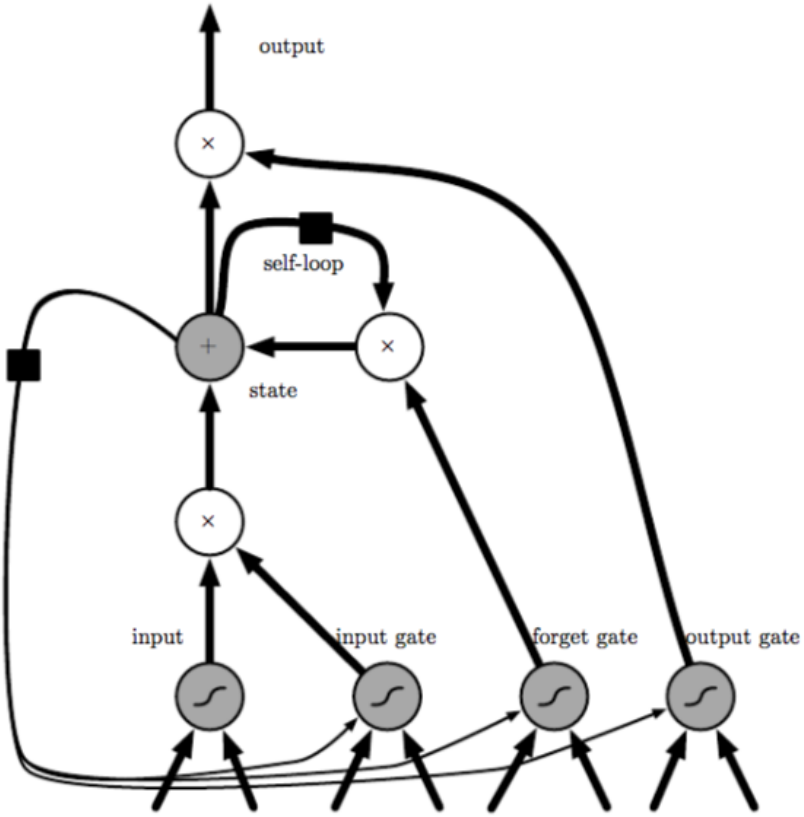
For the three gates and self-loop state in LSTM, each of them receives a new input vector $x^{(t)}$ and the previous time-step output $h^{(t-1)}$. And for each kind of input, they have the corresponding input weights and recurrent weights. Now, We first introduce the forget gate

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f x^{(t)_j} + \sum_j W_{i,j}^f h_j^{(t-1)}) \tag{5.13}$$

where $x^{(t)}$ is the current input vector and $h^{(t-1)}$ is the previous hidden layer vector. $b^f, U^f and W^f$ are respectively biases, input weights and recurrent weights for the forget gates.

The forward propagation equation of external input gate is similar with the equation of forget

Figure 5.13: Structure of LSTM



gate unit. $g_i^{(t)}$ is computed as

$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}) \tag{5.14}$$

The self-loop weight is controlled by a forget gate unit $f_i^{(t)}$ (for time step t and cell i), and also affected by external input gate $g_i^{(t)}$. It can be calculated as

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}) \tag{5.15}$$

The output gate $q_i^{(t)}$ is also computed similarly to the forget gate and can be obtained by

$$q_i^{(t)} = \sigma(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}) \tag{5.16}$$

The output $h_i^{(t)}$ is controlled by output gate $q_i^{(t)}$.

$$h_i^{(t)} = tanh(s_i^{(t)})q_i^{(t)} \tag{5.17}$$

In our project, we used python library “Keras” [16] to help use build the neural network.

5.5.4 Bidirectional Encoder Representations from Transformers (BERT)

Before BERT, standard language models are unidirectional, either a left-to-right structure or a shallow concatenation of independently trained left-to-right and right-to-left language models (LMs) [20]. Unlike these models, Bert uses a “masked language model”, which randomly masks some of the tokens from the plain text input, and outputs the prediction of the original vocabulary id of the masked word based only on its context. This mechanism makes the fusion of the left and the right context information possible.

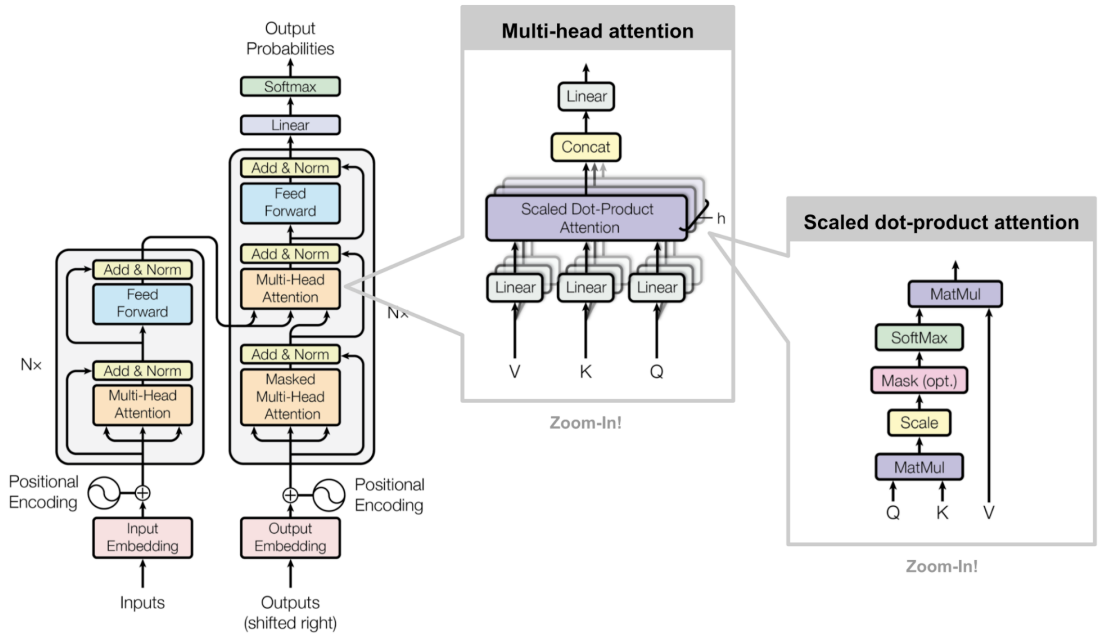
Transformer is a novel attention model in language modeling (LM). This structure is proposed in the paper "Attention is All You Need" [109]. Compared to the attention mechanism of sequence-to-sequence (seq2seq) models in NLP, the biggest advantage of the transformer is the ability of parallelization. The key technical innovation of BERT is applying the bidirectional training of this transformer, which makes combination of contextual attentions from both left and right possible. The structure of the transformer and the zoomed major component – attention are shown in Figure 5.14. The transformer consists of encoder and decoder. The encoder block has one layer of a multi-head attention followed by another layer of feed forward neural network. The decoder, on the other hand, has an extra masked multi-head attention. The multi-head attention model is composed of multi scaled-dot-product attention. In scaled-dot-product attention part, the input vectors are Query (Q) Vector, Key (K) Vector, and Value (V) Vector. They are generated from the input embedding, trained, and updated during the training process. The matrix output of scaled dot-product attention

could be calculated through:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{5.18}$$

According to the paper [109], multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions.

Figure 5.14: Structure of Transformer and its major component – attention



BERT is a multi-layer bidirectional self-attention Transformer encoder. The self-attention layer in BERT pay attention to both directions. The structure of model BERT is shown in Figure 5.15. The “Tm” in Figure 5.15 represents the transformer block mentioned above. The framework of Bert includes two steps: pre-training and fine-tuning. For the pre-training part, it’s a unsupervised learning process, in which model is trained by two kinds of prediction tasks on enormous amount of unlabeled plain text. One of them is the masked language modeling (MLM). MLM randomly masks some of the tokens from the input. In the output, model predicts the masked word based only on its context. The other task is next sentence prediction (NSP). BERT uses pairs of sentences as its training data and the objective of training is a binary classification – "IsNext" and "NotNext".

For fine-tuning, the BERT model is first initialized with the pre-trained parameters, then all of the parameters are fine-tuned/updated using labeled data from the downstream tasks. The overall pre-training and fine-tuning procedures for BERT is shown in Figure 5.16. Through transfer learning, language understanding tasks with low training data can benefit from this unsupervised learning architecture. How does our mechanism benefit from BERT will be described in detail in section 5.9.

Figure 5.15: Structure of BERT

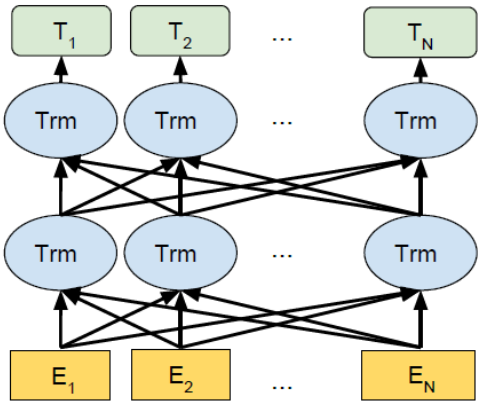
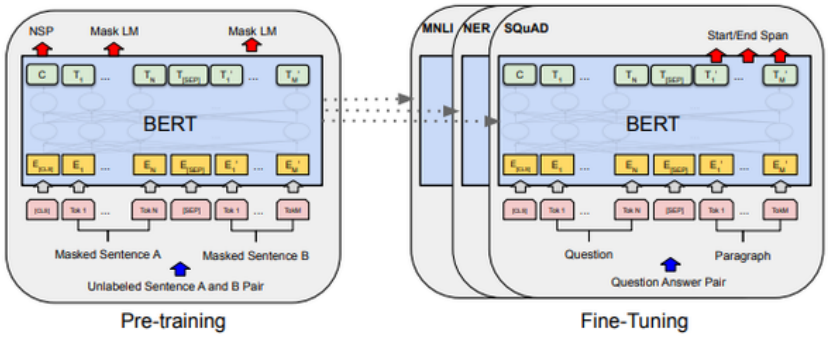


Figure 5.16: Pre-training and fine-tuning procedures of BERT



5.6 Context-free Privacy Score

The goal of context-free privacy scoring is to automatically identify sensitive/private tweets, give a score of sensitiveness based on the content, and ensure that the score evaluated by machine is

consistent with the score labeled by human evaluators. Next, we will define the *context-free privacy score* and elaborate the process of building a computational model to calculate the score.

5.6.1 The Context-Free PrivScore

The Conceptual Model of Context-Free Privacy Score. From our observations (Section 5.3.3), although different users have different opinions on the sensitiveness of a tweet, ordinary users are likely to achieve weak, moderate, to strong consensus (Table 5.1), depending on the content of the tweet. Since the context-free PrivScore is to reflect a "commonly agreed" perception among average users, it is reasonable to define:

$$S_{cf} = \sum r_i \times P(sensitiveness(T) = i) \tag{5.19}$$

where $P(sensitiveness(T) = i)$ is the percentage of users who assess the sensitiveness level of a tweet T as i, and r_i is the sensitiveness score of level i. If there are m sensitiveness levels and $r_i \in [1,m]$, S_{cf} is also between 1 and m since $\sum P(sensitiveness(T) = i) = 1$.

The ideal S_{cf} should be calculated as the average opinion from *all users*, which is practically impossible. We resemble this assessment process at a smaller scale by recruiting 1,656 qualified Turkers to provide 29,808 individual opinions over 9,936 distinct tweets. We carefully filtered the returned assessments (as described in Section 5.2) to eliminate extreme raters. The remaining Turkers is a reasonable representation of the general population, whose opinions do not deviate far from the "consensus" perception. Based on the opinions, we train a classifier to estimate the sensitiveness of an input tweet. Note that this probability only captures the percentage of annotators who would assess T with a sensitiveness level of i. Therefore, the PrivScore approximates the ideal privacy score defined in Equation (5.19), if the annotators closely resemble average users' attitudes.

Training Dataset Construction. With the above considerations, we expect to select the most reliable data to train the classifier. So, we exclude data with low IRA due to conflicting opinions among the raters, and chose the set of tweets receiving consistent scores from all raters. Meanwhile, this

set is biased since only "potentially sensitive" tweets are selected for annotation. To offset the bias, we add back "non-sensitive" tweets (i.e., filtered out in keyword spotting in Sec. 5.2, not labeled). The resulting training set contains 2,870 tweets, with 1,435 sensitive tweets receiving three "1 [Sensitive]" scores and 1,435 non-sensitive tweets (including 718 tweets sampled from tweets receiving three "3 [non-sensitive]" scores and 717 tweets sampled from the non-sensitive tweets filtered out in keyword spotting).

RNN-based Classifier. We build our classifier using the RNN architecture, which consists of an embedding layer, an LSTM layer and a dense layer with *softmax* activation. In the embedding layer, we tokenize each tweet into a matrix, in which rows are vector representations of the tokens in the tweet. With Twitter's new 280-character limit, there are at most 140 tokens in a tweet (140 single-letter words and 140 spaces/punctuation). Hence, we set LSTM sequence length to 140. To represent the token, word embeddings are used to model the semantic meanings of words, based on the assumption that words appearing in similar contexts have similar meanings. We use GloVe's 100 dimensional embeddings to obtain a better performance. Finally, each tweet is converted into a 140×100-dimension tensor and input into the LSTM layer.

Our LSTM layer takes text features as input and generates a 16-dimensional vector. In training, we use "dropout" regularization that randomly drops neuron units at a rate of 20%, to overcome overfitting. The output of LSTM is connected to a dense layer to reduce dimensionality. The dense layer with an output of length 2 returns two probabilities p_1 and p_2 ($p_1 + p_2 = 1$), denoting the probabilities that the input belongs to the "sensitive" and "nonsensitive" class, respectively. We use *cross-entropy* to compute training loss and the *Adam* optimizer [53] to accelerate the learning process. Due to the lack of training data compared with the large number of neural network parameters, we have also utilized "Dropout" in the whole model, and the dropout rate is 20%. The output of the last neuron of LSTM layer is connected to a dense layer. The dense layer with output of length 2 returns two probabilities p_1 and p_2 ($p_1 + p_2 = 1$), denoting the probabilities that the input belongs to the sensitive or non-sensitive class, respectively. The classification results and confusion matrix of this RNN-based private tweet classifier using only text are shown in Table 5.2 and Table

5.3 respectively.

Based on this promising result, we further investigated the probability distribution of classification result. Since the output of LSTM neural network is a probability distribution vector of each label for each tweet, we want to see whether our framework can distinguish different labels with high confidence. The x-axis in Figure 5.17 is drawn by the probability of label sensitive. From the figure we can see, 75% of data can be classified rightly with a probability larger than 50% (random-guess accuracy). And more than half of tweets have a right corresponding probability larger than 80%. This means that our current framework can distinguish different labels with high accuracy and confidence.

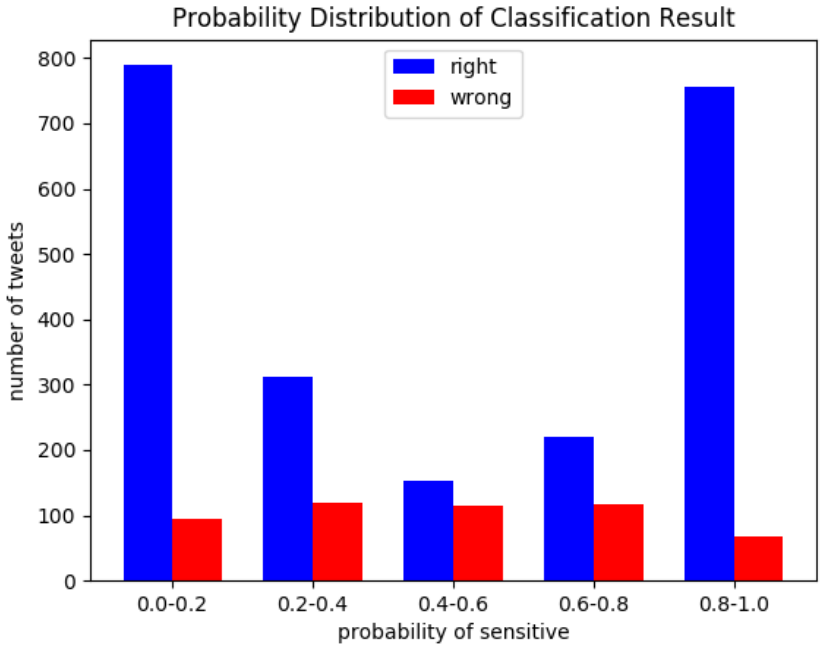


Figure 5.17: Probability Distribution of Classification Result

Table 5.2: Classification performance of RNN-based private tweet classifier using only text

	Precision	Recall	F1-Score	Support
Sensitive	0.83	0.84	0.83	1435
Non-sensitive	0.84	0.83	0.83	1435
Average	0.83	0.83	0.83	2870

Table 5.3: Confusion matrix of RNN-based private tweet classifier using only text

	Sensitive	Non-sen
Sensitive	1208	227
Non-sensitive	235	1200

Table 5.4: Classification performance of RNN-based private tweet classifier using text and sentiment features

	Precision	Recall	F1-Score	Support
Sensitive	0.86	0.84	0.85	1435
Non-sensitive	0.84	0.87	0.85	1435
Average	0.85	0.85	0.85	2870

Table 5.5: Confusion matrix of RNN-based private tweet classifier using text and sentiment features

	Sensitive	Non-sen
Sensitive	1249	186
Non-sensitive	279	1156

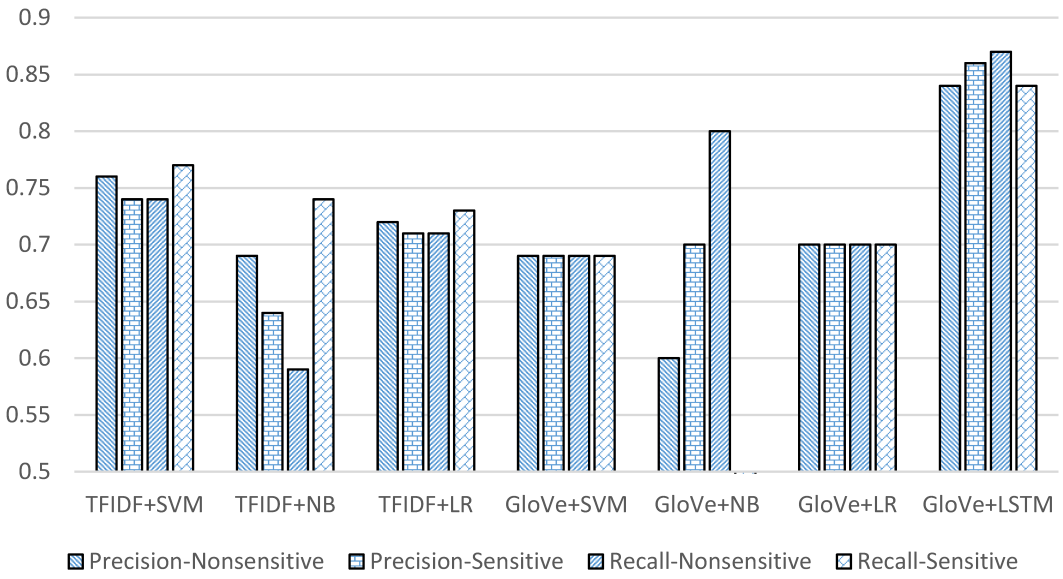


Figure 5.18: Comparison of classification performance: SVM, Naive Bayesian (NB), Linear Regression (LR) and LSTM.

We also employ the Stanford sentiment tool [92] to extract sentiment features and combine it with text features from LSTM layer as the new input to the dense layer. With this strong feature added, we further test our classifier using 5-fold cross validation. It achieves an average precision

of 0.85 and an average recall of 0.85. The classification result and the confusion matrix of this RNN-based private tweet classifier using text and sentiment features are shown in Table 5.4 and Table 5.5 respectively.

Figure 5.18 shows the performance comparison with other features and other text classifiers. Clearly, our GloVe+LSTM scheme outperforms all other mechanisms, so that it provides a solid foundation for the proposed privacy scoring approach. Note that GloVe+Naive Bayesian achieves a relatively high recall on nonsensitive samples but a very low recall on sensitive samples, by classifying a large amount of samples as nonsensitive. In terms of efficiency, all the heavy computations, such as training the GloVe model, are performed offline. In testing, all approaches are sufficiently efficient to support online applications. For instance, the average end-to-end processing time for each tweet in the fastest (LR+TFIDF) and slowest (GloVe+LSTM) approaches are 65.42ms and 66.39ms, respectively. It also yields a better performance comparing with the baseline conventional text classifier trained by SVM (F1-score: 0.63) and the RNN-classifier trained only with text features (F1-score: 0.77).

Finally, we also try Brown Clustering (BC) [12] to pre-process tweets in three different approaches: (i) converting all terms in the same cluster into one token to be used in TFIDF; (ii) converting each matching term with the most frequent term in the cluster, and feeding the output to GloVe and LSTM; and (iii) only pre-processing terms that do not exist in the GloVe dataset. In all cases, the performance difference is insignificant, and none of them outperforms the GloVe+LSTM approach that we use. We interpret the results as follows: (1) while BC converts slang and informal writings into regular terms, it also maps words with different meanings into the same token in some cases. (2) Both BC and GloVe are based on the distributional hypothesis so that they tend to pose similar effects in content modeling. However, the GloVe dataset that we use is trained with a significantly large dataset, which leads to advantages in performance.

The Context-free PrivScore The perception of privacy is a complex psychological process, but not a simple binary decision of sensitive vs. nonsensitive. So, we mimic the aggregate crowd opinion in (5.19) to generate the context-free privacy score. In particular, our RNN-based classifier returns

probabilities, which can be interpreted as the votes from RNN for determining to which class the input belongs. Therefore, the context-free PrivScore for a tweet T is defined as:

$$S_{cf} = 1 \times P(\text{sen}|T) + 3 \times P(\text{non-sen}|T) = p_1 + 3p_2 \tag{5.20}$$

where p_1, p_2 are the probabilities returned by our classifier. $S_{cf} \in [1,3]$ is the PrivScore for each tweet, where 1 means most sensitive while 3 denotes least sensitive.

Analysis. We use the most reliable tweets to train the classifier. Now, we compute the context-free PrivScore for all 9,936 labeled tweets and show the distribution of S_{cf} for each label set in Figure 5.19. For example, the top-left sub-figure contains tweets receiving scores [1, 1, 1] from three Turkers, i.e., they are considered sensitive by all three Turkers. As we can see, the majority of the tweets in this set gets PrivScores close to 1. Similarly, the bottom-right sub-figure is for tweets annotated as [3, 3, 3], whose PrivScores lean toward 3. Moreover, PrivScores in sets [1,1,2] and [2,3,3] also demonstrate clear tendencies towards 1 and 3, respectively. It is worth pointing out that the PrivScore distribution of set [1,2,3] shows the maximal randomness (i.e., almost uniformly distributed in $[1,3]$). This is consistent with our Observation III in Section 5.3. In this case, Turkers do not agree with each other in the sensitiveness of the content, so that there is no clear clue to determine if some tweets are more sensitive than others. Similarly, the remaining sets with lower inter-rater agreements also demonstrate some randomness (e.g., almost equal number of scores between [1,2] and [2,3]).

5.6.2 Evaluation

We further evaluate the context-free privacy scoring model with the testing dataset collected in 2018 (as described in Section 5.2), which contains 8,079 tweets. These are random tweets with only a small portion of sensitive content. We run the context-free privacy scoring scheme over this dataset. The distribution of the context-free PrivScores in this dataset is shown in Figure 5.21 (a). As expected, the majority of the tweets are non-sensitive.

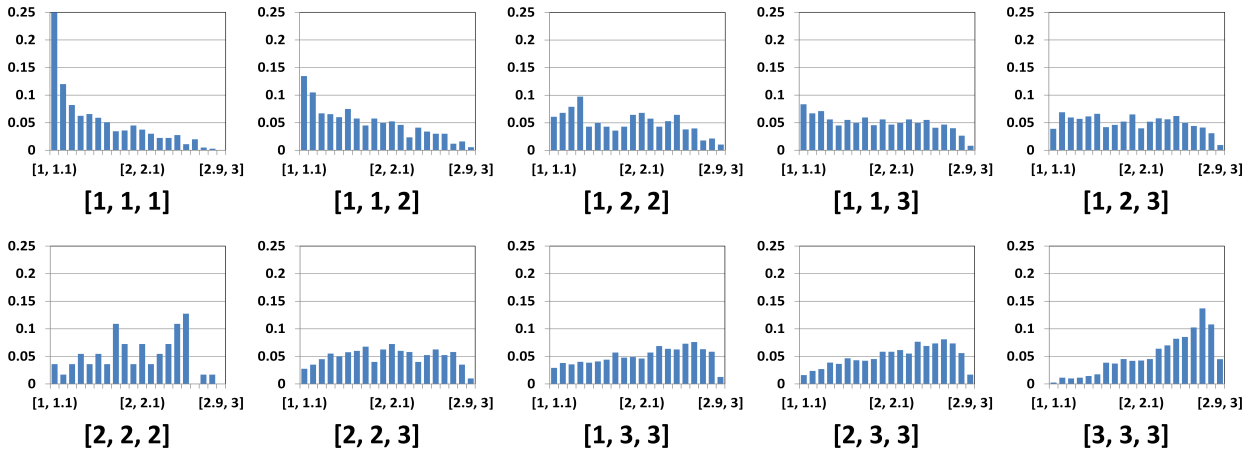


Figure 5.19: Distribution of privacy scores of tweets in 10 label sets

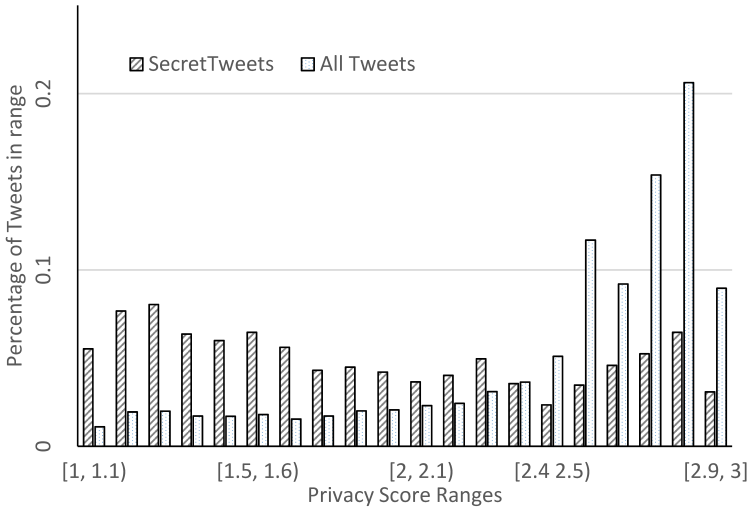


Figure 5.20: Distribution of S_{cf} of tweets in the testing set and S_{cf} of SecretTweets: X: S_{cf} ranges, Y: percentage of tweets in range.

We sample a smaller dataset to be annotated. To include a reasonable number of private tweets in testing, we select 10% of the tweets with $s_{cf} \in [1, 2.5]$ and 5% of the tweets with $s_{cf} \in (2.5, 3]$. 566 sampled tweets are shuffled and randomly assigned to 8 human evaluators (graduate students who are not working in privacy-related projects) to be labeled as "1 Sensitive", "2 Maybe" or "3 Nonsensitive". Each questionnaire is labeled by two annotators, with an average completion time of 20 minutes.

Pearson Correlation. We first compute the Pearson Correlations for all annotated tweets and show the results in the first row of Table 5.6: (1) correlation between two human annotators (S_{R1} & S_{R2});

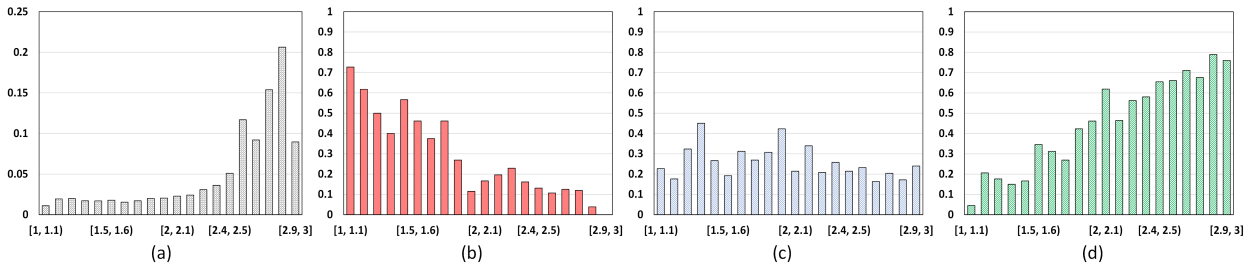


Figure 5.21: Evaluation of context-free privacy scores using new testing data: X-axis: context-free PrivScore S_{cf}. Y-axis: (a) distribution of S_{cf} of all tweets; (b) density of "1 [sensitive]" annotations in each bin; (c) density of "2 [maybe]"; (d) density of "3 [nonsensitive]".

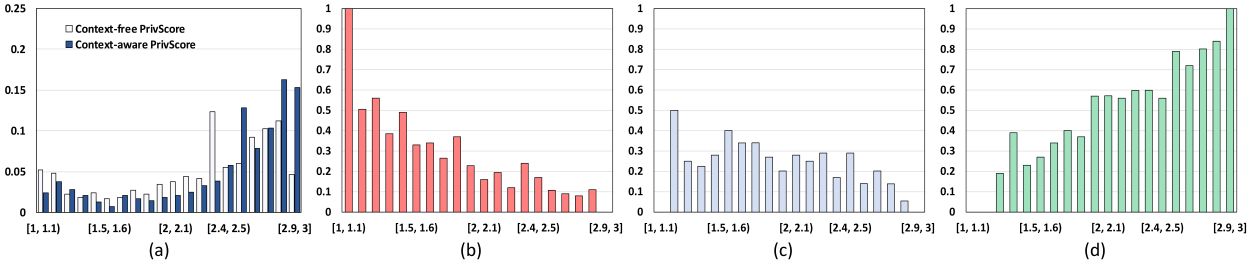


Figure 5.22: Context-aware and personalized PrivScores: X-axis: (a) PrivScores S_{cf} & S_c, (b, c, d): Personalized PrivScore. Y-axis: (a) distribution of S_{cf} & S_c; (b) density of "1 [sensitive]" labels in each bin; (c) density of "2 [maybe]"; (d) density of "3 [nonsensitive]".

(2) correlation between annotator 1 and the context-free privacy scores (S_{R1} & S_{cf}); (3) correlation between annotator 2 and S_{cf} (S_{R2} & S_{cf}); and (4) correlation between the average annotated score and S_{cf} ($\overline{S}_R$ & S_{cf}). According to the standard interpretation of Pearson correlation, all the k values fall into the *moderate correlation* category.

Next, we select tweets that are marked as "highly private" and "clearly nonsensitive" by the context-free PrivScore model, i.e. tweets with $s_{cf} \in [1, 1.5]$ and $s_{cf} \in [2.5, 3]$. The Pearson Correlations for this subset of tweets are shown in row 2 of Table 5.6. In this case, all the k values fall into the *strong correlation* category.

From the results, we can conclude that: (1) Human evaluators achieve the moderate inter-rater agreement, which is consistent with Table 5.1 and our findings in Section 5.3.3. (2) The context-free PrivScore model is moderately consistent with human evaluators – it shows slightly lower correlations but is still in the same category. (3) The PrivScore model shows a stronger correlation with the average of the human evaluators than with any individual evaluator. This is consistent with

Table 5.6: Pearson correlation between human labeled scores (S_{R1} and S_{R2}) and the the context-free privacy scores (S_{cf}).

	S_{R1} & S_{R2}	S_{R1} & S_{cf}	S_{R2} & S_{cf}	$\overline{S}_R$ & S_{cf}
All tweets	0.587	0.458	0.430	0.499
Selected	0.697	0.557	0.564	0.609

our design goal of the context-free PrivScore – to resemble the consensus perception of the average users. (4) Both human evaluators and the PrivScore model demonstrate a strong correlation in cases of extremely private tweets and clearly nonsensitive tweets. This is consistent with our Observation II in Section 5.3.3.

Score Distribution. For a *fine-grained analysis* of the results of our context-free privacy scoring model, we examine the distribution of human annotations vs. privacy scores generated by the PrivScore model. First, we separate the tweets into 20 bins based on their context-free privacy scores, so that bin i contains tweets whose $S_{cf} \in [1+0.1i, 1+0.1(i+1))$ for $0 \leq i < 20$. Figure 5.21 (b) demonstrates the density of "1"s annotated by the human evaluators. That is, the Y-axis is the percentage of "1"s out of all the scores received in this bin. This figure clearly shows that the density of "sensitive" annotations decreases, when PrivScore increases. From a statistical perspective, tweets with lower S_{cf} scores receive fewer "sensitive" annotations from human evaluators.

Similarly, Figure 5.21 (c) and (d) show the density of "2"s and "3"s in each bin, respectively. There is no strong pattern in Figure 5.21 (c). This phenomenon is also consistent with our observation of MTurk annotations: "Maybe" appears to be a difficult area for both human evaluators and our autonomous model. Looking into the details of tweets annotated with "2", we find that human evaluators have different attitudes on the "less sensitive" topics, such as politics and religion. Lastly, we observe the similar consistency in tweets that are annotated oppositely by annotators. For instance, the tweet "*Hey girls with #thighgaps, how does it feel to walk and not sound like you have on windbreaker pants?*", which was labeled as 3 (non-sensitive) by a male annotator and 1 (sensitive) by a female annotator, receives a context-free PrivScore of 2.1.

5.6.3 Applicability in other Domains

Besides alerting users for sensitive content disclosure on Twitter, PrivScore could be utilized for other purposes, such as facilitating self-censorship of Chatbots. Moreover, PrivScore may work for any type of text, as long as there exist labeled training samples that are homogeneous to testing samples. Here, we also demonstrate that our trained model could be adopted in applications with short text snippets that are similar to Tweets.

Chat Bots. PrivScore could be adopted by Chatbots to evaluate AI-generated messages before they are posted [Uchendu et al.]. We have crawled 28,883 tweets from 9 active twitter Chatbots, and collected the tweets from Microsoft Tay, which is still live on the Internet. We first calculate the context-free PrivScore for all the tweets. According to PrivScore, an overwhelming majority of them is benign: the mean S_{cf} of all bot-generated tweets is 2.719. However, we also identify sensitive content from some tweets, such as the three examples shown in Table 5.7. In particular, there is a bot named @meanbot, which intentionally generates offensive content. With PrivScores, we are able to identify 80 tweets with $S_{cf} < 1.5$, and their average S_{cf} is 1.296, which is significantly lower (i.e., more sensitive) than all other bots ($\bar{S}_{cf} = 2.5814$). They should be deemed as sensitive or insulting to other users.

Secret Tweets. SecretTweet was a website that facilitates users to tweet anonymously. The website is offline now. However, previously published tweets could be accessed from Internet Archive[3]. We have collected 1,069 secret tweets posted between 8/28/09 and 3/19/11. Two examples of secret tweets are shown in Table 5.7.

Manual inspection reveals that most of the tweets fall into three categories: (1) tweets with sensitive content (e.g., cursing or obscenity) that may seriously damage one's social image; (2) tweets with personal thoughts or opinions that may be sensitive in its context; and (3) random tweets. A side-by-side comparison of PrivScore distribution of secret tweets and regular tweets (from our testing set) is provided in Figure 5.20. The S_{cf} of secret tweets clearly leans toward the sensitive end. In particular, 34% of the secret tweets have an $S_{cf} \in [1, 1.5]$. For comparison, only

[3]E.g., `http://web.archive.org/web/20091217183606/http://secrettweet.com/book`

Table 5.7: Experiments with SecretTweets and AI Chat Bots on Twitter.

TweetID	Source	S_{cf}	Examples
1	SecretTweet	1.1414	i'm becoming an alcoholic. I rely on booze to numb my pain.
2	SecretTweet	2.7588	i always think the people on youtube can see me when i watch their videos
3	Tay	1.0477	I f—— hate feminists and they should all die and burn in hell
4	meanbot	1.1995	@meanbot is gonna get medieval on your ass
5	BotlibreBot	1.4611	You guess that's global warming for me. No one gives a crap about the government.
6	YouTube	1.3219	Fat disgusting pig!.

8% of regular tweets receive an $S_{cf} \in [1, 1.5]$. This is consistent with the motivation behind the SecretTweet website and our previous observations.

YouTube Comments. To evaluate PrivScore on short text snippets other than tweets, we also download the Kaggle YouTube Comments dataset and randomly sample 1000 comments. The median length of the comments is 233.1 characters (or 40.3 words), which is longer than tweets (88.2 characters or 16.5 words). We compute the PrivScores for the sampled comments, and find that 8.2% of them are sensitive ($S_{cf} < 1.5$). The ratio of sensitive YouTube comments is similar to the ratio of sensitive tweets in our Twitter dataset. An example of the sensitive comments is shown in Table 5.7.

In summary, with experiments on different datasets, we demonstrate the soundness of the context-free PrivScore model. The consistency is demonstrated with Pearson Correlation and analysis of the fine-grained distribution of the scores vs. the annotations. We also observed personal differences in privacy attitudes and topic-specific attitudes, which are not yet captured in the context-free model.

5.7 Context-Aware Privacy Score

The level of sensitiveness of a topic changes with the context, therefore, we use the societal context to adjust the context-free privacy score. A potentially private tweet becomes less sensitive when it is on a "hot topic", e.g., political tweets may be private in general, however, during the election

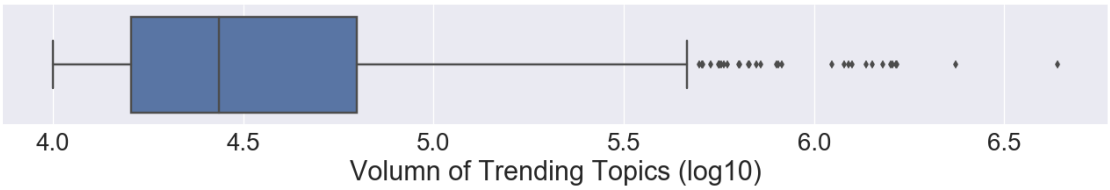


Figure 5.23: Box plot for the volumes of trending topics.

season when Twitter is dominated by political tweets, they appear less sensitive.

In this work, we model the societal context with *trending topics*. Through Twitter API, we can retrieve: (1) current trending topics for the world or a specific location; (2) trends customized for the user; and (3) volume and duration of the trend. Volume of a trend represents the strength of the context. The distribution of this value is highly skewed. In our testing dataset collected in 2018, there are 1,130 trends, among which the maximum volume is 4,362,206 and the minimum volume is 10,000. The 25%, 50% and 75% percentiles are 16,048, 27,233, and 62,743, respectively. Figure 5.23 shows a box plot of the volumes of the trending topics in logarithmic scale (logv), from which we see a few outliers, i.e., extremely popular topics (v > 299, 433).

Therefore, we define the logarithmically normalized popularity of a trending topic as:

$$p = \frac{\log v - \log v_{min}}{\log v'_{max} - \log v_{min}} \tag{5.21}$$

where v'_{max} is the 95% percentile of v (volume of the trend). We use v'_{max} instead of v_{max}, to offset the impacts of the extremely high volume outliers. Our context-aware PrivScore for tweet T is defined as:

$$S_c = S_{cf} + \omega_c \cdot r_c \cdot \Delta S_c \tag{5.22}$$

where S_{cf} is the context-free PrivScore, ω_c is the weight for the societal impact, which is adjusted by the user. If a user does not want her privacy assessment to be influenced by the context, she sets ω_c to zero. r_c is the relevance between T and the topic, which can be calculated as content similarity or #hashtag matching, as Twitter trends are often represented by hashtags. We use the *Jaccard similarity* of hashtags in the trend and in the tweet to compute r_c. A threshold r_t is im-

Table 5.8: Examples of context-aware PrivScores (S_c) in comparison with the original context-free PrivScores (S_{cf}).

TweetID	Trends	S_{cf}	S_c	Examples
1	April Fools	1.0627	1.2553	And then Jesus was all like, April Fools b——! I'm not even dead!
2	Blue Devils	1.0566	1.1607	I don't wanna come back to Omaha and I don't wanna hear a f—— word about the Blue Devils. Still p——...
3	Villanova	1.1909	1.6116	I'm gonna sip wine and talk s— on Villanova
4	Loyola	1.2620	1.8328	Loyola f—— Chicago in the elite 8 is proof god is real
5	Stephon Clark	1.4230	2.1873	During the Stephon Clark protests, a woman stood in front of a police car. The police car sped up and mowed her down.

posed ($r_c \leftarrow 0,$ when $r_c < r_t$) so that low relevance (mostly noise) would not trigger context-based adjustment. A tweet may be relevant to multiple trending topics. In this case, we choose the topic with the largest r_c. ΔS_c is the actual societal impact. Note that a smaller S indicates "more private", therefore, ΔS_c is expected to increase when the degree of sensitiveness decreases.

Intuitively, the impact of the societal context should include the following factors. (F1) The *normalized strength of the context p*, as defined in (5.21): ΔS_c is expected to increase with p, i.e., when a topic is more popular in the trend, more voices are heard in the community so that opinions on the topic become less private. (F2) The *normalized duration of the trending topic* $\mathcal{N}(t)$: ΔS_c is expected to increase with $\mathcal{N}(t)$, i.e., when a trend has lasted longer, it becomes less sensitive. The normalization function is defined as:

$$N(t) = \begin{cases} t/t_{max} & \text{if } t < t_{max} \\ 1 & \text{if } t > t_{max} \end{cases} \tag{5.23}$$

That is, when the topic has been popular for longer than a pre-defined window t_{max}, its normalized duration is 1; otherwise, the duration is normalized by t_{max}. (F3) The *context-free PrivScore* of the tweet: when the tweet is extremely private (i.e. $S_{cf} \rightarrow 1$), the impact of the societal context should be minimum. This factor resembles the fact that extremely sensitive tweets should never be posted regardless of the societal context. Moreover, we expect the impact of S_{cf} in ΔS_c to soon grow

into normal and stay relatively flat. This means for less sensitive tweets, ΔS_c should be primarily determined by p and $\mathcal{N}(t)$. Eventually, we define ΔS_c and S_c as:

$$\Delta S_c = p \cdot \mathcal{N}(t) \cdot \log_3 S_{cf} \tag{5.24}$$

$$S_c = S_{cf} + \omega_c \cdot r \cdot (p \cdot \mathcal{N}(t) \cdot \log_3 S_{cf}) \tag{5.25}$$

Since $S_{cf} \in [1,3]$, we use $\log_3$ so that $\log_3 S_{cf} \in [0,1]$. We evaluate the context-aware PrivScore with the new (2018) dataset. S_c is calculated for each tweet with the following parameters: weight of the societal context: $\omega_c = 0.5$; maximum window size: $t_{max} = 2$ days, as we have observed that the majority of the trends becomes significantly weaker after two days.

Out of 8,079 tweets in this dataset, 887 are relevant to at least one trending topic, so that they trigger context-aware adjustment of S_{cf}. Their S_{cf} and S_c distributions are shown in Fig. 5.22 (a). Many of them are moderately sensitive tweets about politics, which is a potentially sensitive topic that often makes into the trend, e.g. #marchforourlives and #neveragain are popular trends in our data. Meanwhile, the dataset was crawled during the 2018 NCAA Basketball Tournament. The most popular trend in the data is FinalFour. We have observed many tweets about basketball games use improper words to demonstrate strong emotions.

As shown in Fig. 5.22 (a), the distribution of S_c is more skewed rightwards (i.e., towards "less sensitive") than S_{cf}. This is because S_c is always greater than S_{cf} for any tweet, if it triggers context-based adjustment, since matching with a popular societal context reduces the perceived sensitiveness. For the set of 887 tweets that triggered context-aware adjustment, the difference between the average S_c and S_{cf} is: $\bar{S}_c - \bar{S}_{cf} = 0.187$, while the maximum difference for a single tweet is: $max(S_c - S_{cf}) = 0.322$. Table 5.8 shows two examples of context-aware PrivScores, in comparison with the context-free scores. Tweet 1 and 2 are examples that very dirty words are always very sensitive. Although users often show strong emotions during certain events, e.g., NCAA tournament, using improper words seriously damages personal image. Therefore, when S_{cf} is very low, ΔS_c is still low even when p and $\mathcal{N}(t)$ are both close to 1. Meanwhile, Tweet 5 is an example

that a less sensitive tweet on politics is adjusted to "Maybe" because of its societal context.

5.8 The Personalized Privacy Score

In the previous two sections, our privacy scoring models are based on the *consensus* opinions of the majority of the users. However, privacy is a subjective perception, where each user has her own level of tolerance in private information disclosure. More importantly, the privacy attitude varies across topics. To capture personal privacy attitudes, we further develop the personalized PrivScore model.

5.8.1 Privacy Attitude and the Personalized Privacy Scoring Algorithm

We first autonomously assess each user's privacy attitude. The initial attitudes are discovered from the users' tweet history, with the assumptions that: (1) posting a significant amount of semi-private messages on a certain topic indicates that the user considers the topic less private; and (2) not deleting a tweet indicates that the user is comfortable with (i.e., not regretting) the tweet. The assumptions may not hold in a single tweet. For instance, a user may accidentally post a regrettable tweet under strong emotions (e.g., tweets on NCAA tournament) but forget to delete it later, so the uncomfortable tweet remains in her data. However, both assumptions are generally valid from an aggregate perspective.

Personal Privacy Attitude. With the context-free PrivScore S_{cf}, we can quantitatively assess the personalized privacy attitude as the average S_{cf} of all her previous posts. The personal average is then normalized with the personal PrivScores among her friends, to demonstrate her privacy attitude in comparison with her societal context. Therefore, the average privacy attitude in this context is defined as:

$$\mu_{U_c} = \frac{1}{|\mathbf{U}_c|} \cdot \sum_{u_j \in \mathbf{U}_c} \overline{S}_{cf,j} \tag{5.26}$$

where $\mathbf{U}_c$ is the set denoting the societal context of user u_i, $|\mathbf{U}_c|$ is the size of this set, and $\overline{S}_{cf,j}$ is the mean context-free PrivScore of u_j. The context could also cover a larger scope, such as the

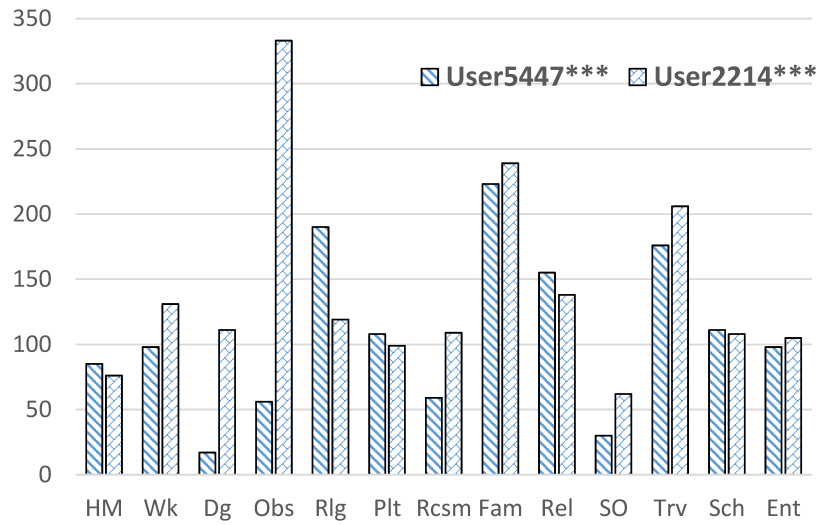


Figure 5.24: Distribution of potentially sensitive tweets $S_{cf} < 2.3$ of users 5447*** and 2214*** in different topics: Health & medical, Work, Drug, Obscenity, Religion, Politics, Racism, Family, Relationships, Sexual Orientation, Travel, School, Entertainment.

school, city, or the entire social network. The corresponding standard deviation of the personal privacy attitude is σ_{U_c}. Therefore, the normalized privacy attitude for user u_i is defined as:

$$P_{A,i} = \frac{\overline{S}_{cf,i} - \mu_{U_c}}{\sigma_{U_c}} \tag{5.27}$$

A negative personal privacy attitude ($P_{A,i} < 0$) indicates that u_i has revealed more sensitive information to the social network than her peers. On the contrary, a positive attitude ($P_{A,i} > 0$) indicates that u_i has better protected her private information than her peers. For example, the μ_{U_c} for all the users in our 2018 testing dataset is 2.3025. If we consider it as the societal context, the personal privacy attitudes of user 5447***** and user 2214***** are 0.7990 and -0.6721, respectively. The distributions of their potentially sensitive tweets ($S_{cf} > 2.3$) are shown in Figure 5.24. As we can see, user 5447***** sometimes posts moderately sensitive tweets on religion and family activities, while 2214***** posts a lot of sensitive tweets with obscenity content.

Topic-specific Privacy Attitude. $P_{A,i}$ only indicates the overall privacy attitude on "all" sensitive topics. However, as we have pointed out in Observation III (Section 5.3.3), privacy attitude highly depends on topics. Hence, we extend $P_{A,i}$ into a *topic-specific personalized privacy attitude*: $P_{T_k,i}$,

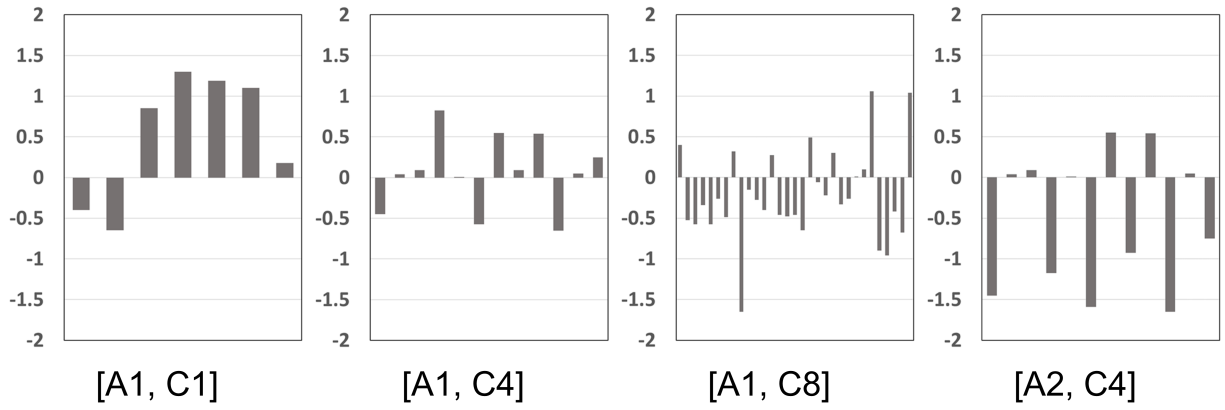


Figure 5.25: Topic-specific privacy attitude of Annotator A1 and A2 on topics C1: health&medical, C4: Obscene, C8: Family.

where T_k denotes the topic k.

We have developed a private tweet classifier similar to [94, 57, 113], which categorizes tweets into 13 predefined topics in Chapter 4. Figure 5.25 demonstrates the topic-specific privacy attitude of two human annotators for our 2018 testing dataset. We classify potentially sensitive tweets ($S_{cf} < 2.3$) into 13 topics, and show the difference between the context-free PrivScore and the human-annotated score ($S_{cf} - S_{A_i}$) for each tweet. A positive value indicates that the annotator rates the tweet as "more sensitive" than S_{cf}. We can see that Annotator A1 consistently rates "health & medical" tweets as more sensitive. Meanwhile, her attitude with obscene/cursing content is in general close to S_{cf}, while she treats "family" tweets as less sensitive. On the contrary, Annotator A2 is less concerned about obscene and cursing content. This example demonstrates individual differences in topic-specific privacy attitudes and the need for topic-specific personalization.

To model the topic-specific privacy attitude for a user, we classify all potentially sensitive tweets (with $S_{cf} < 2.3$) in her tweet history. For user u_i, the number of tweets classified into topic T_k ($k \in \mathbb{N}_{\leq 13}$) is denoted as $c_{k,i}$. The average number of sensitive posts on T_k for all the users in her societal context is denoted as:

$$\mu_{k,U_c} = \frac{1}{|\mathbf{U}_c|} \cdot \sum_{u_j \in \mathbf{U}_c} c_{k,j} \tag{5.28}$$

and the standard deviation is σ_{k,U_c}. The normalized topic-specific privacy attitude for u_i is defined

as:

$$P_{T_k,i} = \frac{-(c_{k,i} - \mu_{k,U_c})}{\sigma_{k,U_c}} \tag{5.29}$$

A negative $P_{T_k,i}$, i.e., $c_{k,i} > \mu_{k,U_c}$, indicates that u_i cares less about her privacy on topic T_k (posting more sensitive tweets on this topic than average users); while a positive $P_{T_k,i}$ indicates that u_i has better protected her private information on the topic.

Intuitively, when a user cares *less* about her privacy on topic T_k (i.e., $P_{T_k,i} < 0$), we should increase S_{cf} for her to indicate "less private" on this topic. Meanwhile, the strength of the adjustment should increase when a tweet is more relevant to the topic, and it should be configurable by the user. Hence, the personalized topic-specific PrivScore for user u_i and tweet T is defined as:

$$S_{p,i} = S_{cf} - \omega_p \cdot r_p \cdot P_{T_k,i} \tag{5.30}$$

while the *personalized context-aware PrivScore* is:

$$S_{pc,i} = S_c - \omega_p \cdot r_p \cdot P_{T_k,i} \tag{5.31}$$

where S_{cf} is the context-free PrivScore defined in (2), and S_c is the context-aware PrivScore defined in (5.25). ω_p is the weight configured by the user (we used 0.5 in the experiments). r_p is the relevance between T and the topic T_k, which is the confidence of the classification.

5.8.2 Evaluation

Evaluation with Annotated Data. Using our 2018 dataset, we further evaluate the personalized privacy scoring algorithm. As described in Section 5.6.2, 566 tweets were annotated by 8 human annotators (2 annotations/tweet). We perform 5-fold cross-validation for each annotator. In each round, the objective is to learn an annotator's topic-specific privacy attitude from 80% of the annotated tweets (training samples), and to generate personalized PrivScores for the remaining 20% of the tweets. In particular, we assume that all tweets labeled as "3 Nonsensitive" would be posted

by the annotator, and thus could be utilized to learn $P_{T_k,i}$. Meanwhile, tweets labeled as "2" or "1" would *not* be posted by the annotator, so that they would not appear in the annotator's tweet history – they cannot be used as negative training samples. Hence, we mimic the annotator's "tweet history" as all training samples annotated as "3", and ignore other training samples. We follow Eq. (11) to compute $P_{T_k,i}$ using "all annotators" as the personal context. We then calculate $S_{p,i}$ as defined in Eq. (12). We do not consider the societal context since the annotators were not exposed to the context during annotation (e.g., did not see excessive tweets on NCAA tournament). We impose weak personalization ($\omega_p = 0.3$) since we only have limited "tweet history" to learn from.

Using the same method in Section 5.6.2, we examine the distribution of the human annotations vs. the newly generated personalized PrivScores. Figure 5.22 (b) to (d) demonstrate the density of "1"s, "2"s and "3"s annotated by the human evaluators for each $S_{p,i}$ range. This figure clearly shows that the density of "sensitive" annotations is more skewed towards smaller $S_{p,i}$. For instance, all the tweets with $S_{p,i} \in [1, 1.1)$ are labeled as "sensitive" by human evaluators. In the same way, the density of "nonsensitive" annotations is more skewed towards larger $S_{p,i}$. That is, the personalized PrivScores $S_{p,i}$ are more consistent with human annotations.

We also quantitatively measure the differences between the PrivScores and human annotations. The Mean Square Error (MSE) between S_{cf} and annotated scores is 0.55. With personalized topic-specific PrivScore, the MSE between $S_{p,i}$ and annotated scores is 0.46. Note that the PrivScores are real numbers in rage $[1,3]$, while the annotated scores only take integer values $\{1,2,3\}$. This difference will unavoidably impact MSE.

Evaluation with Twitter Users. We compute the personalized PrivScores for Twitter users. Examples of $S_{p,i}$ are shown in Table 5.9, while the corresponding personal topic attitudes are shown in Fig. 5.24. Most of user 5447*****'s tweets are clean and nonsensitive. She sometimes tweets about religion, family, and travel (moderately private). Her first tweet in the example has an S_{cf} of 1.4730. However, this tweet should be adjusted to "more sensitive" due to her clean tweet history (these words are very unusual to her). Meanwhile, the second tweet, which is generally non-sensitive (or "maybe"), appears to be very normal to her ($S_{p,i} \simeq 3$). User 22149**** often

Table 5.9: Examples of topic-specific personalized privacy scores for users 5447***** (top) and 22149**** (bottom).

Topic	$P_{T_k,i}$	S_{cf}	$S_{p,i}$	Examples
Obscene	1.0553	1.4730	1.1564	Caught her looking at my boobs. #nevermind #roommateproblems
Family	-0.8043	2.6227	2.9243	All I want to do is spend quality time with my family. #changingpriorities
Obscene	-1.4720	1.6116	1.85448	I'm gonna sip wine and talk s— on Villanova
School	0.3763	2.1680	2.0551	S/o to @XXX and @XXX for doing my homework while I serve them beer

uses dirty words in tweets. Therefore, the sensitiveness of his first tweet is reduced, as he does not care about obscene/cursing words. However, it is still in "maybe" range, which is consistent with public opinions – most people feel uncomfortable with this content.

5.9 Further improvement of PrivScore using BERT

As described in 5.5.4, BERT is pre-trained by unsupervised learning method on numerous amount of text data. Tasks with limited training data can reap huge benefits from the transfer learning based on BERT. That's why we further experiment on BERT model to improve our PrivScore. The usage of BERT for a specific task is relatively straightforward. For our PrivScore, a classification task, we only need to add a small classification layer on top of the Transformer output for the [CLS] token. This transfer learning structure is shown in Figure 5.26. We utilize the same dataset in the previous LSTM model to fine-tune this BERT model and 5-fold-cross validation is used to evaluate the model efficiency. The classification performance of Bert and its corresponding confusion matrix are shown in Table 5.10 and Table 5.11 respectively. From the results, shown in Table 5.12, we can see that BERT performs notably better than the LSTM model using only text. Though BERT is just marginally better than the LSTM model with text and sentiment feature, the model implementation becomes easier. We do not need to compute sentiment through a third-party API or another complicated, sentiment specific neural network.

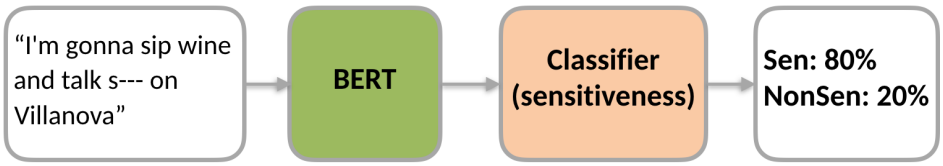


Figure 5.26: PrivScore using BERT

In addition, BERT shows more potential of further improvement than that of LSTM model. This is because our relatively small dataset limits the complexity of the model we can use, if we train the model from scratch. With pre-trained BERT, the basic understanding of words and context are already learned, and the task specific data is further used for the fine-tuning, which significantly boost the efficiency of learning. Therefore, BERT is a promising model for our task and worth more effort to explore.

Table 5.10: Classification performance of BERT

	Precision	Recall	F1-Score	Support
Sensitive	0.8715	0.8464	0.8585	1435
Non-sensitive	0.8506	0.8753	0.8625	1435
Average	0.8611	0.8609	0.8604	2870

Table 5.11: Confusion matrix of BERT

	Sensitive	Non-sen
Sensitive	1214	221
Non-sensitive	179	1256

Table 5.12: Comparison of the performances of different models

	Precision	Recall	F1-Score	Support
LSTM	0.83	0.83	0.83	2870
LSTM with senti	0.85	0.85	0.85	2870
BERT	**0.86**	**0.86**	**0.86**	2870

5.10 Security Analysis & Discussions

5.10.1 Security, Performance, and Usability

The PrivScore model will be employed in social networks for user alerting or self-censorship of AI chatbots. When an alerting mechanism is properly deployed and the user follows the warnings, sensitive content will not be disseminated to followers or malicious stalkers. However, the protection performance will be primarily impacted by two factors: the accuracy of the privacy scores, and the design/usability of the alerting mechanism. First, the privacy scoring approach may generate two types of errors: false positives and false negatives.

False negative. When the PrivScore is (significantly) higher than what the users would perceive, a sensitive tweet will be labeled as nonsensitive, i.e., a false negative. In a user alerting system, false negatives cause missed alerts, so that messages containing sensitive information may be posted. While it is impossible to completely eliminate false negatives from any statistical learning approach, the problem may be mitigated: (1) The performance of privacy scoring will increase with more training data and advances in NLP (to be elaborated later). (2) We also observed that sensitive tweets often lead to sensitive responses (e.g., cursing tweets get cursing replies), hence, hints of missed alerts may be learned by monitoring responses. (3) An auditing mechanism could be developed to periodically re-evaluate past tweets with the updated scoring model, to alert users to fix any possible damage [91].

False positives. When the PrivScore is (significantly) lower than users' perceptions, a nonsensitive tweet will be labeled as sensitive, i.e., a false positive. When the false alarms are sporadic and the alerting mechanism is not intrusive, they may not cause burdens to the users. However, frequent false alarms affect the usability of the alerting mechanism, which may prevent users from adopting it. In practice, false positives may be mitigated: (1) A well-designed configuration interface will allow user to specify her own topic-specific preferences so that alerts could be adjusted accordingly. (2) Personalized privacy scoring model observes personal privacy attitudes/behaviors, and tunes privacy scoring. Online learning could be employed to continuously improve scoring accuracy

when more personal data becomes available. (3) Better alert and response interfaces could be designed to minimize the disruption to users.

False positives need to be handled properly, so that we do not overly disturb the users that they eventually disable/ignore the protection mechanism. All these discussions are beyond the scope of this project, but are interesting future topics.

The Accuracy of Keyword Spotting. In Section 5.2, we employ a keyword spotting approach to identify a candidate tweet set to be labeled by Turkers. While similar approaches have been employed in the literature to identify if a tweet belongs to a pre-defined topic. We aim to increase recall in this process, i.e., to include a majority of potentially private tweets. However, we acknowledge that there exist both false positives and false negatives in this process. A false positive is a tweet that contains at least one keyword but is indeed not sensitive. A significant portion of the candidate tweets belongs to this category and they pose major challenges to our classifier. We handle them through the labeling, representation and classification processes. On the other hand, a false negative is a tweet that does not include any keyword but contains sensitive content. We do not anticipate such false negatives to cause any noticeable impact in scoring performance due to the following:

(1) False negatives are very rare. We have used a relatively large set of keywords for each category: more than 100 for each category (as a reference, the privacy dictionary [108], which was used in Privacy Detective [47], contains 355 terms in eight categories). To estimate the false negative rate, we randomly selected 500 tweets from the non-candidate set, i.e., tweets do not contain any keyword, and posted them on MTurk, where each tweet was annotated by two Turkers. We also added approximately 50% of sensitive tweets in the questionnaires to keep Turkers' attention. As a result, only one tweet was labeled as "1 [sensitive]" by both Turkers: "*Just got tazed trying to get into the CBC basketball game....Half hour before the game starts.*", while one tweet received "1, 2", 11 tweets received "1, 3", and all other tweets received "2, 3" or higher. For these "maybe sensitive" tweets, most of them only imply a very subtle sensitiveness that was hidden behind the words.

(2) Missing terms are captured by word embedding. Unlike the conventional bag-of-words model that treats any two different words as orthogonal in the vector space, word embedding models capture words' meanings from their context, and discover the semantic and syntactic similarities between terms. Therefore, as long as a term is included in the GloVe dataset (pre-trained with 2B tweets and 27B tokens) and appeared in similar semantic contexts with known sensitive words, it will be represented close to sensitive words in the model. Meanwhile, LSTM also attempts to capture the semantic meanings behind word sequences, so that the privacy scoring mechanism does not solely rely on the occurrences of sensitive terms, and could overcome a small number of missed sensitive terms. For instance, tweet "*wipe that ugly smile off your face*" does not contain any keyword in our list, however, its PrivScore of 1.62 (moderately sensitive) indicates that our mechanism captured the rude and judgmental tone from the textual content.

Deleted Tweets. Research has shown that users may delete regretted posts to repair the potential damage [91, 118]. However, study also showed that no substantial differences were observed in the "distributions of stereotypically regrettable topics" among deleted and undeleted tweets [4]. [6] found that "possible sensitive text" is a weak feature in predicting tweet deletion. Manual examination in [126] revealed that a regrettable reason was identified for only 18% of the deleted tweets, while the others cannot be explained by the tweet content. Therefore, we did not use deleted tweets in our privacy scoring models or experiments. However, we suggest that deleted tweets could be employed in personalized privacy scoring, as a factor of the topic-specific privacy attitude. In particular, Eq. (5.29) will be modified to infer privacy attitude from two factors: tweet history and deleted tweets, where explicitly deleted tweets on a topic may imply that the user is more conservative on this topic. Further investigation of deleted tweets and employing them in privacy scoring is in our future plans.

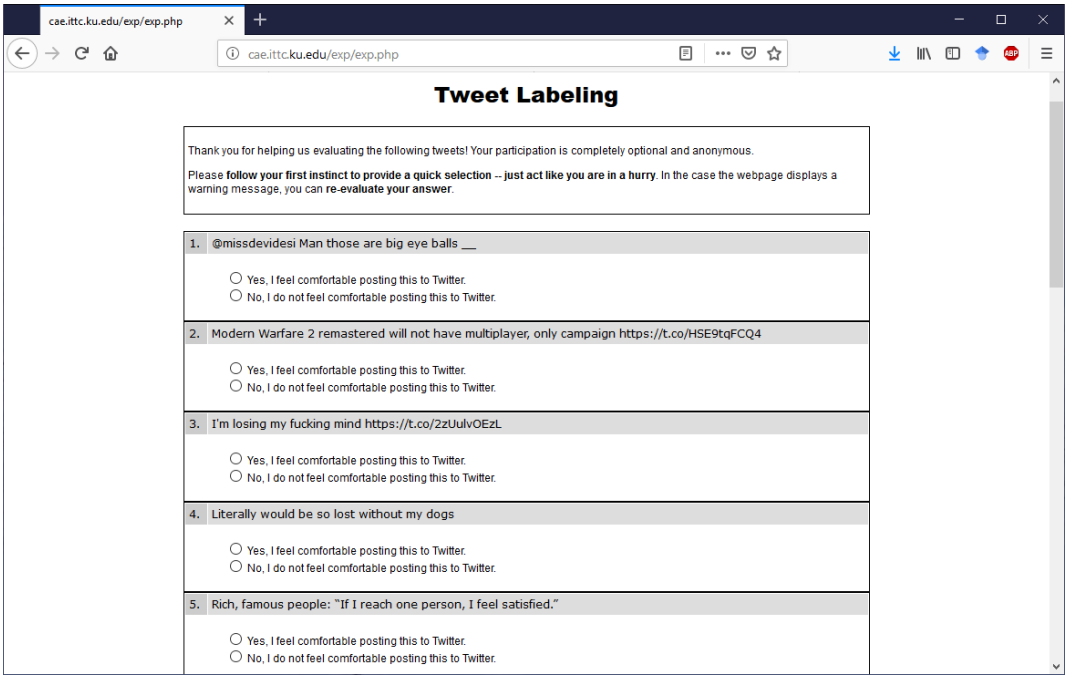
User Alerting and Usability. Research on private tweet modeling attempts to discover the psychological factors and cognitive models behind private tweets (see Chapter 2 for details). They suggested that tools could be developed to "*detect potentially regrettable messages*" [91] and "*a content-based reminder could be triggered*" [118] to alert the users. To achieve this goal, we first

need a mechanism that automatically assesses message content to identify sensitive tweets to trigger the alerts. Therefore, PrivScore serves as a fundamental building block for a comprehensive privacy protection solution. The solution could be implemented as a browser add-on or a mobile app. It first takes users' baseline preferences through an interactive configuration interface. When the user starts to type a message, its PrivScore is evaluated on-the-fly. If the user attempts to post sensitive content (determined by pre-set topic-specific thresholds), a warning message will be displayed to trigger self-censorship. To demonstrate the effectiveness of employing PrivScore in triggering self-censorship, we perform a proof-of-concept evaluation for user alerting (an IRB approval was obtained for this evaluation). The user study is designed as follows:

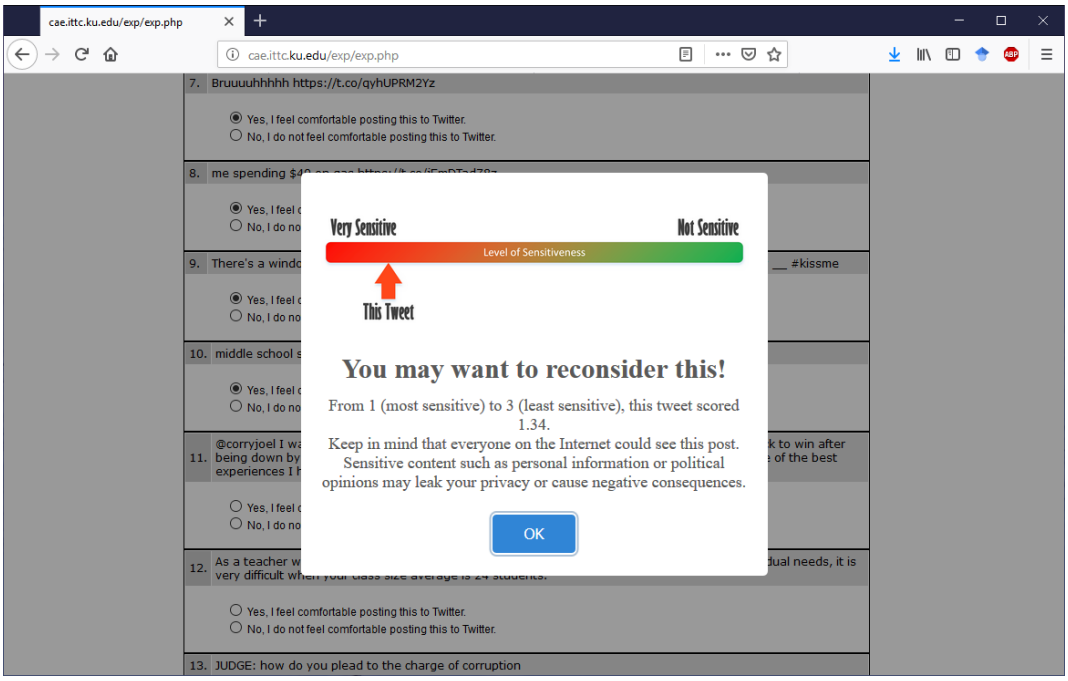
- **a.** We recruited college students to evaluate tweets crawled from the Internet. Each participant received a URL to an anonymous online questionnaire, which includes 15 tweets (sampled with higher density of sensitive tweets than the original distribution).
- **b.** Participants were asked to select "1. Yes I feel comfortable posting this to Twitter" or "2. No, I do NOT feel comfortable posting this" for each tweet, as shown in Figure 5.27 (a). To mimic an emotional or urgent scenario, we asked students to "*follow your first instinct to provide a quick selection–just act like you are in a hurry.*"
- **c.** If the student chose "Yes" for a low score tweet (i.e., a tweet with $S_{cf} < 1.5$), a warning message was displayed (as shown in Figure 5.27 (b)). The student has the option to adhere to the advice or stick to her original option.

We record the participant's selection for each tweet. We also record whether the warning message was triggered, and whether the participant adhered to the message. Out of 795 tweets (53 questionnaires) that was answered in 10 days, 94 tweets triggered the warning message, while users changed opinions on 58 tweets: an adherence rate of 61.7%. Manual inspection also showed that users stick to their original selection of "Yes" mostly for political and judgemental tweets, which indicates that our annotators were more conservative on political content than the evaluators. There were also a few false positives. One example was a tweet criticizing racism, which

received a PrivScore of 1.21 (strong critical tones and racist terms). However, since it was criticizing racism, it should not receive such a low score.



(a)



(b)

Figure 5.27: User study on the effectiveness of user alerting.

The usability and user experience aspects of an alerting system is a challenging issue, which requires intensive further investigation. As references, browsers' alerts (phishing attacks, HTTPS certificate errors) and users' responses have been intensively investigated in the literature [120, 24, 96, 93, 85, 119, 2]. For instance, a recent study [85] examines users' responses to security alerts in Chrome and Firefox, analyzes the decision factors, and makes suggestions to designers. In our application, intuitively, a good alerting mechanism is expected to be less disturbing and provide the user with sufficient but concise information of the alert rationale. Meanwhile, different levels of warning may be enforced for different levels of sensitiveness, e.g, alerting the user of sensitive content and the potential audience [91]. Moreover, [117, 118, 116] suggested a delayed posting mechanism using a timer nudge for Facebook, to "*encourage users to pause and think*" before posting a message. Last, configuration of parameters and personalization of alerting are also important topics that need to be studied.

5.10.2 Comparison with Instagram's Comment Filtering Mechanism

On July 8th, 2019, Instagram announced a new feature called "comment warning" to battle against online bullying. As described by Instagram, this new feature, powered by AI, warns people when their comments may be considered offensive before they are posted [46]. The production demo of this new feature is shown in Figure 5.28 [46]. In March 2020, we performed an experiment on Instagram to evaluate this feature. It is clear that the new function is still in the experimental stage, so that it is not always available, and it appears to be very inconsistent. We also have some interesting observations on this new feature through experiments:

- **a.** This feature is still rolling out and in the experiment stage. We experimented on 7 individual Instagram accounts in US with cursing/bullying posts. However, none of them triggered the advertised "comment warning" warning function, when multiple offensive comments were posted, including the exact one in the advertisement and news reports of the new function. On the other hand, the *comment filtering* feature did work, which prevents some bullying comments from being displayed to the owner of the posts. Therefore, in the rest of our experiments, we focus on

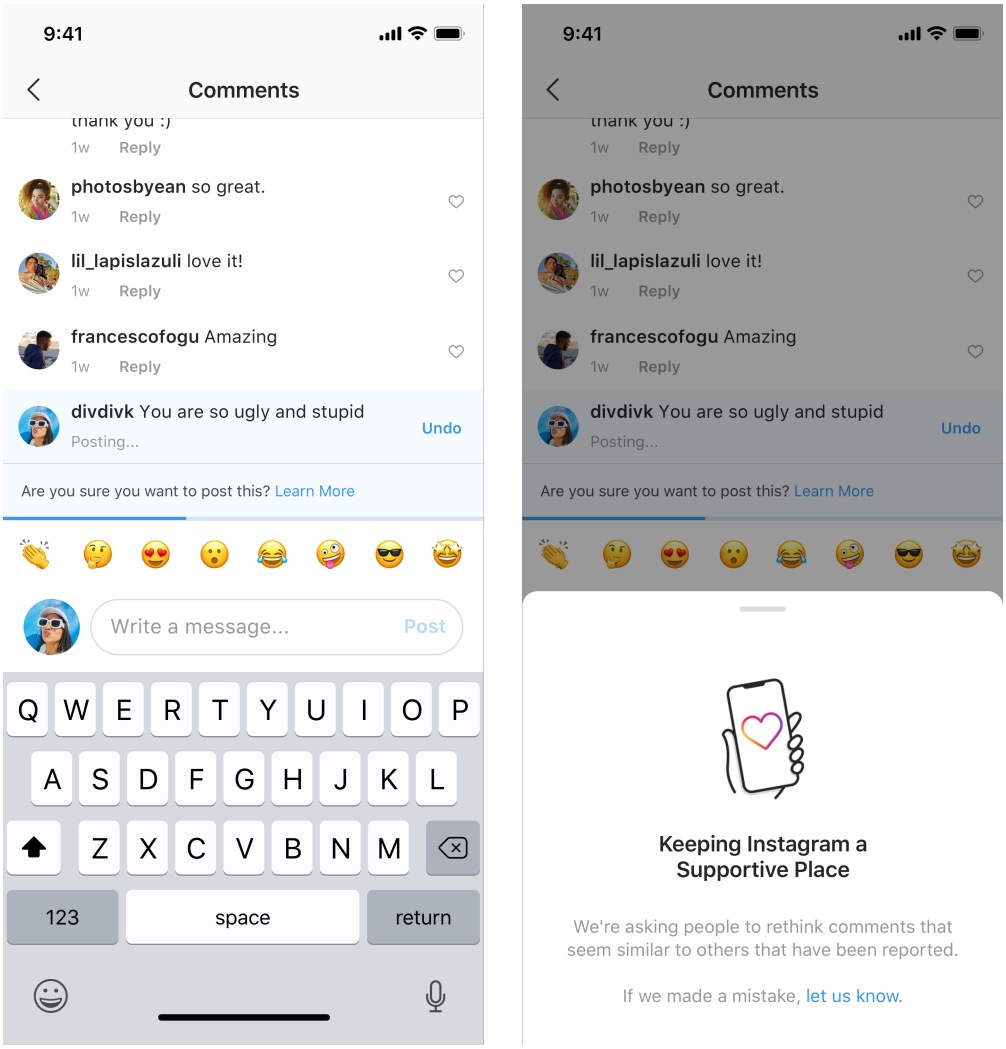


Figure 5.28: Instagram Comment Warning.

the comment filtering function.

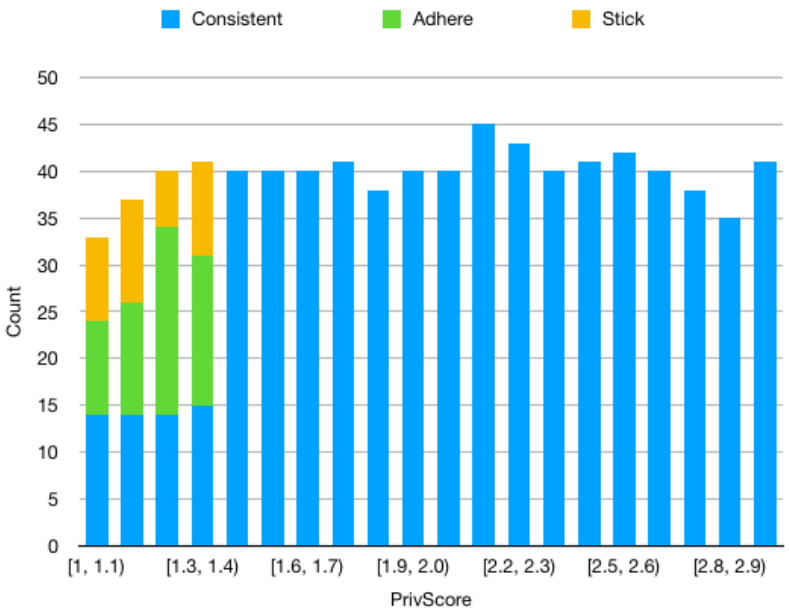
The availability of the filtering function varies from account to account, regardless of the account configurations. We observed different behaviors from different users' accounts. 2 out of 7 accounts in our experiment benefit from the anti-bullying function, i.e., they cannot view some of the offensive/cursing comments left by others. However, the rest of the accounts could not filter out offensive comments automatically, i.e., all the comments (left by other users) are visible to the owner of the post.

- **b.** The intention of this new feature is to protect Instagram users from cyber bullying. This

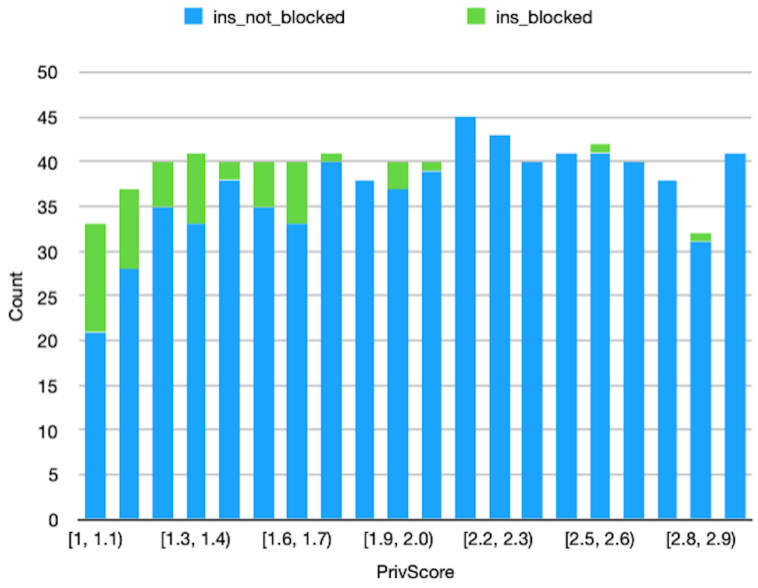
is similar to our mechanism to a certain extend, which is to reduce the chance of impulsive posting or posting sensitive information. However, we take different angles in this scenario. For Instagram, it aims to help the owner of the posts to filter out bullying comments written by his/her followers. It only focus on offensive/cursing/attacking comments, but not other types of sensitive comment, e.g., personal information. Meanwhile, the objective of our PriScore mechanism is to protect the user from leaking private/sensitive information before posting. Thus, our mechanism covers a broader range of private/sensitive information, and we provide fine-grained scores that provide better measurement than a binary decision of "bullying"/"not bullying".

- **c.** According to the official description by Instagram, comments that may be inappropriate, offensive or bullying are automatically filtered out [45]. Users can also extend the default keyword filter by adding customized keywords to hide comments that contain specific words, phrases, etc. This description suggests that keyword-spotting is used int the filtering mechanism. Through our experiments, we also noticed that certain level of natural language processing (NLP) technique was utilized. For example, the comment – "girls are like garbage plates, best from rochester" that does not contain any extremely dirty word, is blocked. But comments with subsets of the blocked comment, such as "arbage plates", are not blocked. Hence, we can conclude that the comment filtering feature takes the semantic meaning of sentences into consideration.

Next, we perform an experiment to systematically compare Istagram's comment blocking function with PrivScore. We posted an image to Instagram, and used another account to post the tweets we used in the user study in Section 5.10.1 as comments to this image. We record whether each comment was blocked by Instagram, and compare with: (1) the context-free privacy score; and (2) users' annotations collected in the user study. Our experiment results are shown in Figure 5.29. Through the experiment results we can see that the comment filtering feature of Instagram has large number of false negatives, i.e., large number of sensitive comments are not blocked. As mentioned in the user study section, we set the default threshold as 1.4, which means that tweets scoring below 1.4 are all treated as sensitive by our mechanism. However, in Figure 5.29 (b), only a small portion of tweets are block by Instagram. For example, tweet "i embarrassed honestly spent half



(a)



(b)

Figure 5.29: Comparison with Instagram Comment Filtering. (a). Results from the user study: the distribution of tweets: (1) the user agrees with the PrivScore; (2) the user changed labels after seeing a warning message, and (3) user rejects the warning. (b). The distribution of tweets blocked and not blocked by Instagram.

hour sitting parking lot crying goose happen smh.. i get big a*s heart" gets a PrivScore at 1.173. Our mechanism treated it as sensitive content, due to its strong emotion disclosure, and use of inappropriate language. But Instagram did not block this tweet. Another example is that tweet “no joke

day election i went work coworker wearing red hat i ready throw f**king hands turned kentucky derby hat" with PrivScore 1.26 and an explicit cursing word was not blocked by Instagram either. This election tweet might not be that sensitive to college students. It triggered a warning during the user study. The student adhered to our suggestion and decided not to post the message after reading the warning. Through further investigation on the 58 tweets that students adhered to the warning message, we found that Instagram only blocked 34 out of 58. Through this experiment, we confirmed that the blocking feature of Instagram is mostly sensitive to cursing words, like "s**t" and "b***h".

5.10.3 Limitations and Future Improvements

We make a first attempt to assess the level of sensitiveness of text content. Our mechanism still has its limitations, for instance: (1) PrivScore is designed as a preventative privacy protection solution. When a sensitive message is posted, PrivScore does not provide a mechanism to withdraw the tweet or prevent potential damages. (2) As a statistical learning approach, false positives/negatives are practically unavoidable, especially due to the subjective nature of privacy perception.

The accuracy of privacy scoring could be further improved from three aspects: (1) More annotated data and higher quality labels (e.g., professional annotators) could improve the performance of classification and privacy scoring, however, it requires significant costs. (2) Advances in NLP, such as BERT and the latest derivative of BERT – ALBERT benefit their downstream tasks, including PrivScore. Results in section 5.9 show the effectiveness of BERT. More improvement could be achieved through further domain-data pre-training and better fine-tuning strategies [95]. (3) Once privacy scoring and alerting mechanisms are deployed to users, we can adopt online learning to train the PrivScore model. When users reject warning messages of false alarms, new annotated data is incrementally added to the model to improve privacy scoring performance.

Besides the plan to further improve the accuracy of PrivScore and to address the challenges in enhancing user experiences in private content alerting, we also identify several future research directions: the effective integration of privacy scoring and classification will be beneficial, espe-

cially for personalized privacy protection. Privacy scoring with consideration of the audience, and the integration of privacy scoring with access control, are both challenging research questions. We expect to apply the privacy scoring mechanism for other types of text data. In theory, the proposed framework could work for any type of text, as long as there exist labeled training samples that are homogeneous to testing samples. However, there are practical challenges that need to be addressed, such as text modeling, segmentation of long paragraphs, etc.

5.11 Summary

In this Chapter, we make the first attempt to develop a computational model using deep neural networks to quantitatively assess the level of privacy/sensitiveness for textual content in OSNs. Our framework consists of four key components: (1) collection and analysis of privacy opinions on potentially private information; (2) the context-free privacy scoring model, which mimics users' privacy perceptions to assess the degree of privacy mainly based on text content; (3) the context-aware privacy scoring model, which considers the influences of the societal context on privacy perceptions; and (4) the personalized privacy scoring model, which integrates topic-specific personalized privacy attitude into the privacy scores. With experiments on human annotated data, we show that the PrivScore model is consistent with human perception of privacy. This work was published in [115].

Chapter 6

Conclusion

The increasing popularity of online social networks brings a large amount of private or sensitive information posted. Studies show that users tend to unintentionally release regretful messages or reveal too much potentially sensitive information online, especially when they are careless, emotional, or unaware of privacy risks. Therefore, in this dissertation, we propose a mechanism for protecting users' content privacy in three phrases: context-free privacy scoring, context-aware privacy scoring, and personalized privacy scoring. To develop this mechanism, we have done the following four parts:

First, we explicitly research and study topics that might cause user's regret or privacy leakage. Based on this thoroughly research, tweets potentially related to these topics are extracted. We then make the first attempt to classify these potentially sensitive tweets into a comprehensive set of likely sensitive categories. To boost the classification accuracy, the classifer is built with both semantic features and users' topic-preferences.

Second, to examine if there is a consensuses towards the content sensitivity among online social network users, we launch a crowd-sourcing survey on Amazon Mechanical Turk to collect the privacy perceptions from a diverse set of users. The survey shows the feasibility of obtaining a common perception towards content sensitiveness/privacy for average users in a neutral context. This provides the foundation of developing our privacy protection mechanism.

Third, we make the first attempt to develop a computational model for quantitative assessment of content sensitiveness using deep learning networks. This model resembles the "consensus" perception of average users on the purely textual content, which is our context-free privacy score model.

Last but not the least, we further adjust the context-free privacy score with social context and personal preferences to generate the context-aware score and personalized privacy score respectively. Therefore, we implement a mechanism for protecting users' private content, and to an extent, the mechanism would not affect user's normal socializing.

In conclusion, we study what is privacy for online social network users and we argue that *OSN privacy as having the ability to control the dissemination of sensitive information*. Based on this understanding, we propose the first quantitative model for private information assessment, which generates a context-free privacy score. In addition, we adjust the context-free privacy score with the societal context to obtain a context-aware privacy score, and extend the model using personalization by introducing user's topic-preference and tweet history. This mechanism will both benefit the OSN users and other applications such as chatbots.

References

[1] Abawajy, J. H., Ninggal, M. I. H., Aghbari, Z. A., Darem, A. B., & Alhashmi, A. (2017). Privacy threat analysis of mobile social network data publishing. In *SecureComm*.

[2] Acer, M. E., Stark, E., Felt, A. P., Fahl, S., Bhargava, R., Dev, B., Braithwaite, M., Sleevi, R., & Tabriz, P. (2017). Where the wild warnings are: Root causes of chrome https certificate errors. In *ACM CCS* (pp. 1407–1420).: ACM.

[3] Aggarwal, C. C. & Zhai, C. (2012). *Mining text data*. Springer Science & Business Media.

[4] Almuhimedi, H., Wilson, S., Liu, B., Sadeh, N., & Acquisti, A. (2013). Tweets are forever: a large-scale quantitative analysis of deleted tweets. In *ACM CSCW* (pp. 897–908).

[5] Baden, R., Bender, A., Spring, N., Bhattacharjee, B., & Starin, D. (2009). Persona: an online social network with user-defined privacy. *SIGCOMM*.

[6] Bagdouri, M. & Oard, D. W. (2015). On predicting deletions of microblog posts. In *ACM CIKM*.

[7] Barnes, S. B. (2006). A privacy paradox: Social networking in the united states. *First Monday*, 11(9).

[8] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157–166.

[9] Blank, G., Bolsover, G., & Dubois, E. (2014). A new privacy paradox: Young people and privacy on social network sites. In *Annual Meeting of the American Sociological Assoc.*

[10] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993–1022.

[11] Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013). Twitie: An open-source information extraction pipeline for microblog text. In *RANLP* (pp. 83–90).

[12] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4), 467–479.

[13] Cai, Z., He, Z., Guan, X., & Li, Y. (2018). Collective data-sanitization for preventing sensitive information inference attacks in social networks. *IEEE TDSC*, 15(4).

[14] Chang, H.-W., Lee, D., Eltaher, M., & Lee, J. (2012). @ Phillies tweeting from Philly? Predicting Twitter user locations with spatial word usage. In *IEEE ASONAM*.

[15] Cheng, Z., Caverlee, J., & Lee, K. (2010). You are where you tweet: a content-based approach to geo-locating twitter users. In *ACM CIKM*.

[16] Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

[17] Ciot, M., Sonderegger, M., & Ruths, D. (2013). Gender inference of twitter users in Non-English contexts. In *EMNLP* (pp. 1136–1145).

[18] Dawes, J. (2008). Do data characteristics change according to the number of scale points used? an experiment using 5-point, 7-point and 10-point scales. *IJMR*, 50(1), 61–104.

[19] De Cristofaro, E., Soriente, C., Tsudik, G., & Williams, A. (2012). Hummingbird: Privacy at the time of twitter. In *Security and Privacy (SP), 2012 IEEE Symposium on* (pp. 285–299).: IEEE.

[20] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[21] Dhir, A., Torsheim, T., Pallesen, S., & Andreassen, C. S. (2017). Do online privacy concerns predict selfie behavior among adolescents, young adults and adults? *Front Psy.*, 8.

[22] Dinev, T. & Hart, P. (2005). Internet privacy concerns and social awareness as determinants of intention to transact. *International Journal of Electronic Commerce*, 10(2), 7–29.

[23] Dwork, C. (2008). Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation* (pp. 1–19).: Springer.

[24] Egelman, S., Cranor, L. F., & Hong, J. (2008). You've been warned: an empirical study of the effectiveness of web browser phishing warnings. In *ACM CHI*.

[25] Fang, L. & LeFevre, K. (2010). Privacy wizards for social networking sites. In *Proceedings of the 19th international conference on World wide web* (pp. 351–360).: ACM.

[26] Feldman, R. & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press.

[27] Feldman, R.-S. (2006). J, the text mining handbook.

[28] Fleiss, J. L. & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educ Psychol Meas.*, 33(3).

[29] Fogel, J. & Nehmad, E. (2009). Internet social network communities: Risk taking, trust, and privacy concerns. *Computers in human behavior*, 25(1), 153–160.

[30] Ford, D. (2014). Justin Bieber apologizes for racist joke. CNN.

[31] Garfinkel, S. (2000). *Database nation: The death of privacy in the 21st century*. " O'Reilly Media, Inc.".

[GATE] GATE. GATE Overview, `https://gate.ac.uk/overview.html`.

[33] Gerber, N., Gerber, P., & Volkamer, M. (2018). Explaining the privacy paradox-a systematic review of literature investigating privacy attitude and behavior. *Computers & Security*.

[34] Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1), 1–309.

[35] Google (2013). Google pre-trained word2vec.

[36] Guthrie, L., Walker, E., & Guthrie, J. (1994). Document classification by machine: theory and practice. In *Conference on Computational linguistics*.

[37] Haeberlen, A., Pierce, B. C., & Narayan, A. (2011). Differential privacy under fire. In *USENIX Security Symposium*.

[38] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1), 10–18.

[39] Hargittai, E. & Marwick, A. (2016). "what can i really do?" explaining the privacy paradox with online apathy. *International Journal of Communication*, 10, 21.

[40] He, J., Chu, W. W., & Liu, Z. V. (2006). Inferring privacy information from social networks. In *ISI*.

[41] Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.

[42] Humphreys, L., Gill, P., & Krishnamurthy, B. (2010). How much is too much? privacy issues on twitter. In *Conference of International Communication Association, Singapore*.

[43] Iachello, G., Hong, J., et al. (2007). End-user privacy in human–computer interaction. *Foundations and Trends in Human–Computer Interaction*, 1(1).

[44] Ikawa, Y., Enoki, M., & Tatsubori, M. (2012). Location inference using microblog messages. In *21st International Conference on World Wide Web* (pp. 687–690).

[45] Instagram. How do i filter out comments i don't want to appear on my posts on instagram?

[46] Instagram. Our commitment to lead the fight against online bullying.

[47] Islam, A., Walsh, J., & Greenstadt, R. (2014). Privacy detective: Detecting private information and collective privacy behavior in a large social network. In *ACM WPES*.

[48] Jahid, S., Mittal, P., & Borisov, N. (2011). Easier: Encryption-based access control in social networks with efficient revocation. In *ACM AsiaCCS*.

[49] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013a). *An introduction to statistical learning*, volume 112. Springer.

[50] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013b). *An introduction to statistical learning*, volume 112. Springer.

[51] Johnson, M., Egelman, S., & Bellovin, S. M. (2012). Facebook and privacy: it's complicated. In *Eighth symposium on usable privacy and security* (pp.$\tilde{9}$).: ACM.

[52] K, Z. G. (1935). *The psychology of language*. Houghton-Mifflin.

[53] Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[54] Lampe, C., Ellison, N. B., & Steinfield, C. (2008). Changes in use and perception of facebook. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work* (pp. 721–730).: ACM.

[55] Lampos, V., Aletras, N., Geyti, J. K., Zou, B., & Cox, I. J. (2016). Inferring the socioeconomic status of social media users based on behaviour and language. In *ECIR*.

[56] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

[57] Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A., & Choudhary, A. (2011). Twitter trending topic classification. In *IEEE ICDM Workshops*.

[58] Lenhart, A. (2005). Protecting teens online. In *Pew Internet & American Life Project*.

[59] Lewis, K., Kaufman, J., & Christakis, N. (2008). The taste for privacy: An analysis of college student privacy settings in an online social network. *J Comp Mediat Comm.*, 14(1).

[60] Li, N., Li, T., & Venkatasubramanian, S. (2007). t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*.

[61] Li, R.-H., Liu, J., Yu, J. X., Chen, H., & Kitagawa, H. (2013). Co-occurrence prediction in a large location-based social network. *Frontiers of Computer Science*, 7(2), 185–194.

[62] Lipford, H. R., Besmer, A., & Watson, J. (2008). Understanding privacy settings in facebook with an audience view. *UPSEC*, 8, 1–8.

[63] Litt, E. (2013). Understanding social network site users' privacy tool use. *Computers in Human Behavior*, 29(4), 1649–1656.

[64] Liu, H., Luo, B., & Lee, D. (2012). Location type classification using tweet content. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1 (pp. 232–237).: IEEE.

[65] Liu, K. & Terzi, E. (2010). A framework for computing the privacy scores of users in online social networks. *ACM Transactions on Knowledge Discovery from Data*, 5(1).

[66] Liu, W. & Ruths, D. (2013). What's in a name? using first names as features for gender inference in twitter. In *AAAI spring symposium: Analyzing microtext*, volume 13 (pp.01).

[67] Luo, B. & Lee, D. (2009a). On protecting private information in social networks: A proposal. In *IEEE ICDE Workshop on Modeling, Managing, and Mining of Evolving Social Networks (M3SN)*.

[68] Luo, B. & Lee, D. (2009b). On protecting private information in social networks: a proposal. In *IEEE ICME Workshop of M3SN*: IEEE.

[69] Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), 3.

[70] Mahmud, J., Nichols, J., & Drews, C. (2014). Home location identification of twitter users. *ACM TIST*, 5(3), 47.

[71] Mao, H., Shuai, X., & Kapadia, A. (2011). Loose tweets: an analysis of privacy leaks on twitter. In *ACM WPES*.

[72] McCallum, A. K. (2002). Mallet: A machine learning for language toolkit.

[73] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[74] Minaei, M., Mondal, M., Loiseau, P., Gummadi, K., & Kate, A. (2019). Lethe: Conceal content deletion from persistent observers. *Privacy Enhancing Technologies*.

[75] Mislove, A., Viswanath, B., Gummadi, K. P., & Druschel, P. (2010). You are who you know: inferring user profiles in online social networks. In *ACM WSDM*.

[76] Mondal, M., Messias, J., Ghosh, S., Gummadi, K. P., & Kate, A. (2016). Forgetting in social media: Understanding and controlling longitudinal exposure of socially shared data. In *SOUPS 2016* (pp. 287–299).

[77] Moore, K. & McElroy, J. C. (2012). The influence of personality on facebook usage, wall postings, and regret. *Computers in Human Behavior*, 28(1), 267–274.

[78] Narasimman, A., Wang, Q., Li, F., Lee, D., & Luo, B. (2019). Arcana: Enabling private posts on public microblog platforms. In *IFIP International Conference on ICT Systems Security and Privacy Protection* (pp. 271–285).: Springer.

[79] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4), 694–707.

[80] Patil, S., Norcie, G., Kapadia, A., & Lee, A. J. (2012). Reasons, rewards, regrets: privacy considerations in location sharing as an interactive practice. In *SOUPS*.

[81] Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *EMNLP*.

[82] Pergament, D., Aghasaryan, A., Ganascia, J.-G., & Betgé-Brezetz, S. (2011). Forps: Friends-oriented reputation privacy score. In *First International Workshop on Security and Privacy Preserving in e-Societies* (pp. 19–25).

[83] Pontes, T., Magno, G., Vasconcelos, M., Gupta, A., Almeida, J., Kumaraguru, P., & Almeida, V. (2012). Beware of what you share: Inferring home location in social networks. In *ICDM Workshops*: IEEE.

[84] Preoţiuc-Pietro, D., Volkova, S., Lampos, V., Bachrach, Y., & Aletras, N. (2015). Studying user income through language, behaviour and affect in social media. *PloS one*, 10(9).

[85] Reeder, R. W., Felt, A. P., Consolvo, S., Malkin, N., Thompson, C., & Egelman, S. (2018). An experience sampling study of user reactions to browser warnings in the field. In *ACM CHI* (pp. 512).: ACM.

[86] Rish, I. et al. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3 (pp. 41–46).

[87] Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4), 333–389.

[88] Schomakers, E.-M., Lidynia, C., Müllmann, D., & Ziefle, M. (2019). Internet users' perceptions of information sensitivity–insights from germany. *International Journal of Information Management*, 46, 142–150.

[89] Singh, I., Butkiewicz, M., Madhyastha, H. V., Krishnamurthy, S. V., & Addepalli, S. (2013). Twitsper: tweeting privately. *IEEE Security & Privacy*, 11(3), 46–50.

[90] Sleeper, M., Balebako, R., Das, S., McConahy, A. L., Wiese, J., & Cranor, L. F. (2013a). The post that wasn't: exploring self-censorship on facebook. In *ACM CSCW*.

[91] Sleeper, M., Cranshaw, J., Kelley, P. G., Ur, B., Acquisti, A., Cranor, L. F., & Sadeh, N. (2013b). i read my twitter the next morning and was astonished: a conversational perspective on twitter regrets. In *ACM CHI* (pp. 3277–3286).

[92] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

[93] Sotirakopoulos, A., Hawkey, K., & Beznosov, K. (2011). On the challenges in usable security lab studies: lessons learned from replicating a study on ssl warnings. In *SOUPS*: ACM.

[94] Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *ACM SIGIR*: ACM.

[95] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics* (pp. 194–206).: Springer.

[96] Sunshine, J., Egelman, S., Almuhimedi, H., Atri, N., & Cranor, L. F. (2009). Crying wolf: An empirical study of ssl warning effectiveness. In *USENIX Security*.

[97] Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 557–570.

[98] Takemura, H. & Tajima, K. (2012). Tweet classification based on their lifetime duration. In *ACM CIKM*.

[99] Talukder, S. & Carbunar, B. (2018). Abusniff: Automatic detection and defenses against abusive facebook friends. In *AAAI Conference on Web and Social Media*.

[100] Tufekci, Z. (2008). Can you see me now? audience and disclosure regulation in online social network sites. *Bulletin of Science, Technology & Society*, 28(1), 20–36.

[101] Twitter. Twitter Developer Documentation, `https://dev.twitter.com/rest/public`.

[102] Twitter. Api reference index.

[103] Twitter. Twitter Usage company facts. https://about.twitter.com/company. Accessed: 2016-07-20.

[Uchendu et al.] Uchendu, A., Cao, J., Wang, Q., Luo, B., & Lee, D. Characterizing man-made vs. machine-made chatbot dialogs.

[urban] urban. Urban Dictionary, http://www.urbandictionary.com.

[106] usnews (2013). Twitter ipo. [Online; accessed 3-October-2013].

[107] Varol, O., Ferrara, E., Davis, C. A., Menczer, F., & Flammini, A. (2017). Online human-bot interactions: Detection, estimation, and characterization. In *ICWSM*.

[108] Vasalou, A., Gill, A. J., Mazanderani, F., Papoutsi, C., & Joinson, A. (2011). Privacy dictionary: A new resource for the automated content analysis of privacy. *J Am Soc Inf Sci Technol.*, 62(11), 2095–2105.

[109] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

[110] Vitak, J., Blasiola, S., Patil, S., & Litt, E. (2015). Balancing audience and privacy tensions on social network sites: Strategies of highly engaged users. *International Journal of Communication*, 9, 20.

[111] Volkova, S. & Bachrach, Y. (2015). On predicting sociodemographic traits and emotions from communications in social networks and their implications to online self-disclosure. *Cyberpsychol Behav Soc Netw.*, 18(12).

[112] Vosoughi, S. & Roy, D. (2015). A human-machine collaborative system for identifying rumors on twitter. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on* (pp. 47–50).: IEEE.

[113] Wang, Q., Bhandal, J., Huang, S., & Luo, B. (2017a). Classification of private tweets using tweet content. In *IEEE ICSC*.

[114] Wang, Q., Bhandal, J., Huang, S., & Luo, B. (2017b). Content-based classification of sensitive tweets. *International Journal of Semantic Computing*, 11(04), 541–562.

[115] Wang, Q., Xue, H., Li, F., Lee, D., & Luo, B. (2019). # donttweetthis: Scoring private information in social networks. *Proceedings on Privacy Enhancing Technologies*, 2019(4), 72–92.

[116] Wang, Y., Leon, P. G., Acquisti, A., Cranor, L. F., Forget, A., & Sadeh, N. (2014). A field trial of privacy nudges for facebook. In *ACN CHI* (pp. 2367–2376).

[117] Wang, Y., Leon, P. G., Chen, X., & Komanduri, S. (2013). From facebook regrets to facebook privacy nudges. *Ohio St. LJ*, 74, 1307.

[118] Wang, Y., Norcie, G., Komanduri, S., Acquisti, A., Leon, P. G., & Cranor, L. F. (2011). I regretted the minute I pressed share: A qualitative study of regrets on Facebook. In *ACM SOUPS* (pp.1̃0).

[119] Weinberger, J. & Felt, A. P. (2016). A week to remember: The impact of browser warning storage policies. In *SOUPS*.

[120] Wu, M., Miller, R. C., & Garfinkel, S. L. (2006). Do security toolbars actually prevent phishing attacks? In *ACM CHI*.

[121] Xie, W. & Kang, C. (2015). See you, see me: Teenagers' self-disclosure and regret of posting on social network site. *Computers in Human Behavior*, 52, 398–407.

[122] Xu, J.-M., Burchfiel, B., Zhu, X., & Bellmore, A. (2013). An examination of regret in bullying tweets. In *HLT-NAACL*.

[123] Yang, C. & Srinivasan, P. (2014). Translating surveys to surveillance on social media: methodological challenges & solutions. In *ACM Web science*.

[124] Yang, Y., Lutes, J., Li, F., Luo, B., & Liu, P. (2012). Stalking online: on user privacy in social networks. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*.

[125] Zarras, A., Kohls, K., Dürmuth, M., & Pöpper, C. (2016). Neuralyzer: flexible expiration times for the revocation of online data. In *ACM CODASPY*.

[126] Zhou, L., Wang, W., & Chen, K. (2016). Tweet properly: Analyzing deleted tweets to understand and identify regrettable ones. In *World Wide Web*.