ANONYMIZING PRIVATE PHRASES AND DETECTING DISCLOSURE IN ONLINE SOCIAL NETWORKS

NGUYEN SON HOANG QUOC

Department of Informatics

The Graduate University for Advanced Studies (SOKENDAI)

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2015

A dissertation submitted to the Department of Informatics,
School of Multidisciplinary Sciences,
The Graduate University for Advanced Studies (SOKENDAI)
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy.

Advisor

Prof. Isao ECHIZEN
National Institute of Informatics (NII),
The Graduate University for Advanced Studies (SOKENDAI)

Sub-advisors

Prof. Noboru SONEHARA and **Assoc. Prof. Yusuke MIYAO**
National Institute of Informatics (NII)
The Graduate University for Advanced Studies (SOKENDAI)

Advisory Committee

Prof. Shigeki YAMADA, **Prof. Hitoshi OKADA**
National Institute of Informatics (NII)
The Graduate University for Advanced Studies (SOKENDAI)
Prof. Akiko AIZAWA
National Institute of Informatics (NII)
The University of Tokyo (TODAI)
Prof. Akihisa KODATE
Tsuda College

Acknowledgements

First of all, I would like to deeply express my appreciation to my supervisor, Professor Isao Echizen. I am fully admire his knowledge, research skills, encouragement ways to overcome my tackles. Beside of the research, I would like to thank you for his financial support via research assistant. It helps me so much for my life in Tokyo. Therefore, I could concentrate into my PhD program. His supervision helps me not only for the doctor program but also for my future careers.

I am grateful for two sub-advisors, Professor Noboru Sonehara and Professor Yusuke Miyao. I also would like to thank you Professor Hiroshi Yoshiura (University of Electro-Communications), Professor Minh-Triet Tran (University of Science – VNU-HCM), an external committee member Professor Akihisa Kodate, and other committee members: Professor Shigeki Yamada, Professor Hitoshi Okada, and Professor Akiko Aizawa. They gave me many valuable comments and suggestions for my research topic. Moreover, they helped me so much to improve my research by expanding my knowledge in natural language processing and security infrastructure. I also say thank you for professors in my research courses and teachers in Japanese courses. I learned numerous useful knowledge for improving my backgrounds, research skills, and Japanese language.

I would like to send my thank you to Mrs. Yumiko Seino, NII staffs, and SOKENDAI staffs for their support. They have enthusiastically assisted me during the time when I not only research in NII but also attend international conferences. I also thank you NII and Japan Student Services Organization (JASSO) supporting me scholarships during my PhD program at SOKENDAI.

I would like to thank you to my laboratory members, NII-Vietnamese group, NII friends, especially Professor Duy-Dinh Le, Dr. Kien Nguyen, Dr. Minh-Quang Tran. I am very comfortable when I discussed with them about research and daily life. It helped me so much for balancing between the research and the life.

Finally, I would like to say special thank you to my family, especially my wife (Mrs. Nguyen Thi Hong Cuc) and my daughter (Miss. Nguyen Nguyet Que). They always encourage me with endless love even in long distance for a long time. They make me optimistic on every occasion.

Abstract

Online social networks (OSNs) have become an important part of modern life, and many people use one or more OSNs every day. However, the ease of collecting personal information from OSN messages can make users feel insecure when they access an OSN. Phrases containing personal information, i.e., "private phrases," should thus be identified and anonymized before they are posted on OSNs. Furthermore, if messages containing personal information that are posted only for friends to see are disclosed, the discloser should be identifiable.

Most previous research on identifying private phrases has focused on comparing candidate phrases with predefined phrases (such as personal names, locations, or diseases). However, attackers easily recognize and replace the phrases with similar ones (e.g., synonyms, generalizations). Such replacements can be identified by using a co-occurrence metric. In this case, non-private phrases, such as "*HIV*" in a non-private message "The human immunodeficiency virus (*HIV*) is a lentivirus (a subgroup of retrovirus) that causes the acquired immunodeficiency syndrome (AIDS)," might also be identified.

We have developed an algorithm for determining whether an OSN message is a private message or a non-private one. The F-score was 92% for 3000 messages, showing that it works well for identifying private messages. The algorithm identifies private messages on the basis of word frequency, so it is not suitable for private messages containing locational phrases. Consequently, we have developed a rule-based algorithm that overcomes this problem by using text semantics. It correctly identified 84.95% of 2917 locational messages, significantly higher than a machine learning method using word frequency and its three extensions (highest accuracy=81.93%).

Phrases that are identified as private might be released for only friends to see. For example, users sometimes share messages containing such information as e-mail addresses, hometowns, and locations. This information should be anonymized to avoid spam or advertising. One approach to such anonymization is to remove or replace the private phrases with generalizations, but this makes the text sound unnatural.

We have developed a way to improve the naturalness of generalized phrases. We have also developed a metric for quantifying information loss due to generalization so that anonymized messages can be distributed to different groups of friends with appropriate levels of privacy.

Time-related information in texts posted on-line is one type of private information targeted by attackers. This is one reason that sharing information online can be risky. We have developed an algorithm for creating anonymous fingerprints for temporal phrases that covers most potential cases of private information disclosure. The fingerprints not only anonymize time-related information but also can be used to identify a person who has disclosed information about the user. In an experiment with 16,647 different temporal phrases extracted from about 16 million tweets, the average number of fingerprints created for an OSN message was 526.05. This is significantly better than the 409.58 fingerprints created by a state-of-the-art algorithm. Fingerprints are quantified using a modified normalized certainty penalty metric to ensure that an appropriate level of information anonymity is used for each friend of the user.

The algorithm we developed to anonymize time-related private information removes the temporal phrases when doing so will not change the natural meaning of the message. The temporal phrases are detected by using machine-learned patterns, which are represented by a subtree of the sentence parsing tree. The temporal phrases in the parsing tree are distinguished from other parts of the tree by using temporal taggers integrated into the algorithm. In an experiment with 4008 sentences posted on a social network, 84.53% of them were anonymized without changing their intended meaning. This is significantly better than the 72.88% of the best previous temporal phrase detection algorithm. Of the learned patterns, the top ten most common ones were used to detect 87.78% of the temporal phrases. This means that only some of the most common patterns can be used to anonymize temporal phrases in most messages to be posted on an OSN.

As mentioned above, private messages could be revealed by a user's friends or even by the user, either unintentionally or intentionally. One approach to overcoming this problem is to create fingerprints by using linguistic steganography algorithms. Unfortunately, such algorithms create an insufficient number of fingerprints to cover all of a user's friends.

The algorithm we developed generates a sufficient number of fingerprints by using various combinations of synonymizations and generalizations. It creates, on average, 140.91 fingerprints per message. This is significantly higher than the 21.29 fingerprints created by the best fingerprinting algorithm using synonymization. In addition, attackers often modify their fingerprints into paraphrased ones. The similarity matching ($SimMat$) metric we developed for detecting paraphrases is based on matching identical phrases and similar words and quantifying the minor words. Evaluation using about 5800 paraphrase pairs taken

from a widely used paraphrase corpus created by Microsoft Corporation demonstrated the effectiveness of the *SimMat* metric. It achieved the highest paraphrase detection accuracy (77.6%) when it was combined with eight standard machine translation metrics. This accuracy is slightly better than the 77.4% rate achieved with the current state-of-the-art method for paraphrase detection.

We have developed a realistic system that controls the disclosure of both private and non-private messages on Facebook, the largest OSN. It automatically identifies private messages after a user composes them using the developed system. The system then recommends anonymous fingerprints for a user's friends on the basis of private phrases in the private messages. This system also detects the disclosure of fingerprinted information and the person who disclosed it.

The system not only efficiently controls the disclosure of private information in OSN messages but also improves security in other sensitive fields (e.g., health, military, and politics). Furthermore, our proposed algorithms have been proven to be common types of attack (such as paraphrasing and generalizing). Although they are poor at identifying private information in a single message, such information can be inferred from various messages (past OSN messages, blogs, web pages, etc.).

Future work includes enhancing the algorithms to enable them to identify private information by inferring it from various messages. In the next stage, we will focus on improving the naturalness and semantic coherence of anonymous fingerprints. In addition, the algorithms will be modified to enable them to anonymize private objects (such as human faces, vehicle license numbers, and home addresses) in digital media (audio, image, video, etc.).

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation and objectives

1.1.1 Sensitive information

Sensitive information is the control of access to information that might result in a lower level of security if it is disclosed to others [29]. There are three types of sensitive information: government, business, and private.

- *Government information* is information (such as information related to strategy development and national interests) that belongs to a specific government entity. Release of this information is frequently authorized for individuals who have different levels of security. Unauthorized disclosure of such information can create serious problems. Disclosers may be punished by imposing on them a fine, a prison sentence (suspended or immediate), or even the death penalty depending on the degree of the violation. Since ancient times, countries have employed spies to steal the sensitive information of other countries.

- *Business information* is information that belongs to a company and typically includes product plans, trade secrets, and financial data. Its unauthorized disclosure can harm the interests of the company. The regulation of its disclosure is usually handled by individual companies. Employees at many companies have to sign a nondisclosure agreement and are subject to punishment if they disclose such proprietary information to unauthorized persons.

- *Private information* is information that belongs to an individual person and can include important information such as credit card numbers and bank account details. A great

amount money can be lost if such information is stolen [1]. Other types of private information (such as e-mail addresses and home addresses) are sometimes illegally obtained by commercial companies and used to generate spam.

There are privacy issues, and many laws have been created to protect user information. For example, EU law [22] forbids the disclosure of personal data. This means that a party cannot disclose a user's personal information to third parties without the user's prior authorization or as required by law.

1.1.2 Private information in online social networks

Although many such laws protect a user's private information, it is often disclosed in online social networks (OSNs), both intentionally and unintentionally. In most of cases, private information is revealed by either the user or one of the user's OSN friends. For example, Stutzman et al. analyzed the Facebook accounts of more than 5,000 students at Carnegie Mellon University and recovered the user's real name for 89% of them, the birthday for 88%, and the current residence for 51% [79]. In another example, analysis of 592,548 accounts, including comments of users and their friends, on Wrech, the biggest OSN in Taiwan, by Lam et al. showed that 72% revealed the account holder's given name, 30% revealed the full name, 15% revealed the age, and 42% revealed the school attended [44].

A decade or so ago, OSN users rarely changed their default settings. For example, Gross and Acquisti reported that most users did not change their default settings [30]. Furthermore, a survey by Ellison et al. showed that only 13% of the Facebook users in the Michigan State University network had limited their information sharing to "friends only." [25] The situation has changed, however; OSN users are now more aware of these settings and many now change them. In a report by the Pew Internet & American Life Project[2], 71% of OSN users between the ages of 18 and 29 changed their OSN privacy settings. This trend towards protecting one's privacy on an OSN means that there is a stronger demand for automatic privacy protection, especially OSN messages, which obtain a lot of private information.

Users often unintentionally disclose their private information via OSN messages. For example, a user expose his/her private locational information via an OSN message "Yesterday, I met Yoko at *Chofu Station*. She looked tired from doing graduate research, like I have."[3] However, some messages may include the locational information but they do not disclose private information such as "To get to Chofu Airport, take the Keio Railway from Shinjuku

[1] http://krebsonsecurity.com/2015/01/how-was-your-credit-card-stolen/
[2] http://pewInternet.org/~/media//Files/Reports/2010/PIP_Reputation_Management.pdf
[3] The OSN message is retrieved from mixi (https://mixi.jp/), the largest OSN in Japan

to *Chofu Station*."[4] Therefore, OSN messages are classified as either private or non-private. *Private messages* disclose private information about individual person while *non-private messages* do not. Phrases in non-private message are called *non-private phrases*. Such phrases, which indicate private information in private messages, are considered as *private phrases*.

1.1.3 Objectives

Due to the demand for improved privacy of OSN messages, our objectives focus on protecting private phrases and thereby avoiding the disclosure of private information. The first objective is to identify private phrases in private messages. The second objective is to anonymize the private phrases before they are posted via OSNs. The last objective is to detect the discloser if private phrases are disclosed. The challenges of meeting these three objectives are described in more detail in the next section.

1.2 Challenges

1.2.1 Identification of private phrases

Many methods have been proposed for identifying private objects in digital media (such as images, audio, and videos). These objects (e.g., people, faces) are identified by using markers (color of hat/vest [72], buttons [62], clothing patterns [18], hand gestures [6], etc.). However, these methods are only suitable for binary data; they cannot identify private phrases in messages.

Methods for identifying suspicious phrases in OSN messages were explored in the DCNL (Disclosure Control of Natural Language Information) Project [38]. The DCNL Project explored not only the identification of direct suspicious phrases but also discovering indirect suspicious phrases. However, the resulting DCNL method simply identified private information on the basis of suspicious phrases in the text although many non-private messages containing suspicious phrases may not actually reveal any private information about the author. For example, suspicious words and phrases, such as "*flu*" and "*H5N1*" in the non-private message "Unlike other types of *flu*, *H5N1* usually does not spread between people" are also identified although this non-private message does not disclose private information. On the other hand, Acquisti and Gross proposed an algorithm to infer users' social security number from their addresses and birthday [2]. Furthermore, with sufficient

[4] http://www.japan-guide.com/e/e8252.html

private information, Chakaravarthy et al. proved that private diseases of a person are easily inferred by attackers [11]. For another example, innocent people are identified via writing-style [1, 89]. However, these approaches identify private information from many messages. It inefficiently infer private phrases in a single short message.

A popular type of private phrases is a locational ones. Most approaches have essentially focused on identifying locational phrases in a general message [28, 68]. For example, a location "*Tokyo*" is identified in a general message "*Tokyo* is very wonderful city," but it does not leak user private information. Another approach identifies a user's location registered in an OSN by analyzing multiple his/her messages. However, this approach cannot apply for tracing the private locational phrases in a single short message.

1.2.2 Anonymization of private phrases

Most methods for anonymizing private information are for digital media (such as videos and images). For example, private objects (such as faces and car license numbers) are blurred before they are broadcast via the Internet [3]. These algorithms are suitable for binary data but not for text messages. Other methods anonymize the private attributes of databases by removing or replacing them with generalizations [9, 39, 73]. These methods only apply for clearly meaning of items in a database; they cannot be used for natural language messages.

Research on anonymizing private information in text has focused on deleting private phrases [42], but an attacker can easily recognize the modified text. Other algorithms use generalizations for anonymizing private phrases [45]. However, the anonymous messages are unnatural.

OSN users usually organize similar friends with same properties (such as relative, interests) in a group (e.g. family, acquaintance, and co-worker) and such groups have different levels of privacy. One approach to determine the privacy level of a OSN group is based on frequency of interaction between the owner user and friends in the group [47]. It is necessary to create message with different amount of information before they are assigned for groups with appropriate levels of privacy. For example, friends in *Best Friends* group should receive more specific information than friends in *Public* one. Information loss metric is used to estimate the quantity of the information. Many metrics are suggested for quantifying the information loss in database [70]. However, these metrics are not suitable for social media (such as OSN messages).

The significant type of private information is time-related information. By extracting and analyzing time information in posted text, attackers can determine a user's habits and use that information for illegal purposes (stalking, robbery, kidnapping, etc.). Therefore, time information should be anonymized before it is posted on an OSN. Research on anonymizing

information in text has focused on replacing private phrases with anonymous phrases [42], but an attacker can often recognize the modified text on the basis of the anonymized phrases. Some research on anonymizing information has investigated the use of a semantic dictionary to generalize private phrases [71]. However, these approaches are not applicable to anonymizing temporal phrases because a generalization dictionary cannot be created for temporal phrases. Although there are many approaches to identifying temporal phrases [13, 60, 78, 82], simply deleting them makes the meaning of the message unnatural.

1.2.3 Detection of disclosure

Well-known applications of detecting disclosure are copyrights of digital media (such as image, video, audio). For example, YouTube, the most famous video-sharing in the world, creates a law for banning upload copyrighted videos. However, the thousands of the copyrighted videos are still uploaded per day[5]. It makes harmful for owner companies which lose about 5000 crore rupees per year (it is equivalent to the cost for hiring about 50000 jobs/year)[6].

One way to identify a person who has disclosed private messages is to "fingerprint" the posted messages. A fingerprint is simply a different way of saying the same thing. The message is fingerprinted differently for each friend receiving it. This enables identification of the friend who has disclosed private information. Messages can be fingerprinted, for example, by reordering their structure [88], using paraphrasing [14], or by synonymizing [90]. However, these methods cannot create a sufficient number of unique fingerprints for all the friends of a typical OSN user.

Plagiarism usually modified their fingerprinted messages into paraphrases ones. Some researchers in the field of paraphrase detection have used vector-based similarity to identify the differences between two sentences [8, 52]. The two sentences are represented by two vectors based on the frequency of their words in text corpora. The vectors are compared to estimate sentence similarity. Plagiarists attempt to thwart this comparison by modifying the copied sentence by inserting or removing a few minor words, replacing words with similar words that have different usage frequencies, etc. Such modification reduces the effectiveness of vector-based similarity analysis.

Other researchers have analyzed the difference in meaning between two sentences on the basis of their syntactic parsing trees [19, 67, 77]. The structure of the trees is a major factor used various sophisticated algorithms such as recursive autoencoders [77], heuristic

[5]http://mashable.com/2012/02/17/youtube-content-id-faq/

[6]http://articles.economictimes.indiatimes.com/2013-03-05/news/37469729_1_upload-videos-youtube-piracy

similarity [67], and probabilistic inference [19]. However, these algorithms are affected by manipulation (deleting, inserting, reordering, etc.) of the words in the sentences. Such manipulations can significantly change the structures of the parsing trees.

Other researchers [12, 52] have used matching algorithms to determine the similarity of two sentences. Mihalcea et al. [52], for example, proposed a method for finding the best matching of a word in a sentence with the nearest word in the other sentence. However, word meaning is often contextual (e.g., 'make sure of,' 'take care of').

Machine translation (MT) metrics, which are generally used to evaluate the quality of translated text, can also be used to judge two texts in the same language. Due to the similarity of machine translation and paraphrase detection, many MT metrics have been applied to paraphrase detection [26, 48]. For example, eight standard MT metrics have been combined to create a state-of-the-art paraphrase detection approach [48]. However, the objectives of machine translation and paraphrase detection differ: machine translation tries to effectively translate text from one language to another while paraphrase detection tries to identify paraphrased text. This difference affects the application of MT metrics to paraphrase detection.

1.3 Research contributions

We propose a system for automatically controlling private phrases in OSN messages. The system has three main processes: identification of private phrases, anonymization of private phrases and detection of disclosure. The three processes are illustrated in Fig. 1.1 with a message composed by a user Adam "Mary comes from *Tokyo* and studied at *MIT*."

In the *identification of private phrases* process, the system receives an input message (such as a blog, comment, or status) from the user. The system then automatically identifies the private phrases in the message. In example used in Fig. 1.1, two private phrases "Tokyo" and "MIT" are identified.

In the *anonymization of private phrases* process, the system automatically anonymizes the private information, creates differently fingerprinted versions of the message. The system then posts the fingerprinted messages so that each friend sees the appropriate version. In example of Fig. 1.1, three fingerprinted messages "Mary *lives in* Tokyo and studied at MIT," "Mary comes from Tokyo and *learned* at MIT," and "Mary comes from Japan and studied in *USA*" are assigned for Ellen, Bob, and Smith correspondingly.

In the *detection of disclosure* process, the system collect candidate messages from the Internet by using fingerprints. Each candidate message is compared with fingerprinted messages to determine whether they have same meaning by our paraphrase detection algorithm.

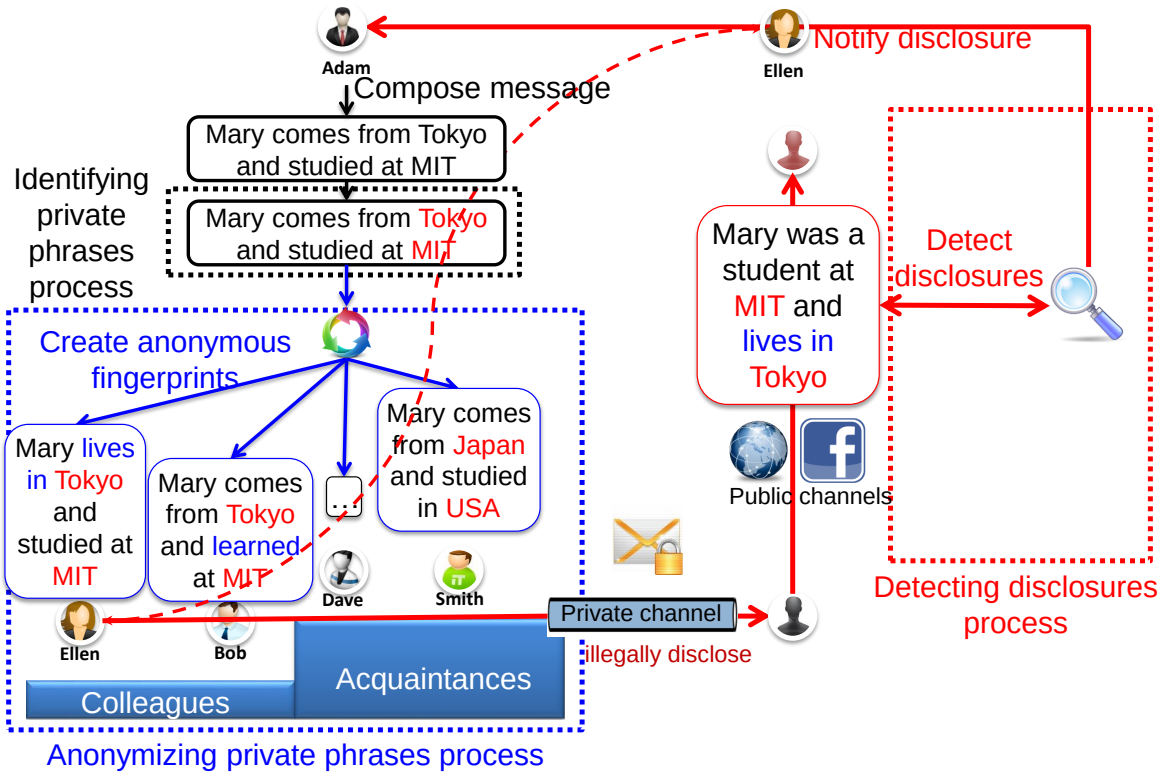


Fig. 1.1 Overview of proposed system.

In case of Fig. 1.1, Ellen sends her friend a copy of Adam's message via private channel (e.g., email, phone, SMS message). Such friend then paraphrases the message in an attempt to avoid detection and post the modified message on a public channel (such as Facebook, blog). However, our system can still detect this disclosure and notify Adam of the disclosure.

The major contributions of this thesis for each process are described as follows:

1.3.1 Identification of private phrases

- *Identification of private phrases by similarity metric*: We develop an algorithm that can be used to identify private phrases in OSN messages. First, we identify a classifier to determine whether a composed message is either a private message or a non-private message to ensure that only private messages are identified private phrases. Finally, we estimate a threshold of co-occurrence metric used to identify private phrases. This prevents attackers from replacing private phrases with similar phrases to avoid identification.

- *Identification of private locational phrases by a rule-based approach*: We develop a rule-based approach for identifying private locational phrases of an OSN message. In

particular, we analyze context and grammar of messages for exploring relationships between the locational phrases and user phrases.

1.3.2 Anonymization of private phrases

- *Anonymization of private phrases by generalization*: We report an algorithm for anonymizing private information in OSNs. The algorithm anonymizes the personal information by generalizing private phrases. It then creates different versions of the message, each with a unique fingerprint, by synonymizing phrases in the message, thereby enabling the detection of disclosure. We propose a distribution metric for quantifying information loss due to anonymization so that the appropriate degree of anonymization can be determined for each group of friends. We estimate a threshold of frequency metric to improve the naturalness of fingerprinted messages. This metric is used to check the substituted phrases used for fingerprinting. The use of this threshold ensures that our algorithm creates a sufficient number of fingerprints for friends and that the fingerprinted messages are natural.

- *Anonymization of temporal phrases by generalization*: We propose an algorithm to automatically anonymize temporal phrases by using generalized ones which are extracted from OSN messages. It then uses as fingerprints in detecting disclosure of personal information. We develop a modified normalized certainty penalty metric to ensure that an appropriate level of information anonymity is used for each user's friend.

- *Anonymization of temporal phrases by deletion*: We analyzed OSN messages and found that the temporal phrases were generally independent of the rest of the message. This means that the temporal phrases can generally be deleted without damaging the grammatical structure of the sentences. On the basis of this finding, we develop an algorithm that uses machine learning to identify the temporal patterns that can be used for identifying and anonymizing temporal phrases in the text without losing its natural meaning. The patterns are a subtree of each sentence's parsing tree. All temporal phrases that have the same structure as a pattern are identified. However, many phrases that are not temporal phrases can also have the same structure. This problem is overcome by integrating temporal taggers of time temporal expressions identification algorithm into time pattern. As a result, only the parsing subtree with time tags is used for deleting temporal phrases.

1.3.3 Detection of disclosure

- *Detection of paraphrase*: We present a heuristic algorithm for finding an optimal matching of *identical phrases* with maximum lengths. We suggest removing the *minor words* from the words remaining in the sentences. These minor words include prepositions, subordinating conjunctions (‘at,’ ‘in,’ etc.), modal verbs, possessive pronouns (‘its,’ ‘their,’ etc.), and the period (‘.’). We present an algorithm for determining the perfect matching of *similar words* by using the matching algorithm proposed by Kuhn and Munkres [43, 54]. The degree of similarity between two similar words is identified using WordNet [65]. These similarities are used as weights for the matching algorithm. We present a related matching (*RelMat*) metric for quantifying the relationship between two sentences on the basis of matching identical phrases and similar words. We present a brevity penalty metric to reduce the effect of paraphrased sentence *modification*. This metric is combined with the *RelMat* metric into a similarity matching *SimMat* metric for effectively detecting paraphrases.

1.4 Organization

This thesis is organized as follows. Chapter 2 presents literature review about identifying, anonymizing private phrases and detecting disclosure of private ones. Chapter 3 describes algorithms on identifying private phrases. The identified private phrases are anonymized in Chapter 4. Chapter 5 presents an algorithm for detecting disclosure of the private information and an application for demonstrating the practicality of the proposed algorithms. Chapter 6 summarizes some key points and mentions future work.

Chapter 2

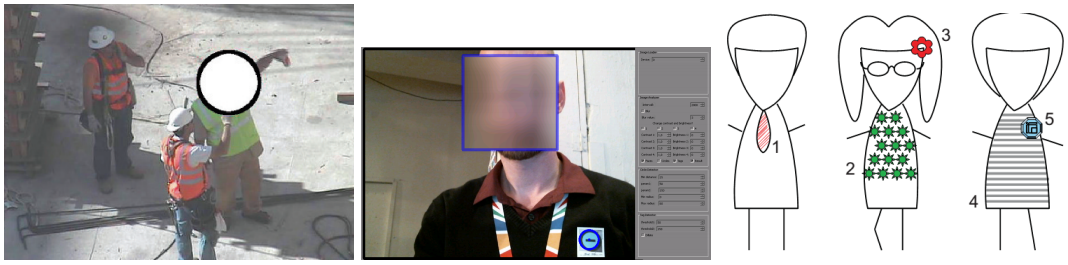
Literature review

2.1 Identification of private phrases

2.1.1 Identification of private information

Research on identifying private information has focused on videos and images. The most common type of private information is the human face. This private information is identified by using a specific marker (e.g., hat/vest color [72] (Fig. 2.1a), a particular button [62] (Fig. 2.1b), a clothing pattern [18] (Fig. 2.1c), or a hand gesture [6]). These markers exist only in visual data, so they cannot be used for text.

Technology for identifying private items in a database has concentrated on the use of predetermined private attributes (e.g., disease, age, hometown) [9, 35, 39]. The advantage of this approach is that such attributes have obvious meanings. Therefore, the private information in a database can be easily identified by analyzing the attributes' meanings. However, this approach is not suitable for natural language text because the structure of the text differs from that of the database.



(a) Hat/vest color. (b) A particular button. (c) Clothing patterns.

Fig. 2.1 Identifying private objects in visual data by using markers.

There are two types of identifying private information on natural language text: *identifying private features* and *identifying private phrases*. Research on *identifying private features* has focused on identifying an individual person via his/her writing-style [1, 89], or identifying social security numbers via dates of birth [2]. On the other hand, most algorithms for *identifying private phrases* are based on pre-defined words (such as diseases [11], organizations, and/or locations). The drawback of these algorithms is that some pre-defined words are non-private phrases such as a disease word "*HIV*" in a message "*HIV* is popular in Africa." Other algorithms identify private phrases on the basis of word similarity using information theory [71] or co-occurrence metric [38]. The limitation of these algorithms is that they need a huge text corpus for efficiently quantifying similarity of words.

2.1.2 Identification of private locational phrases

Identification of locational phrases

A number of studies investigating locational phrases identification have based upon name entity recognition (NER). A famous one, Stanford NER library, is created by Finkel et al. for extracting locational phrases in general text [28]. This library utilized Gibbs sampling, a simple Monte Carlo method to solve long distance. The F-measure was 86.86% since it is evaluated with the CoNLL 2003 named entity recognition dataset illustrated the efficient of the library.

However, general NER algorithms are not suitable for OSN messages which contain numerous noises, incorrect spelling words, and unstructured grammars. A few locations phrases of OSN messages, for example, are identified by using the Stanford library (F-measure=29%). Because of this, Ritter et al. developed a NER library for improving the performance in the identification (F-measure=59%) [68]. This library is thus used as a pre-processing step of our algorithm for identifying private locational phrases. The algorithm is presented more detail in Chapter 3.

Identification of physical locations

Name entity recognition is extended for identifying physical positions of various applications on the Internet. A notable example of those is to identify an original position of a web page based on the distribution of locational phrases [4]. In another example, Fink et al. proposed a method for identifying the positions blogs by analyzing contexts of blogs' texts [27].

On the other hand, identification of positions is a fundamental property of multiple applications in maps (such as navigation system and person tracking system). There are

many famous maps (e.g., Google maps, Bing maps, Yahoo maps, or GeoNLP maps for Japan [40]) supporting APIs for doing this. These APIs receive a query including a phrase and some arbitrary options. Such phrase is then analyzed its contexts and estimated a position. The position including longitude and latitude is used as essential information for the map applications.

A notable approach identifies a position, which is registered by a user on an OSN [17]. All messages of a user are collected for inferring the position. This approach bases on local words for identifying the location. For example, a local word "*casino*" is talked in *Las Vegas* more frequent than other cities while "*rocket*" is commonly mentioned in *Houston*. However, this approach was not suitable for pointing out private phrases on a single and very short message. Therefore, we propose a rule-based algorithm to identify private locational phrases for a single OSN message by analyzing contexts around locational phrases, as described in Chapter 3.

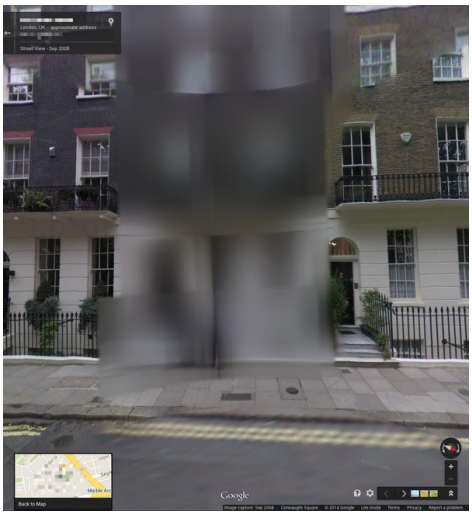
2.2 Anonymization of private phrases

2.2.1 Anonymization of private information on digital media

Many approaches have been proposed to anonymize private information on *image or video*. Some approaches do this by blurring private information. For example, Google supports a function for anonymizing users' houses out of the Google street view[1] (illustrated in Fig. 2.2a). For another example, other private information (such as license plates, or address) should be hidden from images (videos) before they are posted on the Internet[2] (Fig. 2.2b). Other approaches use infrared waves for protecting some private objects (such as monitor [84], film [85]...) as demonstrated in Fig. 2.3. In these approaches, the private objects are able to see by personal eyes but they are blurred if they are captured by digital cameras.

Other approaches have focused on anonymizing *private facial people*. For example, CV Dazzle method proposed by Harvey uses hair and makeup to anonymize private faces to avoid identifying them [34] (Fig. 2.4a). Other approaches use privacy glasses or infrared glasses to avoid facial identification software [86, 87] (Fig. 2.4b).

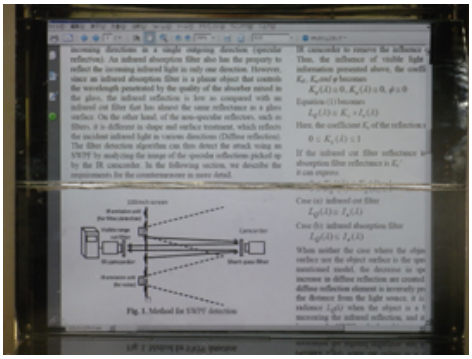
[1] http://www.welivesecurity.com/2014/07/10/remove-house-google-street-view/
[2] http://jalopnik.com/5941797/should-you-bother-obscuring-your-license-plates-in-photos



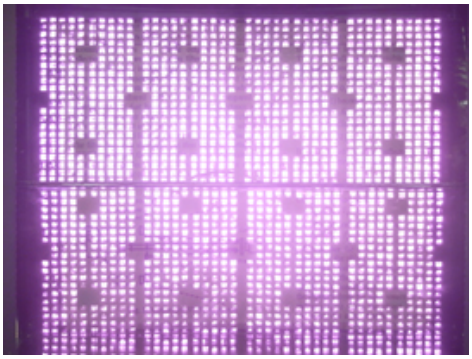
(a) House anonymization.

(b) License plates anonymization.

Fig. 2.2 Anonymizing private objects by using blurring.



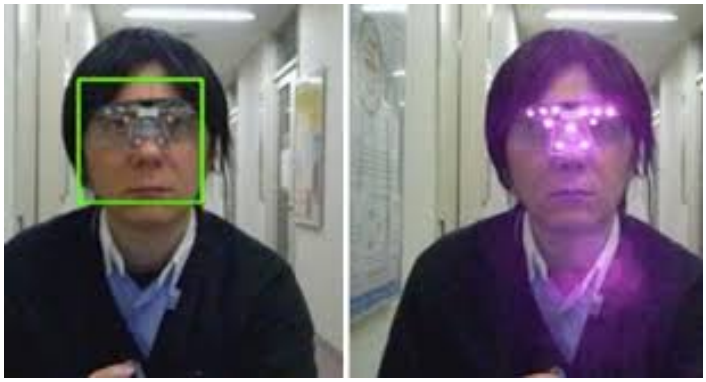
(a) Monitor before anonymizing.



(b) Monitor after anonymizing by using infrared waves.

Fig. 2.3 Anonymizing monitor by using infrared waves.

(a) CV Dazzle.



(b) Privacy visor.

Fig. 2.4 Techniques for avoiding face identification.

2.2.2 Anonymization of private phrases

Anonymization makes a user's personal information sufficiently vague so that identifying the user is difficult. Many methods for anonymizing personal information in natural language text simply remove all private phrases [42], as illustrated in Fig. 2.5. Others replace private information with appropriate categorical words or phrases (such as *location*, *person*, and *organization*) [51]. These methods use name entity recognition to identify entities in messages and anonymize them. However, some OSN messages containing candidate phrases do not disclose private information about the user. For example, the non-private message "*Tokyo* is the capital of *Japan*" does not disclose private information. Therefore, we classify a message containing candidate phrases as either a private message or a non-private message. *Private messages* disclose private information about the user while *non-private messages* do not. We anonymize private phrases only in private messages.

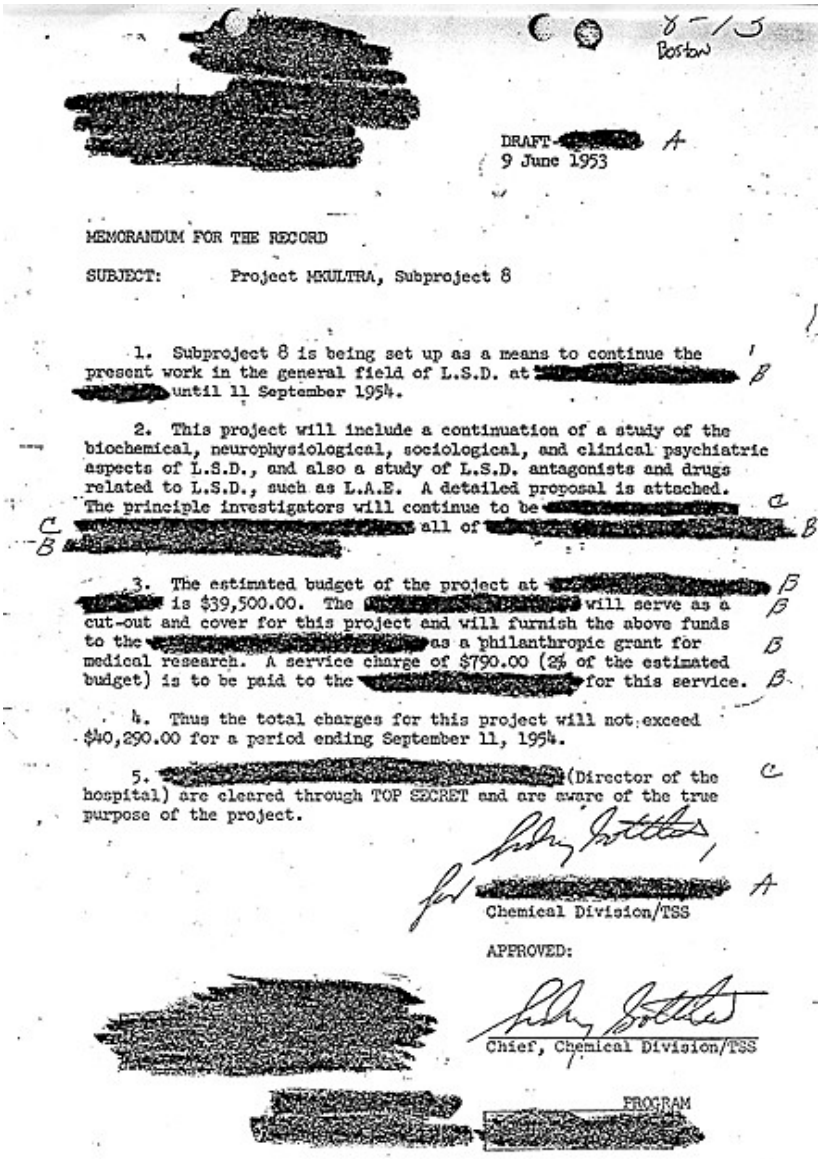
A previous method anonymizes private phrases by generalization [71], but the anonymized messages are unnatural. Attackers recognize such unnaturalness and focus on changing private information to avoid disclosure detection. We have improved the naturalness of anonymized messages by using a frequency metric. Such metric is presented in more detail in Chapter 4.

Metric for quantifying loss due to anonymization

Anonymizing private phrases by generalization results in information loss. Generalization schemes for two quasi-identifiers, "Prefecture" and "University," are diagrammed in Fig. 2.6. Since friends in the *Family* group should receive messages with a lower degree of generalization than those in the *Public* group, friends in the *Family* group should receive the version of the message that has "MIT," for example, and those in the *Public* group should receive the version that has "United States."

One way to quantify information loss for a set of N private phrases $P = \{p_i\}$ is to use the Samarati metric (*Sam*) [70], which is based on the degree of generalization of the i-th phrase l_{p_i}, as shown in Eq. 2.1. For example, if the message contains two private phrases, "*France*" (p_0, degree 1) and "*United States*" (p_1, degree 1), the *Sam* metric is 2. The disadvantage of this metric is that it only gives the degree of generalization of each phrase while the schemas may have different heights. For example, "*France*" and "*United States*" should not have the same metric value because "*France*" still generalizes to "*Europe*."

$$Sam(P) = \sum_{i=0}^{N-1} l_{p_i} \tag{2.1}$$



[redacted]

DRAFT-[redacted] A
9 June 1953

MEMORANDUM FOR THE RECORD

SUBJECT: Project MKULTRA, Subproject 8

1. Subproject 8 is being set up as a means to continue the present work in the general field of L.S.D. at [redacted] until 11 September 1954.

2. This project will include a continuation of a study of the biochemical, neurophysiological, sociological, and clinical psychiatric aspects of L.S.D., and also a study of L.S.D. antagonists and drugs related to L.S.D., such as L.A.E. A detailed proposal is attached. The principle investigators will continue to be [redacted] all of [redacted]

3. The estimated budget of the project at [redacted] is $39,500.00. The [redacted] will serve as a cut-out and cover for this project and will furnish the above funds to the [redacted] as a philanthropic grant for medical research. A service charge of $790.00 (2% of the estimated budget) is to be paid to the [redacted] for this service.

4. Thus the total charges for this project will not exceed $40,290.00 for a period ending September 11, 1954.

5. [redacted] (Director of the hospital) are cleared through TOP SECRET and are aware of the true purpose of the project.

[redacted]
Chemical Division/TSS

APPROVED:

[redacted]
Chief, Chemical Division/TSS

[redacted] PROGRAM

Fig. 2.5 Anonymizing private information on texts by suppression.

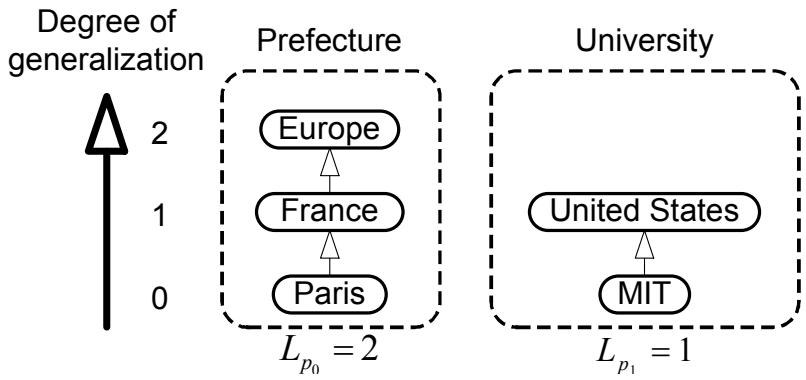


Fig. 2.6 Generalization schemes for two quasi-identifiers.

The precision metric (*Pre*) [70] is calculated on the basis of the number of possible degrees of generalization l_{P_i} with P_i being the highest degree of generalization of private phrase p_i, as shown in Eq. 2.2. It overcomes the disadvantage of the *Sam* metric because, for example, the *Pre* metric of "*France*" (degree 1/2) and that of "*United States*" (degree 1/1) is 1.5. Although this metric automatically quantifies information loss on the basis of the scheme's structure, it is not suitable for practical use. For example, the *Pre* and *Sam* metrics for "*MIT*" and "*Paris*" are 0. However, the number of students studying at *MIT* is around 11,000 per year while over 2 million people live in *Paris* [80]. Therefore, "*MIT*" and "*Paris*" should have different metric values for information loss.

$$Pre(P) = \sum_{i=0}^{N-1} \frac{l_{p_i}}{l_{P_i}} \tag{2.2}$$

Ngoc et al. proposed a metric based on probability and entropy [55] that overcomes the problems with the *Sam* and *Pre* metrics. This metric uses a dataset containing the number of students at each university in Japan created by the Inter-Business Associates Corporation[3]. However, this metric simply quantifies information loss within the scope of universities in Japan. A metric is needed that automatically quantifies all private phrases on the basis of actual data.

To build an automatic anonymization algorithm, we propose a distribution metric (*Dis*):

$$Dis(P) = \sum_{i=0}^{N-1} \frac{log(|p_i|)}{log(|P_i|)}, \tag{2.3}$$

[3]http://www3.ibac.co.jp/univ1/mst/info/univinfo_50.jsp

where $|x|$ is the population of private phrase x. *Dis* is explained in more detail in the Chapter 4.

2.2.3 Anonymization of temporal phrases

Anonymization of temporal phrases by generalization

Information is typically generalized by using a semantic dictionary (such as WordNet, Yago [80], or Wikipedia) to find generalizations for phrases in text. Using this approach, we developed an algorithm for creating anonymous fingerprints to protect private information posted on OSNs [56–58] and to identify possible disclosures. However, semantic dictionaries can only be used to apply a word or simple phrase having a certain meaning. They cannot be used to find generalizations for most temporal phrases (such as "at 9 a.m.," "in 25 minutes," "for a long time").

Metric for quantifying loss due to anonymization of temporal phrases

The use of generalization to anonymize personal information results in information loss. Various metrics have been proposed for quantifying this loss. The higher the level of generalization, the greater the loss and the greater the value of the metric.

The Samarati metric (S) [69], precision metric (P) [69], discernibility metric (DM) [7], and classification metric (CM) [36] are proposed to quantify the loss of information due to generalization. However, they take into account the discrete data. They do not reflect the continuous data such as temporal phrases.

Therefore, we propose a modified normalized certainty penalty metric (NCP^*) described in Eq. 2.4 on the basis of the normalized certainty penalty metric (NCP) [83] previously used for databases. Time of original temporal phrase $time_O$ happens between $start_O$ and end_O while duration of the anonymized temporal phrase $time_A$ is between $start_A$ and end_A. However, $start_X$ and end_X is equal in case of $time_X$ is a point time (such as "10a.m.", "2 o'clock"). Therefore, we add-one in numerator and denominator to ensure that this metric could apply for these cases. The detail of this metric is described in Chapter 4.

$$NCP^*(time_O, time_A) = \frac{end_O - start_O + 1}{end_A - start_A + 1} \tag{2.4}$$

Anonymization of temporal phrases by suppression

Several research efforts have focused on identifying private phrases (location, organization, person, etc.) and replacing them with anonymous phrases [42]. For example, the

iLexIR NLP Consultancy has created a Web site for demonstrating anonymization of text[4]. However, the anonymized text is unnatural and the time information is not always suppressed. This means that the suppression can be easily recognized from the anonymized phrases. An attacker can thus identify the anonymized parts of the text and analysis them.

In our approach, we anonymize the activity of a user by completely removing the temporal phrases in text to be posted on an OSN. Moreover, the anonymous messages have a natural meaning, so an attacker cannot recognize that a message has been anonymized.

Many technologies have been proposed for deleting information from text but none for anonymizing it (such as summarization, text simplification, sentence compression, adjective deletion [16]). However, these technologies are not appropriate for text in OSN messages because such messages are usually very short (status, comment, etc.). One approach to anonymizing on the sentence level is adjective deletion [16]. However, there is usually one adjective at most in a sentence while a temporal phrase can contain several words. Moreover, adjectives greatly depend on the other parts of the sentence (noun, adverb, or another adjective). Therefore, the naturalness of the sentence after adjective deletion depends on the relationship between the deleted adjective(s) and the words remaining in the sentence. This approach thus cannot be applied to temporal phrase deletion because, in English, the temporal phrase is mostly independent of the other phrases in the sentence.

In the case of OSN messages, temporal phrases tend to have similar grammatical structures between messages. We use this characteristic in our proposed algorithm. Only the time information phrases are removed in sentence anonymization. The naturalness and meaning of the sentence (except for the time information) is preserved.

The first step in anonymizing the temporal information is temporal phrase identification. Therefore, we summarize several techniques for temporal phrase identification.

The method proposed by Noro et al. [60] for identifying temporal phrases uses machine learning. The machine learning output of a support vector machine, a naive Bayes algorithm, and an expectation maximization algorithm are used together to identify four periods of time (morning, daytime, evening, and night). While this method can identify implicit temporal phrases such as “nap (daytime)” and “sunset (evening),” it is only able to classify text into the four periods of the day, which is useful for marketing research purposes, it cannot be applied to general time (weeks, months, years).

The main task for temporal phrase identification is a task in SemEval 2010[5], an international competition on natural language processing. In this task, the best performance was achieved by the rule-based Heideltime method [78] (F-score=0.86), and the second best

[4] http://www-dyn.cl.cam.ac.uk/~bwm23/anon/anon.php?response

[5] http://www.timeml.org/tempeval2/

performance was achieved by the TRIPS/TRIOS method [82] (F-score=0.85), which uses a conditional random field (CRF) formulation for finding temporal expressions. After the task in SemEval 2010, the rule-based SUTime tool, developed by Chang in 2012 [13] for identifying temporal phrases, has the best performance (F-score=0.92) to date. While these rule-based approaches have very high precision, they cannot identify some temporal phrases with complicated semantics such as: "a year and a half ago." In this case, they would simply identify "a year" as the temporal phrase.

2.3 Detection of disclosure

2.3.1 Fingerprint

Most methods for detecting disclosure of personal information use fingerprinting. Fingerprinting a message differently for each friend receiving it enables the person who discloses private information in the message to be identified.

Many methods create fingerprints by changing the form of a text message (e.g., active or passive) and/or the structure (simple or complex) [81]. Others use semantic transformation based on word sense disambiguation, semantic role parsing, or anaphora resolution to create fingerprints [5]. The payload of each method is about 0.5 fingerprints per message. The method proposed by Zheng et al. [90] replaces words in a message with synonyms on the basis of the context. It can create an average of 21.29 fingerprints per OSN message. However, this number of fingerprints is insufficient for an OSN user.

2.3.2 Detection of paraphrase

Attackers usually paraphrase messages for avoid detecting of fingerprints. The baseline for paraphrase detection is based on vector-based similarity. Each source message and target message is represented as a vector using the frequencies of its words (such as term frequency [52] and co-occurrence [8]). The similarity of the two vectors is quantified using various measures (e.g., cosine [52], addition and point-wise multiplication [8]). The problem with vector-based methods is to focus on the frequency of separate words or phrases. However, plagiarists can paraphrase by replacing words with similar words that have a very different frequency. Moreover, they can delete and/or insert minor words that do not change the meaning of the original sentences. Such manipulations change the quality of the representation vector, which reduces paraphrase detection performance.

Several methods have been proposed for overcoming the manipulation problem that use syntactic parsing trees of messages. The replacement of similar words and the use of minor

words do not change the basic structure of the trees. Qui et al. [67] reported a method that detects the similarity of two sentences by heuristically comparing their predicate argument tuples, which are a type of syntactic parsing tree. The high paraphrase recall (93%) it attained shows that most paraphrases have the same predicate argument tuples. However, the accuracy was very low (72%). Parsing trees were used for probabilistic inference of paraphrases by Das and Smith [19].

Another method considers these trees as input for a paraphrase detection method based on recursive autoencoders [77]. The drawback of the parsing tree approach is that parsing trees are affected by the reordering words in a sentence such as the conversion of a sentence from passive voice to active voice. Another method finds the maximum matching for each word in two sentences [52]. The similarity of matching two words is based on WordNet. However, the weakness of this method is that a word in a first sentence is probably matched to more than one word in the second sentence. This means that a very short sentence can be detected as a paraphrase of a long sentence in some cases. Another problem with word matching is that the meaning of some words depends on the context. For example, the basic meaning of 'get' changes when used in the phrasal verb 'get along with.'

Commonly used techniques for detecting paraphrases are based on MT metrics. This is because the translation task is very similar to the paraphrase detection task for text in the same language. For example, Finch et al. [26] extended a MT metric (PER) and combined it with three other standard metrics (BLEU, NIST, and WER) into a method for detecting paraphrases. Another method developed by [48] is based on the integration of eight metrics (TER, TERp, BADGER, SEPIA, BLEU, NIST, METEOR, and MAXSIM). However, the main purpose of these metrics is for translating, and their integration is unsuitable for detecting paraphrases. To overcome these weaknesses, we developed a similarity metric and combined it with eight standard metrics, as described below.

Standard MT metrics

Two basic MT metrics for measuring the similarity of two text segments are based on finding the minimum number of operators needed to change one segment so that it matches the other one. The translation edit rate (TER) metric [75] supports standard operators, including shift, substitution, deletion, and insertion. The TER-Plus (TERp) metric [76] supports even more operators, including stemming and synonymizing.

The BADGER MT metric [64] uses compression and information theory. It is used to calculate the compression distance of two text segments by using Burrows-Wheeler transformation. This distance represents for probability that one segment is a paraphrase of the other.

The SEPIA MT metric [31] is based on the dependence tree and is used to calculate the similarity of two text segments. It extends the tree to obtain the surface span, which is used as the main component of the similarity score. After the components of the tree are matched, a brevity penalty factor is suggested for deciding the difference in tree lengths for the two text segments.

Two other MT metrics commonly used in machine translation are the bilingual evaluation understudy (BLEU) metric [63] and the NIST metric [23] (an extension of the BLEU metric). Both also quantify similarity on the basis of matching words in the original text segment with words in the translated segment. Whereas the BLEU metric simply calculates the number of matching words, the NIST metric takes into account the importance of matching with different levels. The main drawback of these word matching metrics is that a word in a segment can match more than one word in the other segment.

Two MT metrics based on non-duplicate matching have been devised to overcome this problem. The METEOR metric [20] uses explicit ordering to identify matching tuples with minimized cross edges. However, it simply performs word-by-word matching. The maximum similarity (MAXSIM) metric [12] finds the maximum matching of unigram, bigram, and trigram words by using the Kuhn-Munkres algorithm. However, the maximum length of the phrase is a trigram. Moreover, the similarities of the phrases (unigram, bigram, and trigram) are disjointly combined. To overcome these drawbacks with the standard MT metrics, we have developed a heuristic method for finding the maximum of matching tuples up to the length of the text segments being compared. We also developed a metric for sophisticatedly quantifying the similarity on the basis of the matching tuples. The detail of our metric is explained in Chapter 5.

Chapter 3

Identification of private phrases

In this chapter, we develop an algorithm for identifying private messages in OSN messages on the basis of machine learning approach. Private phrases in such messages are then identified by using similarity metric, as described in Section 3.1. However, the machine learning approach does not work well for identifying private locational phrases. Therefore, we put forward a rule-based approach to overcome them, as presented in Section 3.2. The rule points out the relationship between locational phrases and user phrases for identifying the private phrases.

3.1 Identification of private phrases by similarity metric

3.1.1 Proposed method

Throughout this section, we use user blog t as an illustrative example: "My hometown is Tokyo. My favorite food is sushi. After graduating from Tokyo University, I studied at Harvard University for three years as a computer science major." Our proposed method for identifying private phrases in t has two steps: *identifying a private message* and *identifying private phrases*. The following subsections explain each step in detail.

Identifying a private message (Step 1)

Since messages posted in an OSN may include candidate phrases that do not disclose private information about the user, messages containing candidate phrases are classified as either private or non-private. *Private messages* disclose personal information about the user while *non-private messages* do not. If this step determines that the message is private, the

algorithm posts the same version of the message for all friends. Otherwise, the message is identified private phrases in Step 2.

In an evaluation (described in the next section), we found that sequential minimal optimization [66] is the best approach to creating classifier β used here. This classifier is used to determine whether input message t is a private or non-private message by using Eq. 3.1. In example input message t, the result of classification is $true$. This means that t is a private message. The algorithm thus uses t to identify the private phrases in the subsequent step.

$$flag = IsPrivateMessage(t) = true \tag{3.1}$$

Identifying private phrases (Step 2)

The data in a user's personal profile comprise seven main attribute types (hometown, education, work, religion, politics, sports, and personal interests) and are all noun phrases. Therefore, we use noun phrase chunking library[1] to extract noun phrases as much as possible from input message t. The noun phrases in t are "My hometown," "Tokyo," "My favorite food," "sushi," "Tokyo University," "I," "Harvard University," "three years," and "computer science major." All private phrases in input message t are identified by comparing each attribute a_i in the user's personal data profile A with each noun phrase in t. A co-occurrence metric is used to quantify each comparison [38].

The co-occurrence of two phrases X and Y ($Co(X,Y)$), Eq. 3.2 is the number of pages retrieved using a search engine from a huge dataset (such as Google or Wikipedia) containing both X and Y ($Fr(X \cap Y)$) divided by the number containing X or Y ($Fr(X \cup Y)$). We use Wikipedia for creating a search engine here. Two example co-occurrence metrics are $Co(\text{Shinjuku}, \text{Tokyo}) = \frac{1{,}578}{60{,}084} = 0.0263$ and $Co(\text{Shinjuku}, \text{Harvard University}) = \frac{28}{40{,}219} = 0.0007$. The result of co-occurrence analysis reveals that "Tokyo" is more similar to "Shinjuku" than "Harvard University."

$$Co(X,Y) = \frac{Fr(X \cap Y)}{Fr(X \cup Y)} \tag{3.2}$$

Table 3.1 shows the results of identification. If the value of the co-occurrence metric is greater than the threshold α, the phrase is considered private. On the basis of the 1,589 different private phrases described in the evaluation section, we set α to 0.0169. By using

[1] http://www.dcs.shef.ac.uk/~mark/phd/software/chunker.html

User profile A		**Noun phrases in** t	$Co(A,B)$
a_0 =Full name	Adam Ebert	My hometown	$0 < \alpha (= 0.0169)$
a_1 =Work	Student	...	...
a_2 =University	*Harvard*	*Harvard University*	$0.3550 > \alpha$
...	...	...	...
a_i =Prefecture	*Shinjuku*	*Tokyo*	$0.0263 > \alpha$
...	...	...	...

Table 3.1 Private phrase identification.

α, we identify two private phrases, p_0="Harvard University" and p_1="Tokyo," as shown in Eq. 3.3. In this example, even if a friend changes a private phrase directly by using a synonym phrase like "Harvard University" or indirectly by using a similar phrase like "Tokyo," the algorithm identifies the modified phrase.

$$P = IdentifyPrivatePhrases(t,A) = \{p_0, p_1\} = \{\text{"Harvard University"}, \text{"Tokyo"}\} \quad (3.3)$$

3.1.2 Evaluation

Distinguishing between private and non-private messages

Messages containing private phrases were identified by comparing each phrase in the normalized tweets with certain phrases in a corpus of cities for hometown[2], universities, and colleges for education[3], careers[4], sports[5], religions[6], politics[7], and interests[8]. Co-occurrence threshold α was used to quantify each comparison. Finally, we extracted 137,628 tweets containing private phrases as a dataset for evaluation.

We manually labeled 3,000 random tweets and ran 11 algorithms with 10-fold cross validation, as shown in Table 3.2. The 11 algorithms were combined with features extracted from 4 models to create classifiers. The 11 algorithms were support vector machine (SVM), multinomial logistic regression (Logistic), K-nearest neighbors (IBk), multinomial Naive Bayes (NaiveBayesMulti), an ensemble of randomizable base classifiers (RandomCommittee), One R, AdaBoost M1, Naive Bayes, Repeated Incremental Pruning to Produce Error Reduction

[2]http://www.maxmind.com/en/worldcities
[3]http://www.odditysoftware.com/page-datasales161.htm
[4]http://www.careerdirections.ie/ListJobs.aspx
[5]http://listofsports.com/
[6]http://www.guavastudios.com/religion-list.htm
[7]http://www.gksoft.com/govt/en/parties.html
[8]http://www.hobby-hour.com/hobby_list.php

(JRip), SVM + Logistic Regression + Linear Regression (SGD), and sequential minimal optimization (SMO). The features were extracted using 4 models (1-gram, 1-gram+2-gram, 1-gram+2-gram+3-gram, and 1-gram+2-gram+3-gram+4-gram). Our method uses one algorithm with one model to create classifier β. Therefore, sequential minimal optimization with the (1-gram + 2-gram + 3-gram) model is the optimal algorithm for creating classifier β ($F1 = 92\%$). β was used to extract 54,621 private tweets from the 137,628 ones used for estimating the threshold α of the co-occurrence metric in the next subsection.

Algorithm	1-gram			1-gram+2-gram			1-gram+···+3-gram			1-gram+···+4-gram		
	Pre	**Rec**	$F1$	**Pre**	**Rec**	$F1$	**Pre**	**Rec**	$F1$	**Pre**	**Rec**	$F1$
Support vector machine (SVM)	**63.3%**	**63.2%**	**63.1%**	56.3%	55.8%	55.0%	63.3%	63.2%	63.1%	62.6%	62.4%	62.3%
Multinomial logistic regression (Logistic)	**79.5%**	**79.2%**	**79.2%**	72.1%	70.5%	69.9%	78.1%	67.5%	64.1%	76.7%	74.6%	74.0%
K-nearest neighbors (IBk)	**82.9%**	**80.6%**	**80.2%**	72.3%	71.0%	71.8%	81.4%	79.1%	78.8%	74.3%	51.0%	37.5%
Multinomial Naive Bayes	**82.3%**	**80.2%**	**79.9%**	74.5%	72.0%	71.5%	79.5%	79.2%	79.2%	79.1%	76.0%	76.5%
An ensemble of randomizable base classifiers	82.5%	82.1%	82.1%	70.3%	68.0%	67.9%	**82.3%**	**82.3%**	**82.3%**	77.5%	76.0%	76.0%
One R	**85.1%**	**83.4%**	**83.2%**	77.8%	62.0%	57.1%	85.1%	83.4%	83.2%	84.3%	82.0%	81.9%
AdaBoost M1	**88.1%**	**87.4%**	**87.3%**	77.8%	62.0%	57.1%	88.1%	87.4%	87.3%	86.8%	85.0%	85.8%
Naive Bayes	87.6%	87.5%	87.5%	76.5%	72.0%	71.5%	**88.3%**	**88.3%**	**88.3%**	86.2%	86.0%	86.1%
Repeated Increment. Pruning to Produce Error Reduct. (JRip)	91.9%	91.7%	91.7%	77.9%	72.0%	70.9%	**91.9%**	**91.9%**	**91.9%**	89.8%	89.0%	89.7%
SVM+ Logistic Regress.+ Linear Regression (SGD)	90.2%	90.2%	90.2%	81.6%	81.0%	81.6%	**91.9%**	**91.9%**	**91.9%**	87.4%	87.0%	87.3%
Sequent. minimal optimizat. (SMO)	90.7%	90.7%	90.7%	81.5%	81.0%	81.4%	**92.0%**	**92.0%**	**92.0%**	85.4%	85.0%	85.4%

Table 3.2 Classifier creation results.

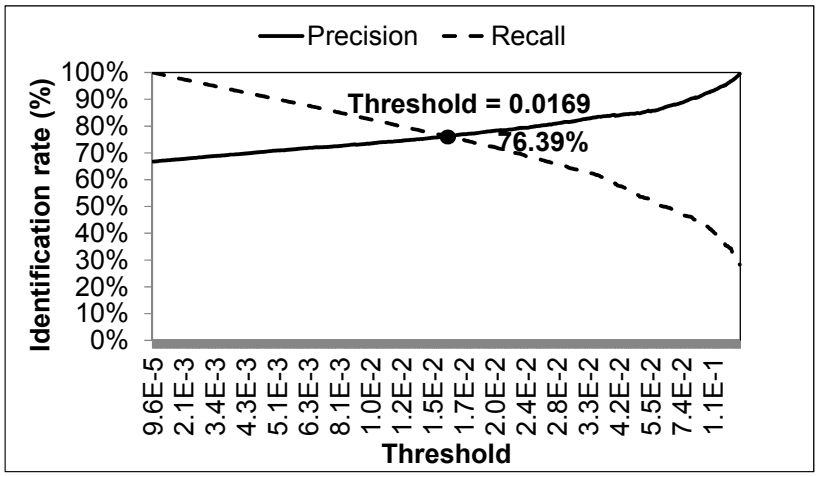


Fig. 3.1 Co-occurrence threshold.

The best performances of the algorithms are shown in bold in Table 3.2. We did not experiment with the (1-gram+$\cdots$+5-gram) model because the (1-gram+$\cdots$+4-gram) model did not find any better solutions. Moreover, some algorithms with the (1-gram+$\cdots$+5-gram) model took a long time to create classifiers. For example, the multinomial logistic regression (Logistic) algorithm with the (1-gram+$\cdots$+4-gram) model took 3.3 hours when it was run on a computer with an Intel Xeon e5-2690 32Core Processor CPU 2.9GHz , and 250GB RAM, and it did not completely run with the (1-gram+$\cdots$+5-gram) model.

Estimating threshold for co-occurrence metric of private phrases identification

We used a name entity recognize algorithm [46] to extract 1589 different locations in the private tweets. The locations were compared with a list of countries[9] to find the best matches using the co-occurrence metric. The precision and recall for the 1589 locations are plotted in Fig. 3.1. We used 0.0169 as the co-occurrence threshold α in order to balance precision with recall (precision=recall=76.39%).

The co-occurrence metric works well for identifying private phrases in general attributes but it is unsuitable for specific one such as location. Therefore, we put forward a rule-based approach that identify efficiently private locational phrases in an OSN message.

[9]http://www.internetworldstats.com/list2.htm

3.2 Identification of private locational phrases by rule-based approach

We propose a rule-based algorithm for identifying private locational phrases in an OSN message t. Figure 3.2 illustrates the process of our algorithm containing seven main steps described below:

3.2.1 Proposed algorithm

- *Step 1 (detecting language)*: An OSN message t is determined whether or not it is written in English.

- *Step 2 (normalizing)*: Misspelling words of the English message t are corrected by using normalization.

- *Step 3 (extracting candidate phrases)*: The normalized message t is analyzed for extracting candidate phrases $P = \{p_i\}$. A candidate p_i is defined as a phrase that begins with *the word "I"* and ends with *a location* l_i. Each candidate p_i is evaluated with three rules in Step 4, Step 5, and Step 6 for identifying private locational phrases accumulated in a set of *privatePhrases L*.

- *Step 4 (checking negative phrase)*: p_i is checked whether it includes negative words (such as not, no, and never). If none of the negative words is found, the phrase p_i is analyzed with other criteria in next steps. Otherwise, p_i does not contain any private phrases and the algorithm examines the next candidate phrases by increasing counter i.

- *Step 5 (checking non-private locational verbs)*: This step checks whether p_i contains at least one non-private locational verb. If p_i does not include any non-private locational verbs, it is tested with the last condition in the next step.

- *Step 6 (checking private locational verbs)*: The candidate phrase p_i is analyzed whether it contains one or more private locational verbs. It means that p_i contains a private locational phrase l_i via message t. Therefore, l_i is added to the *privatePhrases L*. The algorithm then sends a notification of the private phrases L to the user in the final step (Step 7).

- *Step 7 (notifying)*: Our algorithm notifies about private phrases L to owner user whenever L is not $null$.

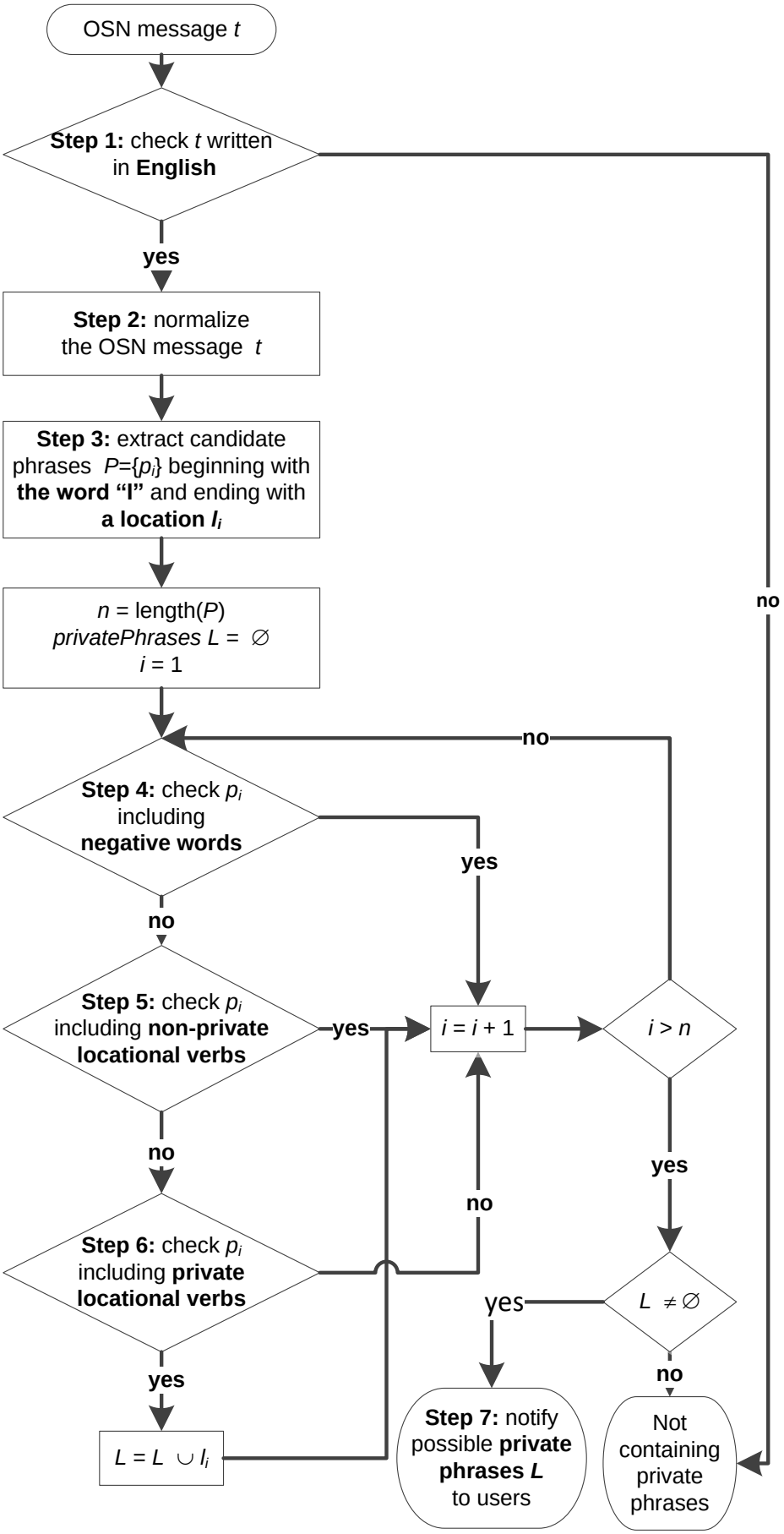


Fig. 3.2 Process of the rule-based algorithm for identifying private locational phrases of an OSN message.

Details of first six main steps are presented with three example messages in below. These messages are crawled from Tweet2011 dataset [61].

t_1: "How manyy time I'll tell that I'm not froom California?"

t_2: "I raelly want to fly out to Los Angeles and meet all the amazing people/artists out there."

t_3: "I live in Seattle, do you knoww what station is showing your new show?"

Detecting language (Step 1)

OSN message t is determined where it is written by English or not by language detection. The language detection library [74] created by Shuyo is used in here. If the detected language is English then the message t is normalized in the next step.

Normalizing (Step 2)

The English message t is normalized (e.g., words are converted to correct misspelling words) with lexical normalization created by Han et al. [33]. For three examples used here, normalized texts are listed below with normalized words in italic:

t_1: "How *many* time I'll tell that I'm not *from* California?"

t_2: "I *really* want to fly out to Los Angeles and meet all the amazing people/artists out there."

t_3: "I live in Seattle, do you *know* what station is showing your new show?"

Extracting candidate phrases (Step 3)

Firstly, a name entity recognition library is used for identifying locational phrases in message t. This library used here is created by Ritter et al. for identifying locational phrases in tweets [68]. In three example messages used here, a location "*California*" is identified for t_1, "*Los Angeles*" for t_2, and "*Seattle*" for t_3. Finally, all candidate phrases $P = \{p_i\}$ is extracted from t. Each candidate p_i is a phrase which begins with the word "I" and ends with a location. In message t_1, two candidate phrases p_1 and p_2 ,which begins with the "*I*" and ends with the location ("*California*"), are extracted below:

p_1:"*I*'ll tell that I'm not from *California*"

p_2:"*I*'m not from *California*"

Checking negative phrases (Step 4)

In this step, each separate words in p_i is converted into their lemmas by using Stanford CoreNLP library [50]. The lemmas are then compared with negative words (not, never,

Type of verb	Explanation	Example
Emotion verb	verbs of feeling	*like*, *love*, *want*
Perception verb	verbs of seeing, hearing, feeling	*hear*, *feel*
Body verb	verbs of grooming, dressing and bodily care	*wear*
Cognition verb	verbs of thinking, judging, analyzing, doubting	*think*
Communication verb	verbs of telling, asking, ordering, singing	*ask*
Contract verb	verbs of touching, hitting, tying, digging	*touch*
Creation verb	verbs of sewing, baking, painting, performing	*paint*
Social verb	verbs of political and social activities and events	*make*
Possession verb	verbs of buying, selling, owning	*buy*

Table 3.3 Types of non-private locational verbs.

and no) for determining where p_i contains at least one negative word or not. For example, message t_1 includes two candidate phrases p_1:"I tell that I'm *not* from California" and p_2:"I'm *not* from California." Each phrase contains a negative word "*not*." Therefore, they are ignored for identifying private locational phrases. Two remaining messages t_2 and t_3 do not occur any negative words. They are thus analyzed private phrases in next steps.

Checking non-private locational verbs (Step 5)

A parser tool of the Stanford CoreNLP library is used to extract verbs in each candidate phrase p_i. Using WordNet library [53], these verbs are divided into eleven types including emotion, perception, body, cognition, communication, contract, creation, social, possession, stative, and motion verb. Nine types of them are chosen as non-private locational verbs listed in Table 3.3. For example of a candidate phrase "I *want* to fly out to Los Angeles" in t_2, it contains a non-private locational verb "*want*" as an *emotion verb*. Therefore, this phrase does not reveal user's locations. t_3 does not include any of the non-private locational verbs. It is thus checked with a final criterion in Step 6.

Checking private locational verbs (Step 6)

We use two remaining types of verbs from the total eleven ones as private locational verbs listed in Table 3.4. For example of a candidate phrase "I *live* in *Seattle*" in t_3, it contains a private locational verb "*live*" as a *stative verb*. Therefore, t_3 reveals user's location ("*Seattle*") and the algorithm thus notifies the identified locational phrases to user in the final step (Step 7).

Type of verb	Explanation	Example
Stative verb	verbs of being, having, spatial relations	*live*, *stay*
Motion verb	verbs of walking, flying, swimming	*fly*

Table 3.4 Types of private locational verbs.

3.2.2 Evaluation

Accuracy metric

We crawled about 16 million tweets from Tweets2011 dataset [61] for evaluating our method. These tweets are filtered using a four-step process as follows:

- *Step 1 (detecting language)*: The 16 million tweets are analyzed to extract English tweets by a language detection tool [74].
- *Step 2 (normalizing)*: The English tweet messages, an informal language, contain several misspelling words (e.g., 2morrow, g0, w/o). These words are corrected using lexical normalization on the basis of a normalization dictionary created by Han et al. [33].
- *Step 3 (detecting user tweets)*: The normalized tweets are chosen as user tweets if they contain one or more *words* "*I*."
- *Step 4 (identifying location)*: The user tweets containing locational phrases are extracted on the basis of a name entity recognition (NER) tool [68].

We manually labeled 2817 random user tweets into 1150 tweets as private locational tweets and the remaining ones as non-private locational tweets. These labeled tweets are evaluated with five approaches: a baseline by using machine learning, three machine learning extensions, and our approach. In the machine learning approach and its extensions, the labeled tweets are extracted features using N-gram model where N is equal three. The N-gram model is chosen for extracting features because it is claimed as the best model for taking advantage of the relationship between words in a message [37]. We used these features for creating a classifier with different main machine learning algorithms such as support vector machine (SVM), logistic, Naive Bayes, and decision trees. The best classifier with SVM algorithm archives the highest accuracy among the machine learning approach and its extensions shown in Table 3.5 since they are evaluated with 10-fold cross validation. These approaches are described in detail below:

Approach	Accuracy
Words approach	79.30%
Filtered words approach	74.51%
Type of words approach	80.87%
Type of filtered words approach	81.97%
Our approach	**84.95**%

Table 3.5 Accuracy of a machine learning approach, three its extensions, and the rule-based approach.

- *Words approach*: Each message from the 2817 labeled tweets is analyzed for extracting into separate words. These words are used for creating the classifier. 79.30% of the labeled tweets are correctly classified using this approach.
- *Filtered words approach*: *Words approach* is extended by filtered words of each labeled tweet. Filtered words are extracted from the shortest phrase which begins with the word *"I"* and ends with a location. The set of filtered words probably equals *null* since messages contain locational phrases before the word *"I."* The accuracy of this extension (74.51%) is lower than of *words approach* using word features (79.30%). The result shows that the importance of words is not filtered for creating the classifier.
- *Type of words approach*: This approach uses types of each word in messages for creating the classifier. The types of each word are determined by WordNet library [53]. The accuracy of this approach (80.87%) is significantly higher than two previous approach using words features. It points out the impact of using type of words for identifying private locational phrases.
- *Type of filtered words approach*: The final extension combines two previous extensions including *filtered words approach* and *type of words approach*. Firstly, filter words are extracted in a manner similar to the *filtered words approach*. Finally, only type of filtered words are used to creating features. This extension achieves the highest accuracy (81.97%) in the base line (*word approach*) and other extensions. The result illuminates that type of filtered words are significant features for identifying private locational phrases of users.

Our rule-based approach correctly identifies 84.95% of the labeled tweets. This is significantly better than the 81.97% rate of the best machine learning's extension approach (*type of filtered words approach*). This demonstrates the efficiency of our rule for identifying private locational phrases of OSN messages.

		Predicted dataset	
		Non-private	**Private**
Labeled dataset	**Non-private**	1658	9
	Private	574	576

Table 3.6 Confusion matrix of a machine learning approach (*words approach*).

Precision	**Recall**	**F-measure**	**Class**
74.3%	99.5%	85.0%	Non-private
98.5%	**50.1**%	66.4%	Private

Table 3.7 Precision, recall, and F-measure metrics of a machine learning approach (*words approach*).

Precision, recall, and F-measure metrics

This section analyzes a machine learning approach (*words approach*) and our approach in more detail with precision, recall and F-measure metrics. For calculating these metrics, a confusion matrix of the machine learning approach is created from the 2817 labeled tweets above and shown in Table 3.6. The last two rows describe private class and non-private class for labeled dataset; and the last two columns represent these classes for predicted dataset.

As can be seen from Table 3.6, 99.64% (1658/1667) labeled tweets of non-private class are correctly classified. However, the tweets of private class are almost randomly distributed for each class of predicted dataset (575 tweets assigned for private and 576 ones for non-private). It obviously illustrates the imbalance of the machine learning approach for identifying private locational phrases.

Precision, recall and F-measure metrics are calculated for each class presented in Table 3.7 based on the confusion matrix in Table 3.6. The imbalance of the machine learning approach is pointed out again. Precision metric is almost perfect (98.5%) for private class but the recall metric is very low (50.1%).

Table 3.8 describes a confusion matrix of our rule-based approach. Precision, recall, and F-measure metrics are quantified in Table 3.9. Similar values of these metrics demonstrate more balance of our approach than the machine learning approach.

3.2.3 Analysis

The strength of our approach not only works well for OSN messages but also applies for other areas (such as email, news, and military) because these messages are similar structure

		Predicted dataset	
		Non-private	**Private**
Labeled dataset	**Non-private**	1470	197
	Private	227	923

Table 3.8 Confusion matrix of the rule-based approach.

Precision	**Recall**	**F-measure**	**Class**
86.6%	88.2%	87.4%	Non-private
82.4%	80.3%	81.3%	Private

Table 3.9 Precision, recall, and F-measure metrics of the rule-based approach.

with OSN messages. Our approach can also be used for other entities (personal name, diseases, etc.) by simple changing our rule. For example, word "I" in our rule is replaced by a personal name for identifying private locational phrases of him or her. The rule also is easily changed for applying other languages based on their grammars.

In experiment with the 2817 labeled tweets above, 84.95% of them are correctly identified private locational phrases by our rule-based approach. The remaining 424 error tweets (15.05%) are analyzed for finding limits of the rule. We categorize these errors into four types: errors by indirect inference, errors by backward inference, errors by tools, and errors by others. Figure 3.3 summarizes ratios of each type of errors.

- *Errors by indirect inference*: Our rule of identifying private locational phrases is based on candidate phrases that begin with the word *"I"* and end with a location. However, a few private locational phrases are indirectly inferred by other phrases of messages. For example of a tweet: "I *wish* I could teleport to *South Beach*.", a phrases locational phrase "*South Beach*" is indirectly indicated by a verb "*wish*." This type of errors

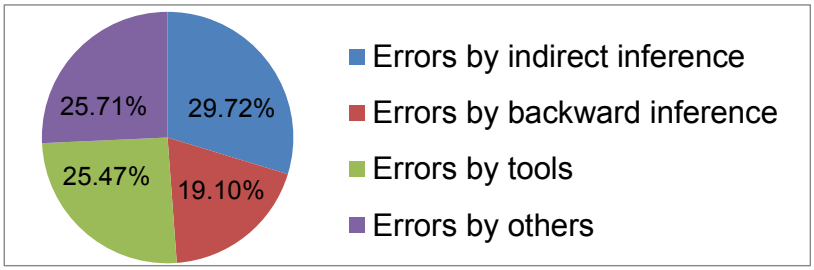


Fig. 3.3 Analysis of errors for the rule-based approach.

occupies the largest ratio (29.72%) in total errors. In order to overcome such errors, the rule must be extended to analyze outside contexts of the candidate phrases.

- *Errors by backward inference*: Messages of these errors mention a location before the word *"I."* For example, a private locational phrase "*ATLANTA*" is identified in a tweet: "When you come *ATLANTA*, *I*'ll make sure you meet all of them." The type of these error need a sophisticated algorithm for pointing out the complex semantic of messages.

- *Errors by tools*: These errors are made from three tools used in the proposed process of our rule. They include a name entity recognition (NER) tool for identifying locational phrases, a normalization tool for correcting the user's misspelling words, and a WordNet tool for identifying type of verbs. For example of a message "I'm Miss American Dream since I was 17 Don't matter if *I* step on the scene Or sneak away to the *Philippines*", it demonstrates a wrong location identification "*Philippine*." due to the NER tool. More factors should be added into the rule to decreasing the impact of these tools.

- *Errors by others*: Other errors are made by wrong identifying relationship of user and location such as a message: "*I* Love Lucy is on again, and *they*'re still on their way to *California*." A location "*California*" belongs to a pronoun "*they*" but it does not relate to a user "*I*." Another case of these errors is derived from possession of locations (such as: "*I* need to go to Bank of *America*!"). We plan to extend the annotated corpus to determine the major errors of this type and add more criteria for the rule to cut down these errors.

3.2.4 Limitation and future work

Our proposed rule does not cover all cases of private locational phrases. Few messages contain some pronouns (such as "we," "me," and "us"), they probably contain private locational phrases such as a tweet: "*We* in *Brazil* we cannot wait to watch your movie, we are very anxious!" However, these cases rarely occur in OSNs. We randomly analyzed 100 similar messages from Tweets2011 dataset [61], but only five tweets contain private locational phrases.

Each location describes different levels of privacy such as "United States," "California," and "Stanford." The first location information is safer than the others. In future work, we will quantify the privacy level of locations by using distribution metric [59] and estimate a threshold for private locational phrases. Our rule will also be extended for other private

phrases of user (such as illness, interests, and hometown). Finally, the rule will be used as a plug-in of an existing OSN (such as Facebook, Twitter, and Google+) for enhancing user's privacy.

3.3 Summary

This chapter presents a method for identifying private phrases in messages to be posted in OSNs. We identify a classifier to determine whether a composed message is either a private message or a non-private message to ensure that only private message are identified private phrases. A co-occurrence metric is used to identify private phrases even if an attacker directly or indirectly changes the fingerprints.

We also propose a rule-based algorithm for identifying private locational phrases in natural language messages to be posted in OSNs. The rule is created on the basis of relationships between negative phrases, type of verb phrases, user's information phrases, and locational phrases in each message. The accuracy was 84.95% for 2817 messages, showing that this approach efficiently identifies private locational phrases of users in OSNs. It is significantly better than 70.30% of a machine learning approach and its extensions with (best accuracy 81.97%). Moreover, the balance of our approach is proved via precision, recall, and F-measure metrics.

Private information about a user in OSN message is generally conveyed in private phrases. The problem is that such information could be used by commercial companies to make annoy for the owner user. Therefore, the private phrases are anonymized in Chapter 4 to overcome the issues.

Chapter 4

Anonymization of private phrases

In this chapter, on the basis of private phrases identified in Chapter 3. We develop an algorithm for anonymizing the private ones by using generalization, as described in Section. 4.1. The generalizations of private phrases are created by using a semantic dictionary (WordNet). However, the semantic dictionary can only be used to apply a word or a simple phrase having a certain meaning. They cannot be used to find generalizations for temporal phrases (such as "at 9a.m.," "in 25 minutes," "for a long time"). Therefore, we propose a method for extracting temporal patterns from OSN corpus. Such patterns are used for anonymizing temporal phrases by generalization as described in Section. 4.2 or by suppression as mentioned in Section. 4.3.

4.1 Anonymization of private phrases by generalization

4.1.1 Proposed method

Throughout this section, we use user blog t as an illustrative example: "My hometown is Tokyo. My favorite food is sushi. After graduating from Tokyo University, I studied at Harvard University for three years as a computer science major." Our proposed method for anonymizing private phrases in t has three steps: *creating generalization schemas*, *quantifying information loss of generalizations*, and *creating fingerprints*. The following subsections explain each step in detail.

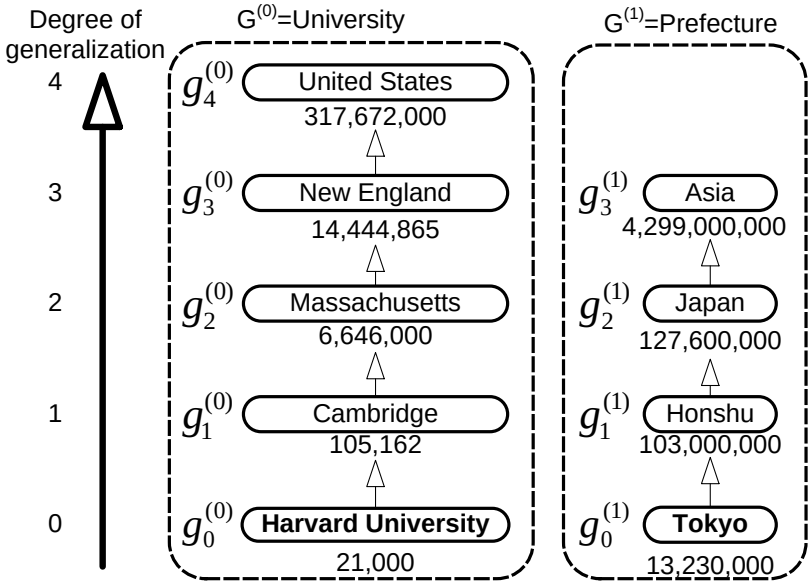


Fig. 4.1 Generalization schemas for two quasi-identifiers.

Creating generalization schemas (Step 1)

All private phrases in input message t are identified by using co-occurrence metric ($\alpha = 0.0169$), which is presented in Chapter 3. For this example, we identify two private phrases p_0="Harvard University" and p_1="Tokyo," comprise set P shown in Eq. 4.1.

$$P = IdentifyPrivatePhrases(t,A) = \{p_0, p_1\} = \{\text{"Harvard University"}, \text{"Tokyo"}\} \quad (4.1)$$

Our algorithm creates generalizations for each private phrase in P by using generalization schemas (Eq. 4.2). The generalization-related information in the WordNet lexical database [53] are used to create these schemas. For example, according to WordNet, "Cambridge" is a direct generalization of "Harvard University." Fig. 4.1 shows the results of generalization for two private phrases: $G^{(0)}$ ="University" and $G^{(1)}$ ="Prefecture."

$$G = CreateGeneralizationSchemas(P) = \{g_j^{(i)}\} \quad (4.2)$$

From the about 16 million tweets in the TREC Tweets2011 dataset [61], we extracted 75,464 private phrases exceeding co-occurrence threshold α. Of these private phrases, 98.47% are covered by WordNet. This shows that WordNet covers most cases of private phrases. For the few remaining cases, we use the private phrase as the sole level of generalization.

Generalization g	**Samarati**	**Precision**	**Distribution**
{Harvard University, Tokyo}	0	0.00	1.25
{Cambridge, Tokyo}	1	0.25	1.33
{Harvard University, Honshu}	1	0.33	1.34
{Harvard University, Japan}	2	0.67	1.35
{Cambridge, Honshu}	2	0.58	1.42
{Cambridge, Japan}	3	0.92	1.43
{Harvard University, Asia}	3	1.00	1.51
...	...	...	...

Table 4.1 Quantify possible generalizations.

Quantifying information loss of generalizations (Step 2)

As mentioned, since anonymization by generalizing results in information loss, this function was used to quantify the loss and thus ensure that each group of friends receives a version of the message with an appropriate degree of anonymization. To quantify information loss by using the distribution metric (*Dis*):

$$Dis(P) = \sum_{i=0}^{N-1} \frac{log(|p_i|)}{log(|P_i|)}, \tag{4.3}$$

where $|p_i|$ and $|P_i|$ is the population of private phrase p_i and P_i, correspondingly. We extract the population of generalizations shown in Fig. 4.1 from the $HasPopulation$ attributes of YAGO [80]. YAGO uses infoboxes of pages in Wikipedia to create $HasPopulation$ attributes. For example, the population of "Tokyo" is approximately 13,230,000. We use logarithmic scaling in this metric to reduce the effect of the huge population of each generalization. Dividing for the highest level generalization maintains the balance between private phrases. The *Dis* for the generalizations of two private phrases in P is described in Eq. 4.4. Table 4.1 shows the values of the precision metric, Samarati metric, and distribution metric for all possible generalizations of schemas G.

$$\begin{aligned} Dis(P) = \sum_{i=0}^{N-1} \frac{log(|p_i|)}{log(|P_i|)} = Dis(\text{Harvard University}) + Dis(\text{Tokyo}) = \\ = \frac{log(21,000)}{log(317,672,000)} + \frac{log(13,230,000)}{log(4,299,000,000)} = 1.25 \end{aligned} \tag{4.4}$$

However, some generalizations do not have the $HasPopulation$ attribute. Therefore, we develop an information loss metric ($InfoLoss$) for N generalizations $g = \{g_i\}$.

In this metric, shown in Eq. 4.5, the distribution metric is used if the generalization has the $HasPopulation$ attribute. Otherwise, the precision metric is used. The $InfoLoss$ metric for the generalizations of two private phrases in P is described in Eq. 4.6. A special case of the $InfoLoss$ metric is presented in Section 4.1.3.

$$InfoLoss(g) = \sum_{i=0}^{N-1} InfoLoss(g_i) \tag{4.5}$$

$$InfoLoss(g_i) = \begin{cases} Dis(g_i) & \text{if } g_i \text{ has } HasPopulation \text{ attribute} \\ Pre(g_i) & \text{otherwise} \end{cases}$$

$$InfoLoss(P) = Dis(\text{Harvard University}) + Dis(\text{Tokyo}) = 1.25 \tag{4.6}$$

Creating fingerprints (Step 3)

This step creates the fingerprints used to identify a friend who has disclosed personal information. Simply replacing a phrase p in input message t to create a fingerprint can make the message unnatural. The naturalness of fingerprinted messages is improved by using the frequency score (Fre) defined in Eq. 4.7. The position of phrase p in t is denoted as p_{index}. The $subphrase(t, pos, n)$ function retrieves a sub-phrase from message t; the sub-phrase starts at position pos and has n words. The fr is the n-gram frequency count from the Google Web 1T 5-gram corpus[1] (we consider $ln0$ to be equal to 0). This score is an extension of the substitution metric proposed by Change et al. [15]. Evaluation using 1113 OSN message revealed that the threshold γ when using this score is 80.55. Use of this value ensures that our algorithm creates a sufficient number of fingerprints for friends and that the fingerprinted messages are natural. An example of replacing the word "three" in input message t with a numerical "3" is shown in Table 4.2.

$$Fre(p,t) = \sum_{n=2}^{5} \sum_{pos=p_{index}-n+1}^{p_{index}} ln(fr(subphrase(t, pos, n))) \tag{4.7}$$

[1] http://catalog.ldc.upenn.edu/LDC2006T13

n	*subphrase s*	**Frequency**
2	for **3**	$fr(s) = ln(4,277,726) = 15.27$
	3 years	$fr(s) = ln(6,631,904) = 15.71$
3	University for **3**	$fr(s) = ln(787) = 6.67$
	for **3** years	$fr(s) = ln(472,665) = 13.07$
	3 years as	$fr(s) = ln(26,564) = 10.19$
4	Harvard University for **3**	$fr(s) = ln(0) = 0$
	University for **3** years	$fr(s) = ln(444) = 6.10$
	for **3** years as	$fr(s) = ln(4,561) = 8.43$
	3 years as a	$fr(s) = ln(9,135) = 9.12$
5	at Harvard University for **3**	$fr(s) = ln(0) = 0$
	Harvard University for **3** years	$fr(s) = ln(0) = 0$
	University for **3** years as	$fr(s) = ln(0) = 0$
	for **3** years as a	$fr(s) = ln(1,634) = 7.40$
	3 years as a computer	$fr(s) = ln(0) = 0$
$Fre(p,t) = Fre(\text{“}\mathbf{3}\text{”},t) = 91.94 > \gamma(= 80.55)$		

Table 4.2 Check naturalness of message though replacement.

Generalization g	$InfoLoss$	**Group**	$Fre(g,t) = min\{Fre(g_i,t)\}$
{Harvard University, Tokyo}	1.25	Families	$80.90 > \gamma(= 80.55)$
{Cambridge, Tokyo}	1.33	Colleagues	$82.15 > \gamma$
{Harvard University, Honshu}	1.34	Best Friends	$80.61 > \gamma$
{Harvard University, Japan}	1.35	Friends	$90.80 > \gamma$
~~{Cambridge, Honshu}~~	~~1.42~~		~~$76.25 < \gamma$~~
{Cambridge, Japan}	**1.43**	**Acquaintances**	**87.54**$> \gamma$
...	...	...	...

Table 4.3 Assign generalizations for each group.

The appropriate degrees of generalization are shown in Table 4.3. All possible combinations of generalizations are sorted by the distribution metric so that the degree of anonymization increases from top to bottom. All unsuitable generalizations are eliminated on the basis of the frequency threshold γ used for naturalness checking. The remaining generalizations are used for groups with proper levels. We use Shimon's method [47] to determine the proper levels of groups.

Each generalization is used to replace corresponding private phrases in t of each group. A message is modified using synonyms to create a fingerprint for each friend in a group. The synonyms are obtained using WordNet. The best fingerprints are checked by using frequency threshold γ. For example, our algorithm suggested the following fingerprint for *"Friend 1"* in the "*Best Friends*" group. "My hometown is *Honshu*. My favorite food is sushi. After

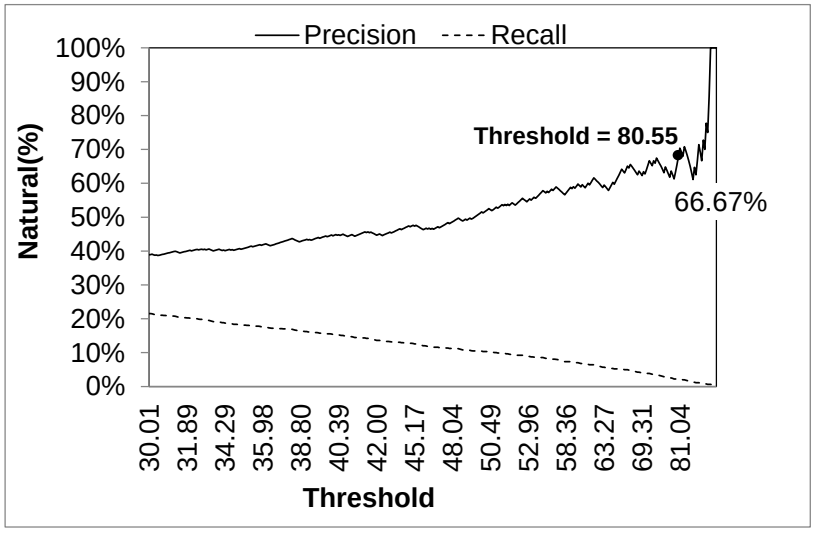


Fig. 4.2 Frequency threshold.

graduating from Tokyo University, I studied at *Harvard University* for *3* years as a computer science major."

4.1.2 Evaluation

Estimating threshold for frequency metric for naturalness checking

Using the frequency metric improved the naturalness of the fingerprinted messages. We created many fingerprinted versions of the 54,621 personal tweets in Chapter 3 by using generalizations for the private phrases and synonyms for the other phrases. We randomly selected 1113 fingerprinted versions and manually labeled them as either natural or unnatural. The results are plotted in Fig. 4.2. We chose a threshold γ of 80.55 for balancing between creating natural versions of the message and creating a sufficient number of generalizations fingerprints. With this value, 66.67% of the fingerprinted messages were natural, and an average 16.59 generalizations and 140.91 fingerprints were created. The following subsections describe these results in detail.

Number of possible generalizations for groups

The number of generalizations $\bar{T}$ was calculated using

$$\bar{T} = \prod_{i=0}^{N-1} |T_i|, \tag{4.8}$$

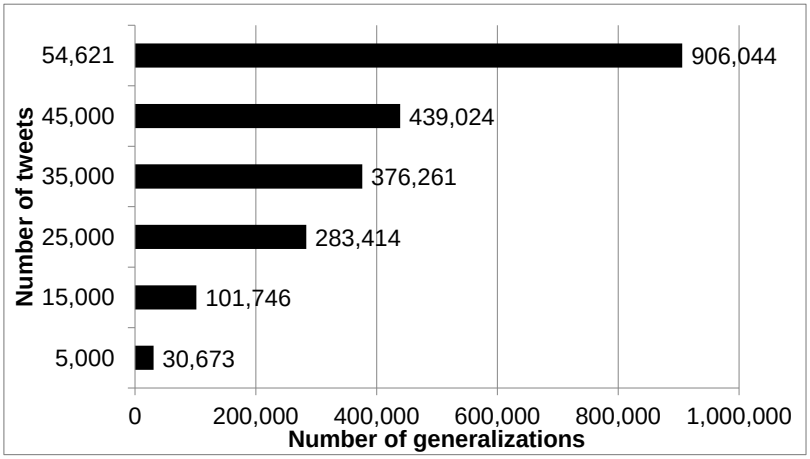


Fig. 4.3 Number of possible generalizations for groups.

where N is the total number of private phrases, and $|T_i|$ is the number of generalizations of private phrase i-th.

The number of possible generalizations for groups T is the number of generalizations in $\bar{T}$ exceeding frequency threshold γ used for checking the naturalness.

The total number of possible generalizations is shown in Fig. 4.3. We created 906,004 generalizations from the 54,621 personal tweets, an average of 16.59 generalizations per tweet. These results show that our algorithm can create a sufficient number of generalizations for both the default groups (*Families*, *Friends*, *Public*) and many other groups created by the user.

In Fig. 4.3, many tweets in 9,612 tweets (from 45,000 to 54,621) are long diary blogs. Each blog contains more than seven private phrases. Moreover, some private phrases are repeated several times in the blog. Each blog creates more than 2,000 generalizations by using our algorithm and is retweeted a few times. Therefore, these 9,612 tweets create more generalizations than other ones.

Number of possible fingerprints for friends

The number of fingerprints $\bar{F}$ depends on the number of synonyms $|S_j|$ of the j-th private phrase in each i-th generalized message; n_i is the length of the i-th generalization.

$$\bar{F} = \sum_{i=0}^{T-1} \prod_{j=0}^{n_i-1} |S_j| \tag{4.9}$$

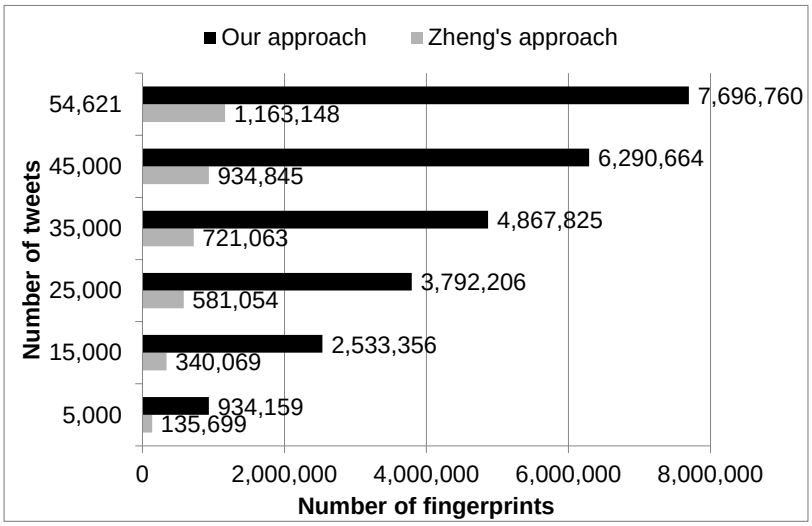


Fig. 4.4 Number of possible fingerprints for friends.

The number of possible fingerprints for friends F is the number of fingerprints in $\bar{F}$ exceeding frequency threshold γ used for checking the naturalness.

Using the 54,621 personal tweets, we calculated the number of possible fingerprints using two approaches. Message naturalness was checked using frequency threshold γ. The first approach was to use synonyms generated by Zheng's algorithm [90] to create fingerprints. The second was to use generalizations for private phrases and synonyms for other phrases (our algorithm).

As shown in Fig. 4.4, the Zheng algorithm approach created an average of 21.29 fingerprints per tweet while our approach created an average of 140.91. On Facebook, the average number of friends per user is 130[2]. Therefore, our approach creates a sufficient number of fingerprints for the average Facebook user.

4.1.3 Discussion

Strength of fingerprints

A frequency score is used for checking the naturalness of the replacements so that they do not catch the attention of attackers and thereby preventing them from transforming the fingerprints.

Although attackers might change a private phrase to avoid disclosure detection, the use of the co-occurrence metric to identify private phrases thwarts their efforts. The co-occurrence

[2]http://www.statisticbrain.com/facebook-statistics/

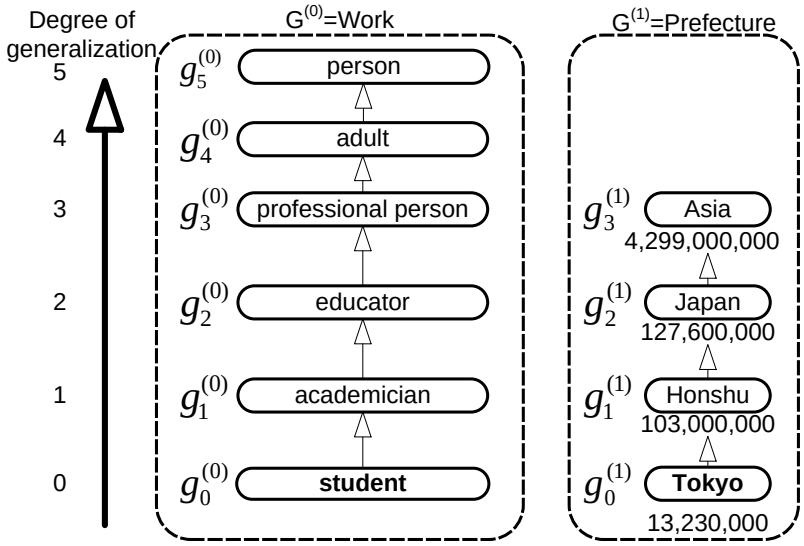


Fig. 4.5 Generalizations for two private phrases.

metric directly identifies the exact private phrase if the attacker uses a synonym or indirectly identifies it if the attacker uses a similar phrase.

Limitation

In our algorithm, the distribution metric supports quantifying information loss for private phrases related to location (home, education, etc.) that have information about population in YAGO. However, YAGO does not have a population attribute for other private phrases (such as ones related to work, religion, politics, sports, personal interests). We use the precision metric to quantify information loss related to those types of phrases.

Generalization schemas for two example private phrases about work ("student") and hometown ("Tokyo") are shown in Fig. 4.5. The $InfoLoss$ for them is calculated using

$$InfoLoss(P) = Dis(\text{Tokyo}) + Pre(\text{student}) = \frac{log(13,230,000)}{log(4,299,000,000)} + \frac{0}{5} = 0.74. \quad (4.10)$$

If the user changes a synonym used in a fingerprint, the algorithm can still identify the group to which the discloser belongs. If an attacker removes several private phases, the algorithm identifies a set of candidate groups to which the discloser belongs.

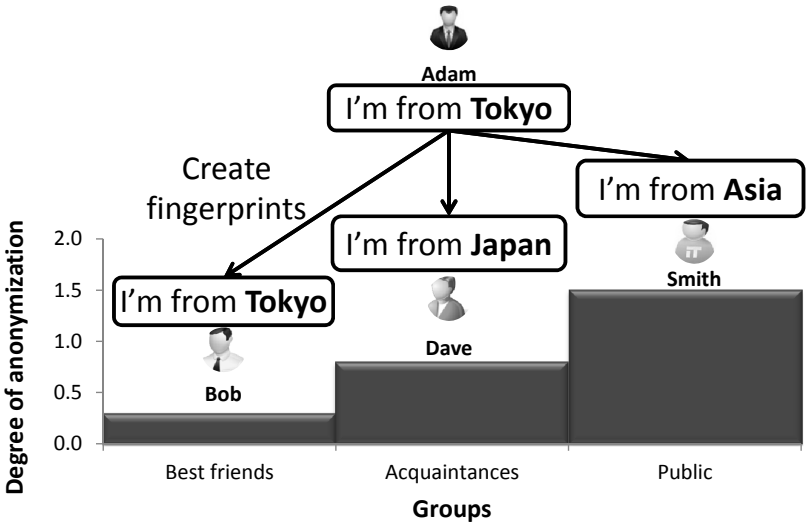


Fig. 4.6 An example of benefit and risk of our approach.

Benefit and risk

Byun and Bertino [10] suggested that groups with higher benefit should receive a version with a lower degree of anonymization. However, it make higher risk of private information disclosure. A distribution metric is used in our algorithm to ensure that the degree of anonymization is appropriate for each group of friends.

For example, for the message "I'm from Tokyo," our algorithm automatically creates fingerprints with different degrees of anonymization ("Tokyo" for *Best friends*, "Japan" for *Acquaintances*, and "Asia" for *Public* group), as shown in Fig. 4.6. In this example, friends in the *Best friends* group with the highest benefit receive the lowest degree of anonymization (Tokyo) while friends in the *Public* group receive the highest one (Asia).

4.2 Anonymization of temporal phrases by generalization

4.2.1 Proposed algorithm

Throughout this section, we use an OSN user status as an input message t "I have an important meeting with my supervisor at 10AM." The steps to anonymize temporal phrases are described in Algorithm 1. The following is description in details of main functions in the algorithm 1.

Algorithm 1 Anonymization of temporal phrases.

```
1: function ANONYMIZETEMPORALPHRASES(input message t, OSN messages S)
2:     TP ← CreateTemporalCorpus(S);
3:     P ← IdentifyTemporalPhrases(t);
4:     if P is not null then
5:         G ← FindGeneralizedTemporalPhrases(P, TP);
6:         NCP* ← QuantifyGeneralizations(G);
7:         F ← CreateFingerprints(G, NCP*, t);
8:         DisplayFingerprints(F);
9:         SaveFingerprints(F);
10:    else
11:        DisplayOriginalMessage(t);
12:    end if
13: end function
```

CreateTemporalCorpus function

Depending on SUTime [13] algorithm, all temporal phrases corpus TP from OSN messages S are extracted by using $CreateTemporalCorpus$ function in Eq. 4.11. TP are used to anonymize temporal phrases on input message t in next steps.

$$TP = CreateTemporalCorpus(S) = \{\text{this morning}, \text{yesterday}, ...\} \tag{4.11}$$

IdentifyTemporalPhrases function

Temporal phrases in t are identified using $IdentifyTemporalPhrases$ function in Eq. 4.12. In the example above, the set of detected temporal phrases P is composed of $p_0 =$ "at 10AM".

$$P = IdentifyTemporalPhrases(t) = \{p_0\} = \{\text{at 10AM}\} \tag{4.12}$$

FindGeneralizedTemporalPhrases function

Each temporal phrase in P is compared with all temporal phrases corpus TP to find all possible generalized temporal phrases by using $FindGeneralizedTemporalPhrases$ function in Eq. 4.13. The set $G^{(i)}$ includes all generalized phrases can replace for temporal phrase p_i to anonymize time-related information. It means the temporal phrase p_i can be anonymized

Normalization	Description	Time duration
MO	Morning	[05:00:00, 11:59:59]
MI	Midnight	[23:00:00, 01:00:00]
AF	Afternoon	[12:00:00, 17:59:59]
EV	Evening	[18:00:00, 20:29:59]
NI	Night	[20:30:00, 23:59:59]
DT	Daytime	[05:00:00, 16:00:00]

Table 4.4 Time duration.

by $g_j^{(i)}$ if the duration of temporal phrase $g_j^{(i)}$ fully covers duration of temporal phrase p_i. Technique to extract duration of temporal phrase is described below.

$$G^{(0)} = FindGeneralizedTemporalPhrases(p_0, S) = \{g_j^{(0)}\} = \{\text{this morning}, \text{today}\} \tag{4.13}$$

The SUTime algorithm determines normalization of temporal phrases. For example, temporal phrase p_0="at 10AM" is normalized by "2014-04-04T10" which the time is pointed to the specific day "April 4, 2014". Another example, temporal phrase in corpus TP "this morning" is normalized by "2014-04-04TMO."

SUTime refers date duration of temporal phrases on basis of the normalization. For example, normalized phrase "2014-04-04T8" and "2014-04-04TMO" is referred as date duration being between "2014-04-04" and "2014-04-04."

However, the SUTime does not mention about time duration (hour, minute, second). Therefore, we extend the SUTime method by extracting time duration. For example, time duration of the normalized phrase "2014-11-01T10" is between "2014-04-04 10:00:00" and "2014-04-04 10:00:00". We also define other time durations in Table 4.4. Depending on this table, time duration of the normalized phrase "2014-04-04TMO" is between "2014-04-04 05:00:00" and "2014-04-04 11:59:59".

QuantifyGeneralizations function

All possible generalized temporal phrases $G^{(i)}$ are quantified information loss by using modified normalized certainty penalty metric (NCP^*):

$$NCP^*(time_O, time_A) = \frac{end_O - start_O + 1}{end_A - start_A + 1}. \tag{4.14}$$

Generalizations $G^{(0)}$	**NCP***	**Friends**
$g_0^{(0)}$="this morning"	3.97E-5	Adam Ebert
$g_1^{(0)}$="today"	1.16E-5	Bob Smith
$g_2^{(0)}$="this month"	1.45E-7	Charlie Lambert
$g_3^{(0)}$="this spring"	1.27E-7	Dave Henderson
...	...	...
$g_j^{(0)}$="this year"	3.17E-8	Ellen Anderson
...	...	...

Table 4.5 Generalizations $G^{(0)}$ of p_0 "at 10AM."

Time of original temporal phrase $time_O$ happens between $start_O$ and end_O while duration of the anonymized temporal phrase $time_A$ is between $start_A$ and end_A. However, $start_X$ and end_X is equal in case of $time_X$ is a point time (such as "10a.m.", "2 o'clock"). Therefore, we add-one in numerator and denominator to ensure that this metric could apply for these cases.

For example, NCP^* of original time "at 10AM" and generalized time "this morning" is calculated in Eq. 4.15. All possible generalizations $G^{(0)}$ of the temporal phrase p_0 "at 10AM" is described to Table 4.5. Sorting the rows in the table by the NCP^* value results in the level of anonymization increasing from top to bottom, as shown in the second column of Table 4.5. A generalization with an appropriate level of anonymization is then used for each proper level of friends to receive the posted message, as shown in the last column. The proper levels of user's friends are identified by using Descioli's method [21].

$$\begin{aligned} NCP^*(\text{"at 10PM"}, \text{"this morning"}) &= \frac{end_O - start_O}{end_A - start_A} \\ &= \frac{\text{[10:00:00]-[10:00:00]+1}}{\text{[11:59:59]-[05:00:00]+1}} = 3.97E-5 \end{aligned} \tag{4.15}$$

CreateFingerprints function

Using the generalized temporal phrases $G^{(i)}$, the algorithm creates fingerprints by replacing the temporal phrases p_i of the input message t with appropriate generalized temporal phrases. For example, the blog entry t' "I have an important meeting with my supervisor *this morning*" is created for *Adam Ebert*.

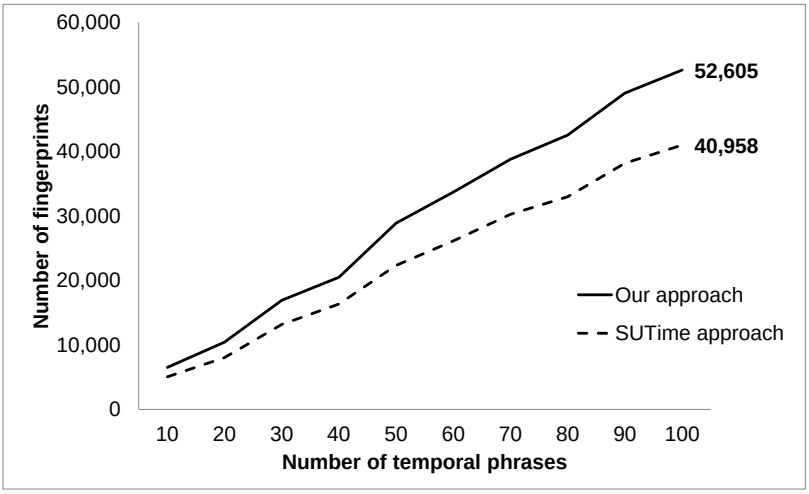


Fig. 4.7 Number of fingerprints for user's friends.

4.2.2 Evaluation

Temporal phrases are created using a dataset, which was drawn from the 16 million tweets in the TREC Tweets2011 Dataset [61]. The dataset was created in three steps.

- *Step 1*: The English tweets in the TREC Dataset were extracted using a language detection tool [74].
- *Step 2*: The extracted tweets were normalized (i.e., words were converted to correct spellings) using lexical normalization [32].
- *Step 3*: Temporal phrases in the normalized tweets are extracted by using the SUTime [13].

The result is a dataset containing 16,647 different temporal phrases. 16,547 temporal phrases from the dataset are used as temporal phrases corpus. The remained dataset is used to calculate the number of fingerprints. The experiment with SUTime approach just extracting date duration of temporal phrases and our approach fully extracting time duration is shown in Fig. 4.7.

With 100 temporal phrases using in the experiment, the number of fingerprints with our proposed algorithm is 52,605 fingerprints. This is significantly higher than the 40,958 fingerprints with obtained with SUTime. The average number of fingerprints is 526.05 fingerprints make sure that it covers most of the potential cases of OSN disclosure.

4.2.3 Discussion

Strength of fingerprints

The algorithm can be used to anonymize private information revealed by temporal phrases not only for OSNs but also for other areas (e.g., health, military, and news) that store private information.

It can be used to anonymize not only time-related information but also other types of information reflected in other parts of the parsing tree of message (location information, objective information, etc.).

Although we focused on English OSN message in this section, the algorithm can be applied to other languages as well.

Limitation and future work

Many messages include temporal phrases but do not reveal any private information about user. For example, a general message "it is rainy *tomorrow*" does not contain any private information.

Future work identifies messages containing private temporal phrases. Moreover, we will anonymize private actions of users (verbs in messages) posted in OSN.

4.3 Anonymization of temporal phrases by suppression

4.3.1 Temporal phrase features

Temporal phrases in English are mostly independent of the other phrases in the sentence. This is illustrated by the parsing tree in Fig. 4.8 for the text "I go to Tokyo with friends at 9AM". Deleting the temporal phrase ("at 9AM") corresponds to deleting the portion of the parsing tree shown in bold. The portion of the sentence remaining ("I go to Tokyo with friends") sounds natural and retains its meaning.

Moreover, many temporal phrases have a similar parsing tree structure. For example, the parsing tree structure for "at 9AM" is the same as that for "at night" in "Mary eats sushi at night" (Fig. 4.9). Therefore, the parsing tree structure for one time phrase can be used to identify other temporal phrases with the same structure.

However, parts of a parsing tree that are not a temporal phrase may have the same structure. For example, the parsing tree for "with friends" in Fig. 4.8 has the same structure as that for "at 9AM." However, "with friends" is not a temporal phrase. Therefore, to distinguish the temporal phrase parts from the other parts, we combine the results of temporal phrase

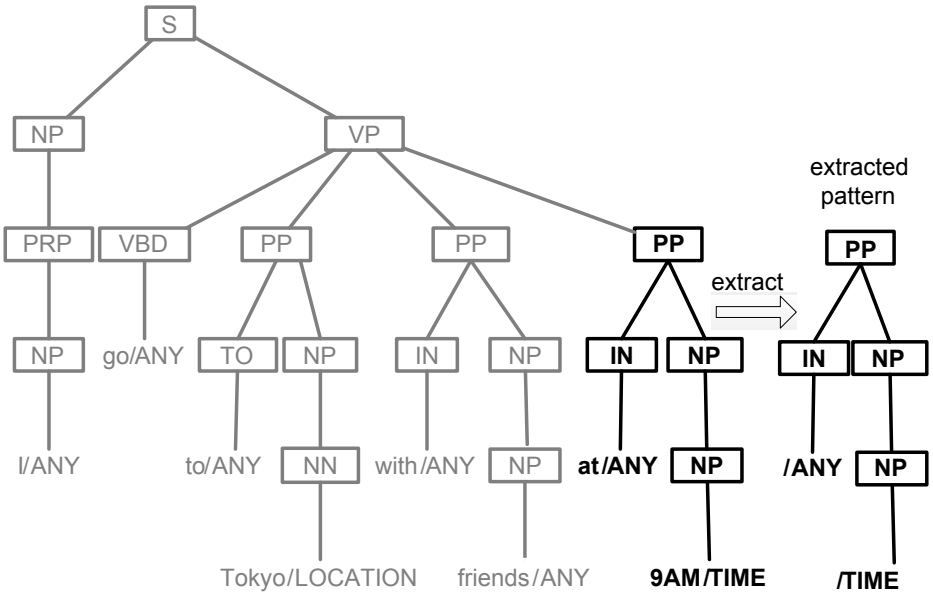


Fig. 4.8 Parsing tree for "I go to Tokyo with friends at 9AM."

tagging using SUTime for all leaves of the parsing tree to create patterns for identifying temporal phrases. An example of such a pattern is shown on the right in Fig. 4.8.

By using these learned patterns, the algorithm identifies all temporal phrases have the same parsing tree structure and the same tags for all leaves. The result for "Mary eats sushi at night" is shown on the right in Fig. 4.9. The temporal phrase "at night" has been removed from the sentence.

4.3.2 Proposed method

Our proposed algorithm has two main phases, *pattern extraction* and *text anonymization*, as illustrated in Fig. 4.10.

In the *pattern creation* phase, the input is OSN corpus T (e.g., status updates, blog entries, comments). From T, the algorithm extracts many temporal phrase patterns. These patterns are used in the text anonymization phase to anonymize OSN.

In the *text anonymization* phase, all OSN text is anonymized by deleting all temporal phrases using the patterns P extracted in the first phase.

Pattern extraction

$T = \{t_0, t_1, ..., t_n\}$. Patterns P are extracted from each text in three steps.

- *Step 1*: Normalize input text t.

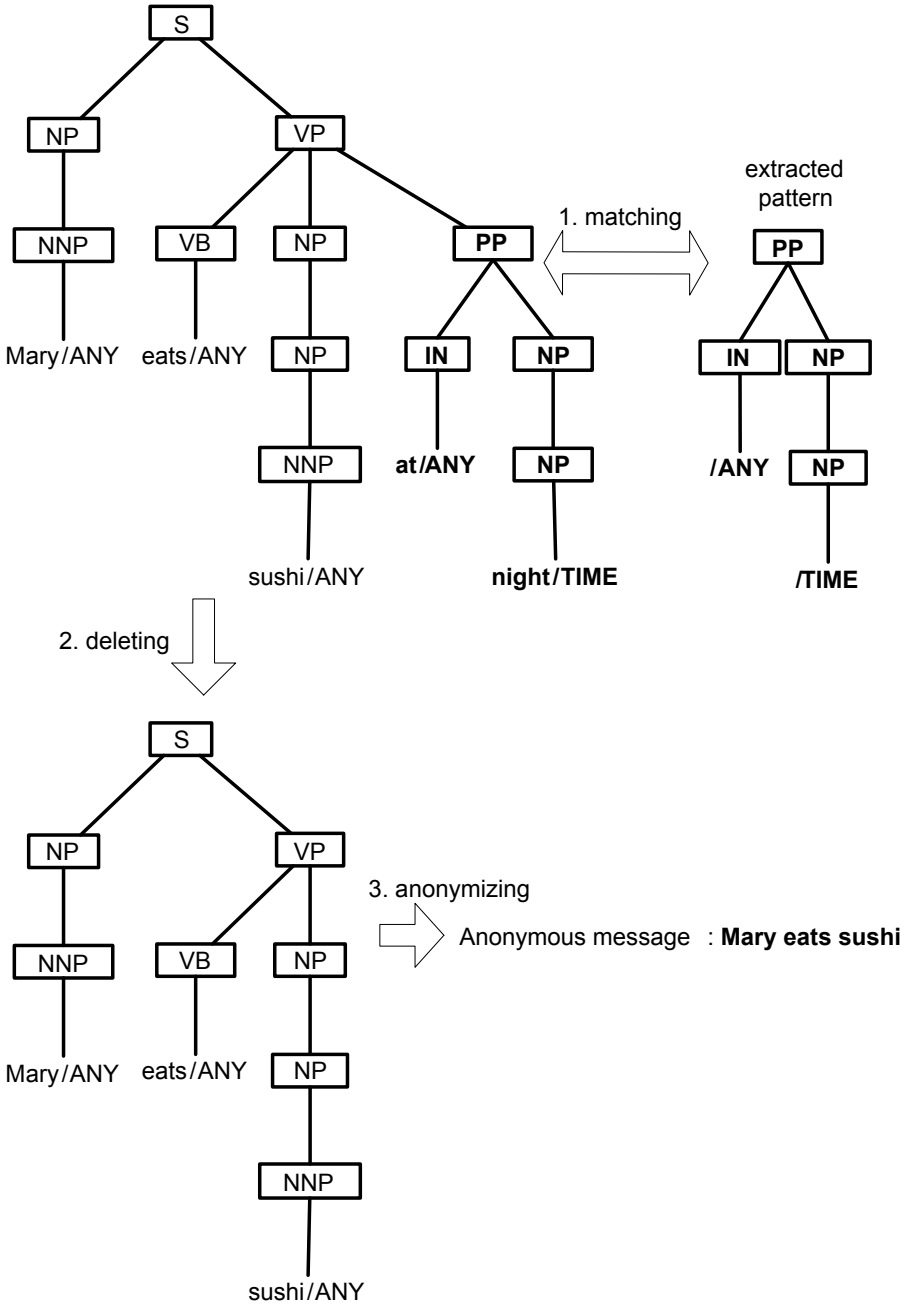


Fig. 4.9 Identify various temporal phrases using one pattern.

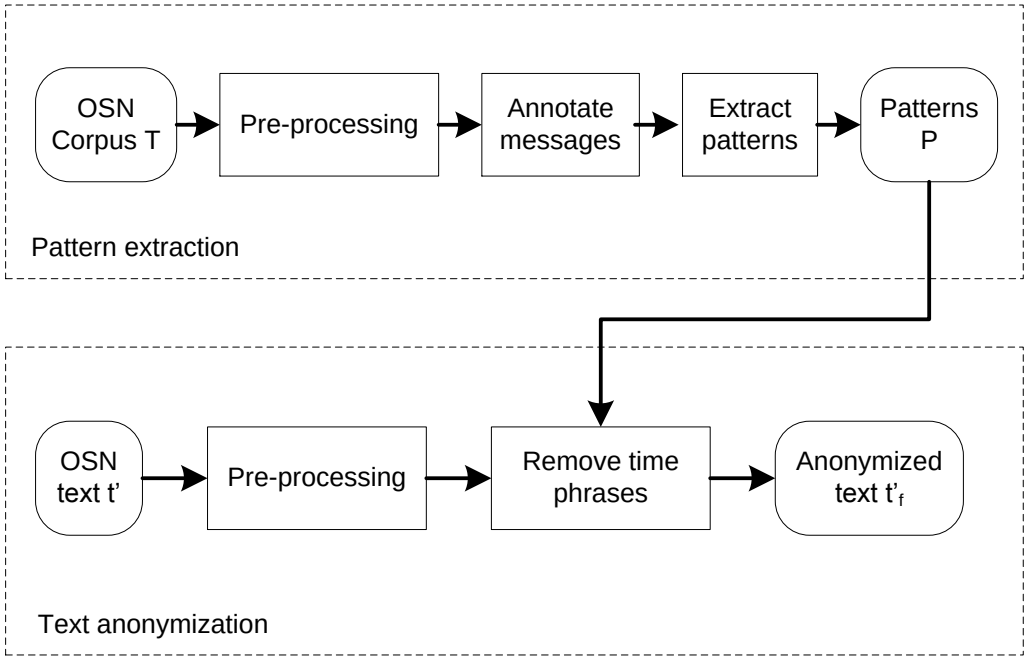


Fig. 4.10 Phases of proposed algorithm.

- *Step 2*: Manually determine temporal phrases in normalized text t_n.
- *Step 3*: Extract pattern from annotated text t_a.

The following is a step-by-step description of this process using the blog entry t "I g0 to Tokyo w/ friends at 9AM."

Normalize input text (Step 1)

The input text is normalized in two sub-steps:

1. Determine whether input text is English by using language detection technique using function γ in Eq. 4.16. Here, we use the language detection library created by Shuyo [74]. If the detected language is English ("en"), text t is normalized in the next sub-step.

$$L = \gamma(t) = \text{en} \tag{4.16}$$

2. The English message is normalized (i.e., words are converted to correct spellings) using function θ in Eq. 4.17. The lexical normalization used here was created by Han et al.[32]. Normalized text t_n of text t is "I go to Tokyo with friends at 9AM."

$$t_n = \theta(t) = \text{"I go to Tokyo with friends at 9AM"} \tag{4.17}$$

Manually determine temporal phrases (Step 2)

In this step, the temporal phrases in the normalized text t_n are manually annotated. In the example above, the temporal phrase "at 9AM" is marked in bold by using function δ in Eq. 4.18.

$$t_a = \delta(t_n) = \text{"I go to Tokyo with friends } \textbf{at 9AM}\text{"} \tag{4.18}$$

Extract pattern from annotated corpus (Step 3)

A parsing tree s_t of the annotated text t_a is created using the Stanford parsing tool [41] by using function η in Eq. 4.19. Parsing tree s_t is created using format of Penn Treebank[3].

$$\begin{aligned} s_t = \eta(t_a) = \text{"(S (NP (PRP I)) (VP (VBP go) (PP (TO to)} \\ \text{(NP (NNP Tokyo))) (PP (IN with) (NP (NP (NNS friends))} \\ \textbf{(PP (IN at) (NP (NN 9AM)))))))}\text{"} \end{aligned} \tag{4.19}$$

Each leaf of the tree is then combined with taggers obtained from the SUTime library [13] by using function μ in Eq. 4.20 as shown on the left in Fig. 4.8.

$$\begin{aligned} s_a = \mu(s_t) = \text{"(S (NP (PRP I/ANY)) (VP (VBP go/ANY)} \\ \text{(PP (TO to/ANY) (NP (NNP Tokyo/LOCATION)))} \\ \text{(PP (IN with/ANY) (NP (NP (NNS friends/ANY))} \\ \textbf{(PP (IN at/ANY) (NP (NN 9AM/TIME)))))))}\text{"} \end{aligned} \tag{4.20}$$

[3] http://www.cis.upenn.edu/~treebank/

The dependence on the temporal phrases is annotated in Step 2. The algorithm can extract the patterns P from input text t by using function λ in Eq. 4.21, as shown on the right in Fig. 4.8.

$$P = \{p_i\} = \lambda(s_a) = p_0 = \text{“(PP (IN /ANY) (NP (NN /TIME))”} \tag{4.21}$$

Text anonymization

OSN text t' is anonymized in two steps by using patterns P extracted in the pattern creation phase.

- *Step 1*: Normalize OSN message t'.
- *Step 2*: Remove temporal phrases from t'.

For example, text t', “Mary eeats sushiii at nite,” is anonymized as follows.

Normalize OSN message (Step 1)

Determine whether the language is English, and normalize the text t' in a manner similar to step 1 in the pattern creation process using functions γ and θ (Eq. 4.22 and Eq. 4.23). In the example above, normalized text t'_n is “Mary eats sushi at night.”

$$L = \gamma(t') = \text{en} \tag{4.22}$$

$$t'_n = \theta(t') = \text{“Mary eats sushi at night”} \tag{4.23}$$

Remove temporal phrases from OSN text (Step 2)

The text t_n is anonymized in the three sub-steps shown in Fig. 4.9.

1. The parsing tree s'_t for the normalized OSN message t'_n is created using the Stanford parser library and function η in Eq. 4.24.

$$\begin{aligned} s'_t = \eta(t'_a) = \text{“(S (NP (NNP Mary)) (VP (VBZ eats)} \\ \text{(NP (NP (NNP Sushi)) (PP (IN at) (NP (NN night))))))”} \end{aligned} \tag{4.24}$$

The SUTime library is then used to determine the bold taggers of for all each leafs nodes in the parsing tree using formula function η in Eq. 4.25.

$$\begin{aligned} s'_a = \mu(s'_t) = \text{“(S (NP (NNP Mary/}\mathbf{ANY}\text{))} \\ \text{(VP (VBZ eats/}\mathbf{ANY}\text{) (NP (NP (NNP Sushi/}\mathbf{ANY}\text{))} \\ \text{(PP (IN at/}\mathbf{ANY}\text{) (NP (NN night/}\mathbf{TIME}\text{))))))”} \end{aligned} \tag{4.25}$$

2. The tree s'_a is matched against all patterns P extracted in the pattern extraction phase by using function α in Eq. 4.26. For each pattern that matches the tree, the parts of tree with that pattern are deleted from the tree.

$$\begin{aligned} s'_r = \alpha(s'_a, P) = \alpha(s'_a, p_0) = \text{“(S (NP (NNP Mary/ANY))} \\ \text{(VP (VBZ eats/ANY)(NP (NP (NNP Sushi/ANY)))))”} \end{aligned} \tag{4.26}$$

3. All remaining leaves are combined to create normalized anonymous text t'_r by using function β in Eq. 4.27. For the previous example, the remaining text t'_r after deleting the temporal phrase is “Mary eats sushi.”

$$t'_r = \beta(s'_r, t_a) = \text{“Mary eats sushi”} \tag{4.27}$$

The normalized anonymous text t'_r is compared with original OSN text t' to create the anonymized text t'_f by using function ϕ in Eq. 4.28. The final anonymized text t'_f is “Mary eeats sushiii.”

$$t'_f = \phi(t'_r, t') = \text{“Mary eeats sushiii”} \tag{4.28}$$

4.3.3 Evaluation

Performance

In order to evaluate the proposed method, we use a dataset which was drawn from the 16 million tweets in the TREC Tweets2011 Dataset [61]. The tweets which contain temporal phrases are extracted in three steps.

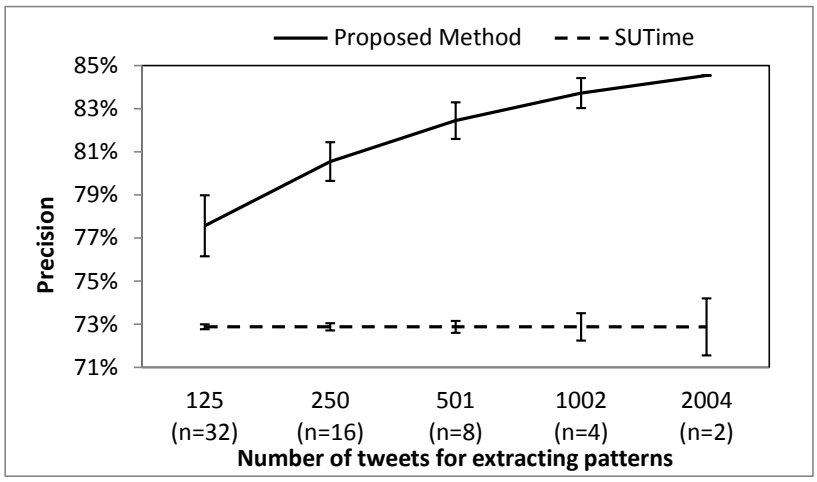


Fig. 4.11 Anonymization results.

- *Step 1*: The English tweets in the TREC Dataset were extracted using a language detector. The library used was created by Shuyo et al. [74]
- *Step 2*: The extracted tweets were normalized (i.e., words were converted to correct spellings) using lexical normalization on the basis of the normalization dictionary created by Han et al. [32].
- *Step 3:* The temporal phrases in the normalized tweets were detected using the Stanford Temporal Tagger: SUTime [13].

We manually annotated the 4008 tweets in which temporal phrases were identified.

These 4008 tweets are randomly partitioned into n equal size sub-parts. Of these n sub-parts, a single sub-part is selected for extracting patterns and the remaining $n-1$ ones were used for testing. The n results are averaged to produce precision and its standard deviation. The precision was metric, meaning that the anonymized tweets sounded natural and that all temporal phrases had been deleted. The results are plotted in Fig. 4.11.

The highest precision with our proposed algorithm is 84.53% (standard deviation 0.01%). This is significantly higher than the 72.88% (standard deviation 1.32%) with obtained with SUTime. Its precision was even higher than that of SUTime method when only 125 patterns were used. This means that only a few patterns are needed to anonymize most temporal phrases in OSN messages.

Common patterns

Some common patterns for anonymizing temporal phrases are extracted from 2000 tweets shown in Table 4.6. The top ten patterns accounted for 87.78% of the remaining 2208 tweets.

Patterns	**Frequency**
/DATE	2334
/DURATION	371
/TIME	334
(ADVP (RB /DATE) (RB /DATE))	106
/SET	56
(NP (DT /DATE) (NN /DATE))	47
(NP (CD /DURATION) (NNS /NUMBER))	47
(PP (IN /ANY) (NP (NN /DATE)))	37
(NP (DT /ANY) (NN /TIME))	36
(PP (IN /ANY) (NP (NNP /DATE)))	35
(NP-TMP (JJ /DATE) (NN /DATE))	25
(NP (DT /DURATION) (NN /DURATION))	25
(PP (IN /ANY) (NP (RB /DATE)))	24
(PP (IN /ANY) (NP (CD /DURATION) (NNS /NUMBER)))	22
(NP (DT /ANY) (NN /DURATION))	20
(PP (IN /ANY) (NP (CD /TIME)))	20
(NP-TMP (JJ /TIME) (NN /TIME))	20
(PP (IN /ANY) (NP (NN /TIME)))	19
(PP (IN /ANY) (NP (DT /DATE) (NN /DATE)))	18
(PP (IN /ANY) (NP (DT /ANY) (NN /TIME)))	17
(PP (IN /ANY) (NP (CD /DATE)))	15
(PP (IN /ANY) (NP (DT /DURATION) (NN /DURATION)))	15
(PP (IN /ANY) (NP (NNS /DURATION)))	13
(NP (DT /ANY) (NNS /DURATION))	11
(NP-TMP (DT /DATE) (NN /DATE))	11

Table 4.6 Popular Patterns.

These patterns are shown that the temporal phrases most same structure of parsing tree. The results indicate that only a few main patterns matched most of the temporal phrases in the test dataset.

4.3.4 Discussion

The algorithm can be used to anonymize private information revealed by temporal phrases not only for online social networks but also for other areas (e.g., health, military, news) that store private information.

It can be used to anonymize not only time-related information but also other types of information reflected in other parts of the parsing tree of text (location information, objective information, etc.).

Although we focused on English OSN text in this section, the algorithm can be applied to other languages as well.

Although temporal phrases generally have the same parsing tree structure and are mostly independent of other parts of the tree, they are important parts of the text in many sentences. In such sentences, deleting the temporal phrases results in an unnatural sentence. Therefore, the temporal phrases cannot be deleted, as illustrated by the following examples.

Example 1: "*Tomorrow* is ma mums birthday how i wish i was home to celebrate with ma family!"

Example 2: "Man *today* is the big day Chicago vs Green Bay let 's go #TeamBears"

Example 3: "Man - a creature made at *the end of the week*'s work when God was tired."

Example 4: "*Today* was my day to bathe!"

Example 5: "@JonathanRKnight hey babe, hope your having a wonderful *weekend* : sending you muchos kisses!"

In these sentences, the temporal phrases are often the main subject of objective. In such cases, the temporal phrases should be replaced with anonymous phrases rather than deleting them.

4.4 Summary

4.4.1 Anonymization of private phrases

We propose an algorithm to anonymize private phrases in text messages to be posted in an OSN by generalizing them in accordance with the disclosure level for each group of friends and fingerprints the messages by synonymization. By using these fingerprints, the algorithm can detect which friend has disclosed private information about the user. The detection of disclosure based on the fingerprints is described in Chapter 5.

A frequency threshold is used to check the naturalness of the fingerprinted messages to avoid attracting the attention of attackers. A co-occurrence metric is used to identify private phrases even if an attacker directly or indirectly changes the fingerprints. A distribution metric is used to ensure that each group of friends receives a version of the message with an appropriate degree of anonymization.

Evaluation using about 55,000 personal tweets in English showed that our algorithm can create more fingerprints than previous ones that use synonyms. It can create a sufficient number of fingerprints for all of the friends of a typical Facebook user.

Future work includes anonymizing the actions described in the natural language texts of users in order to prevent detection of a user's location by using messages posted on the Internet.

4.4.2 Anonymization of temporal phrases

We develop an algorithm for anonymizing temporal phrases of messages to be posted in online social networks by generalization. With our proposed algorithm, time-related information can be anonymized differently for each user's friend. Moreover, a user can detect and identify the disclosing person if his or her information has been disclosed. The algorithm is based on the use of generalized temporal phrases extracted form OSN corpus to anonymize time-related information and create a unique fingerprint for each person who will see the message. The algorithm creates sufficient fingerprints for a unique one to be assigned to each of the user's friends. The fingerprints are quantified using a modified normalized certainty penalty metric so that appropriate level of anonymization is used for each user's friend.

We propose another algorithm for deleting temporal phrases so as to anonymize the time-related private information. Analysis of OSN texts showed that most temporal phrases in OSN messages have the same parsing tree structure. The proposed algorithm is thus based on the use of temporal phrase patterns. Tagging of the identified temporal phrases is used to integrate into structure of temporal phrase parsing tree to distinguish with other parts of parsing tree. The precision was 84.53% for 4008 OSN sentences, showing that this approach is good for anonymizing temporal phrases in OSN text. Of the learned patterns, the top ten most common ones were used to identify 87.78% the temporal phrases. This means that only some of the most common patterns can be used to anonymize temporal phrases in most messages to be posted on an OSN. Our algorithm is applicable not only to temporal phrases but also to other phrases (location, objective, etc.), to other areas (health, religion, politics, military, etc.) that store private information about user's activities, and to other languages.

Chapter 5

Detection of disclosure

After creating fingerprints in Chapter 4, all disclosed messages containing fingerprints are collected by using search engines (such as Google search engine, and/or Facebook API search). Each disclosed message is compared with fingerprinted messages for determine whether they have the same meaning or not by paraphrase detection. We propose a *SimMat* metric for detecting paraphrases that is based on matching identical phrases and similar words, as described in Section 5.1. The practicality of the thesis was demonstrated by a web application for controlling messages posted on Facebook, as illustrated in Section 5.2.

5.1 Detection of paraphrase

5.1.1 Proposed method

Similarity matching (*SimMat*) metric

Our proposed similarity metric (*SimMat*) for quantifying the similarity of input text comprises four steps, as illustrated in Fig. 5.1.

- *Step 1 (match identical phrases)*: All phrases with the same lemmas and order are matched using a heuristic algorithm. The key idea of the heuristic algorithm is that it matches the two identical phrases (if any) with the maximum length in each iteration.

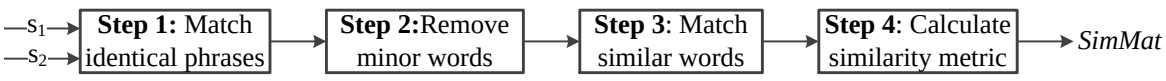


Fig. 5.1 Four steps in calculation of similarity matching (*SimMat*) metric.

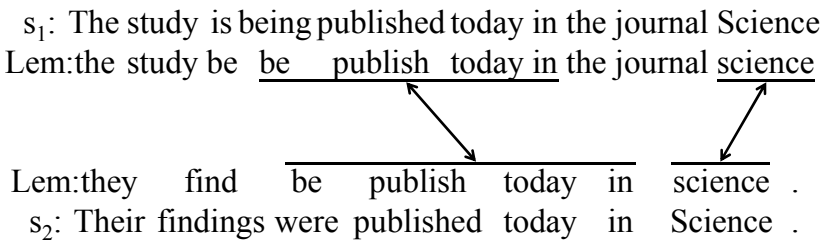


Fig. 5.2 Matching identical phrases with their maximum lengths (Step 1).

- *Step 2 (remove minor words)*: The words remaining after identical phrase matching are analyzed and the minor words are deleted. We specify four types of minor words: prepositions and subordinating conjunction, modal verbs, possessive pronouns, and periods (".").
- *Step 3 (match similar words)*: The words remaining of the two sentences after minor word removal, are matched using the Kuhn-Munkres algorithm [43, 54]. The WordNet similarities of the words in the two text segments are used as weights for the matching algorithm.
- *Step 4 (calculate similarity metric)*: The similarity matching *SimMat* metric is calculated on the basis of the results of the matched identical phrases and similar words.

The following is a step-by-step description of our method using two sentences, which is an actual paraphrase pair from the MSRP corpus.

s_1: "The study is being published today in the journal Science"

s_2: "Their findings were published today in Science."

Match identical phrases (Step 1)

The individual words in the two input sentences are normalized using lemmas. The Natural Language Processing (NLP) library of Stanford University [49] is used to identify the lemmas. The lemmas for the two example sentences are shown in Fig. 5.2.

The heuristic algorithm we developed for matching the lemmas in the two sentences repeatedly finds a new matching pair in each round. In each round, a new pair with the maximum phrase length is established. The pseudo code of the algorithm is illustrated in Algorithm 2. The stop condition is when there is no new matching pair. For example, two identical lemma of phrases, "be publish today in" and "science," are matched (as shown as Fig. 5.2).

In algorithm 2, the function $getLemmas(s)$ extracts the lemmas of sentence s using the NLP library. The function len_L gets the number of elements in set L. The function

Algorithm 2 Match identical phrases with maximum length.

```
1:  function MATCHIDENTICALPHRASES(s1, s2) : P
2:      L1 ← getLemmas(s1);
3:      L2 ← getLemmas(s2);
4:      P ← ∅;
5:      repeat
6:          new ← ∅;
7:          for i = 0 to len_L1 − 1 do
8:              for j = 0 to len_L2 − 1 do
9:                  if {L1[i], L2[j]} ∉ P then
10:                     tmp ← match(L1[i], L2[j]);
11:                     if len_tmp > len_new then
12:                         new ← tmp;
13:                     end if
14:                 end if
15:             end for
16:         end for
17:         if new is not null then
18:             P = P ⋃ new;
19:         end if
20:     until (new=∅);
21:     return P;
22: end function
```

$match(L_1[i], L_2[j])$ finds the maximum length matching of phrase L_1, which starts at the i-th position in the first sentence, and that of phrase L_2, which starts at the j-th position in the second sentence.

Remove minor words (Step 2)

The words remaining after phrase matching in Step 1 are used for removing minor words. First, the part of speech (POS) for each word is identified. The Stanford library tool [49] is used for this purpose. The POSs for the words in the two example sentences are shown in Fig. 5.3.

Our analysis of the common practices of plagiarists showed that four types of minor words should be removed: prepositions and subordinating conjunctions (IN), modal verbs (MD), possessive pronouns (PRP$), and periods (“.”). These minor POSs generally do not change the meaning of the paraphrased text as they are often used to simply improve the naturalness of the paraphrased text. For example, the two minor POSs (PRP$ and “.”) were deleted from sentence s_2 in Fig. 5.3. An example of preposition deletion is illustrated in

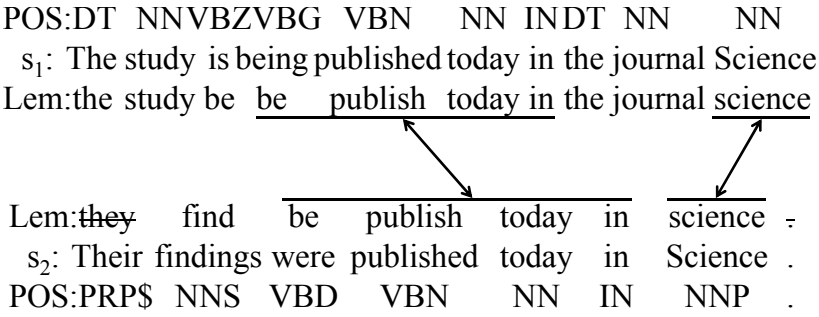


Fig. 5.3 Remove minor words (Step 2).

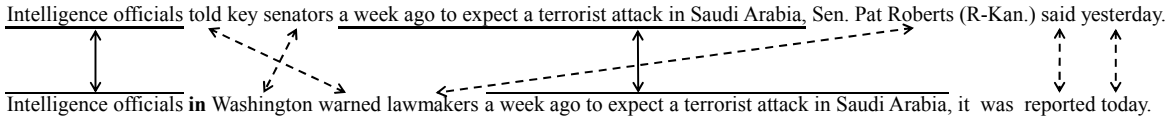


Fig. 5.4 Example paraphrase pair taken from MSRP corpus.

Fig. 5.4. Detection of remaining type of minor words (modal verbs) is illustrated for an actual paraphrase pair in Fig. 5.5.

Match similar words (Step 3)

After minor word deletion in Step 2, the perfect matching of similar words is done using the algorithm we developed on the basis of the Kuhn-Munkres algorithm [43, 54]. The weights of each pair in the algorithm are calculated from the similarity of the two lemmas of the words using the *path* metric [65]. The $path(w_1, w_2)$ metric computes the shortest path (*pathLength*) between two words w_1 and w_2 in the 'is-a' hierarchies of WordNet, as shown in Eq. 5.1. The *pathLength* is constrained to be a positive integer to ensure that $0 <= path <= 1$. For example, the *path* metric for the "study" and "find" pair is 0.33. The perfect matching found for the two example sentences is shown in Fig. 5.6. The word "study" in sentence s_1 is matched with a similar word, "findings," in sentence s_2.

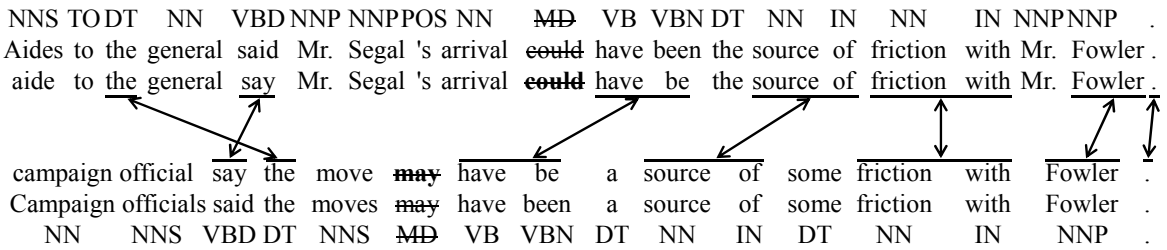


Fig. 5.5 Example of removing minor words (modal verbs).

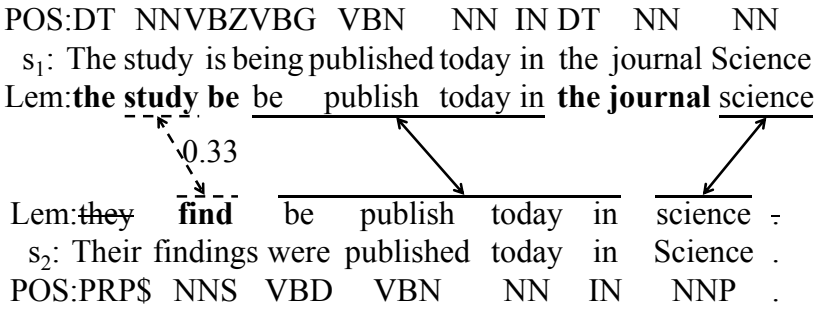


Fig. 5.6 Find perfect matching of similar words using Kuhn-Munkres algorithm [43, 54] (Step 3).

$$path(w_1, w_2) = \frac{1}{pathLength(w_1, w_2)} \tag{5.1}$$

Calculate similarity metric (Step 4)

Finally, the *RelMat* metric is calculated using the results of identical phrase matching in Step 1 and similar word matching in Step 3:

$$RelMat(s_1, s_2) = \frac{\#Np + \sum_{i=0}^{N-1} len(p_i)^\alpha + \sum_{j=0}^{M-1} path(w_j)^\alpha}{\#Np + \#Nw + \sum_{i=0}^{N-1} len(p_i)^\alpha + \sum_{j=0}^{M-1} 1^\alpha}, \tag{5.2}$$

where $\#Np$ is the total number of words in the matched identical phrases, $\#Nw$ is the number of matched similar words, N and M are the corresponding numbers of matched identical phrases and similar words, p_i is the i-th matched phrase in Step 1, $len(p_i)$ is the number of words in the phrase p_i, and $path(w_j)$ is the *path* metric of the j-th matched word in Step 3.

Eq. 5.2 ensures that $0 <= RelMat <= 1$. The *RelMat* metric equals 1 only if the two sentences are identical. Using $\#Np$ only in the numerator means that the matching of identical phrases is more important than the matching of similar words. The $len(p_i)^\alpha$ and $path(w_j)^\alpha$ with $\alpha >= 0$ indicate the respective contributions of matched phrase p_i and matched word w_j to the *RelMat* metric. The greater the value of α, the greater the contribution of the identical phrases and the lesser the contribution of the similar words. Because $0 <= path(w_j) <= 1$, we use 1^α to normalize the contributions of the matched words.

Threshold α is set to an optimal value of 0.2, as described in more detail in Section 5.1.2. The *RelMat* metric for the two example sentences is calculated using

$$RelMat(s_1, s_2) = \frac{5 + (4^{0.2} + 1^{0.2}) + 0.33^{0.2}}{5 + 1 + (4^{0.2} + 1^{0.2}) + 1^{0.2}} = 0.87.$$

The remaining words are probably modified by few manipulations (e.g., insertion, deletion). Such modification is typically intended to improve the naturalness of text. Therefore, the two sentences being compared frequently have different lengths. To reduce this effect, we developed a brevity penalty metric p based on the METEOR metric [20]. It is calculated as shown in Eq. 5.3, where $\#ReW(s)$ is the number of words remaining in sentence s after phrase matching and minor word removal. Penalty p is combined with *RelMat* into the similarity matching *SimMat* metric, as shown in Eq. 5.4.

$$p(s_1, s_2) = 0.5 \times \left(\frac{|\#ReW(s_1) - \#ReW(s_2)|}{max(\#ReW(s_1), \#ReW(s_2))}\right)^3 \tag{5.3}$$

$$SimMat = RelMat \times (1 - p) \tag{5.4}$$

Penalty metric p and the *SimMat* metric are respectively calculated for the example sentences using Eq. 5.5 and Eq. 5.6. To calculate the $\#ReW$, the remaining words (in bold) are shown in Fig. 5.6.

$$p(s_1, s_2) = 0.5 \times \left(\frac{|5 - 1|}{max(5, 1)}\right)^3 = 0.26 \tag{5.5}$$

$$SimMat = 0.87 \times (1 - 0.26) = 0.64 \tag{5.6}$$

Combination of *SimMat* metric and MT metrics

We proposed paraphrase detection method by combining the *SimMat* metric with the eight standard MT metrics described below, as shown in Fig. 5.7.

- *Step 1 (calculate SimMat metric)*: The *SimMat* metric is calculated for the two input sentences as described in Section 5.1.1.

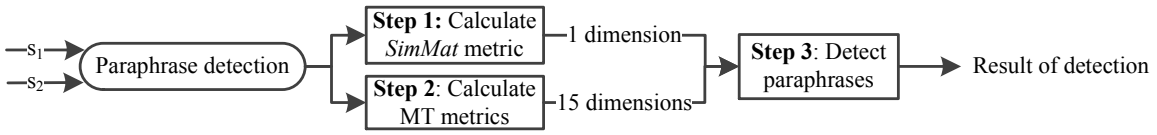


Fig. 5.7 Combination of *SimMat* metric with eight MT metrics.

- *Step 2 (calculate MT metrics)*: The eight MT metrics are calculated using standard libraries to create fifteen dimensions. These libraries are suggested by the state-of-the-art approach for paraphrase detection [48] and the National Institute of Standards and Technology (NIST), United States[1].

- *Step 3 (detecting paraphrases)*: One dimension from our similarity metric (*SimMat*) and 15 from the 8 MT metrics are combined to detect paraphrases. Logistic regression is used for this as it is the best machine learning algorithm for detecting paraphrases.

The last two steps are described in detail below.

Calculate MT metrics (Step 2)

The eight standard MT metrics are calculated for the two sentences. Eight libraries are used to quantify them. These libraries are suggested by NIST and the state-of-the-art approach for paraphrase detection [48]. The libraries are described in more detail in the evaluation section. The first six MT metrics (MAXSIM, SEPIA, TER, TERp, METEOR, and BADGER) create six dimensions in total. The two remaining metrics (BLEU and NIST) using the n-gram model create four (n=1..4) and five (n=1..5) dimensions, respectively. These 15 dimensions metrics are combined with that of our proposed metric (*SimMat*) for detecting paraphrases in the last step.

Detecting paraphrases (Step 3)

The 16 dimensions, 15 from the MT metrics and 1 from our proposed metric (*SimMat*) are combined for detecting paraphrases using a machine learning approach. Several commonly used machine learning algorithms (including support vector machine, naive Bayes, and logistic regression) were evaluated with these dimensions. Such algorithms are run with 10-fold cross validation in the training set of the MRPS corpus for choosing the best classifier. Logistic regression had the best performance and was thus used for detection.

[1] http://www.itl.nist.gov/iad/mig/tests/metricsmatr/2010/results/metrics.html

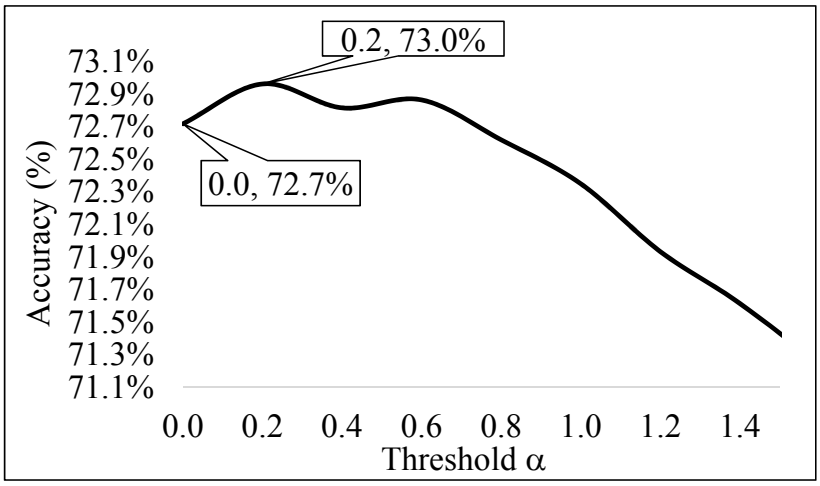


Fig. 5.8 Estimated threshold α.

5.1.2 Evaluation

MSRP corpus

We used the Microsoft Research Paraphrase (MSRP) corpus [24] to evaluate our method. It is the most commonly used corpus in the paraphrase detection field. It contains 5801 sentences pairs including 4076 for training and the remaining 1705 for testing.

The corpus has 2753 (67.5%) and 1147 (66.5%) paraphrase cases corresponding to training and testing datasets. The corpus was annotated by two native speakers. Disagreements in annotation were resolved by a third native speaker. Agreement between the two annotators was moderate to high (averaging 83%). This means that a perfect algorithm for detecting paraphrases would have 83% accuracy.

Estimating threshold α for *SimMat* metric

A threshold α is used to adjust the contributions of matched identical phrases and matched similar words. It was estimated using Eq. 5.2 and the training dataset of the MRPS corpus. Only the *SimMat* metric was used as the single dimension for the logistic regression algorithm with 10-fold cross validation, as shown in Fig. 5.8. Using only the training dataset ensured that the results did not overfit the test data.

The higher the threshold α, the greater the contribution of the matched identical phrases and the lesser the contribution of the matched similar words. If α is small, the contributions of identical phrases are low and the contributions of similar words are high, resulting in lower accuracy. However, the *SimMat* metric is over-estimated if the value of α is too

MT metric	Re-implementation			MTMETRICS	
	Version	Accuracy	F-score	Accuracy	F-score
MAXSIM	1.01[3]	67.5%	79.4%	67.2%	79.4%
SEPIA	0.2[4]	68.3%	79.8%	68.1%	79.8%
TER	1.01[5]	70.1%	81.0%	69.9%	80.9%
TERP	1.0[6]	70.7%	81.0%	74.3%	81.8%
BADGER	2.0[7]	67.2%	79.9%	67.6%	79.9%
METEOR	1.5[8]	71.7%	80.0%	73.1%	81.0%
BLEU	13a[9]	72.1%	80.8%	72.3%	80.9%
NIST	13a[10]	71.8%	80.4%	72.8%	81.2%
Integration		76.6%	83.1%	77.4%	84.1%

Table 5.1 Results for re-implemented MT metrics and MTMETRICS algorithm [48].

large, resulting in lower accuracy. The highest accuracy (73.0%) was achieved for $\alpha = 0.2$. Therefore, α was set to 0.2 for the subsequent experiments.

MT metrics result

In our approach, the proposed metric (*SimMat*) is combined with eight MT metrics (MAXSIM, SEPIA, TER, TERp, METEOR, BADGER, BLEU, and NIST). These metrics are integrated to create what we call the MTMETRICS algorithm, which is state of the art for paraphrase detection. The eight metrics are re-implemented on the basis of standard libraries suggested by both of the state of the art and an organization – NIST[2]. The details of the re-implementation are shown in Table 5.1.

The versions of the eight libraries for the re-implemented metrics are shown in column 2. They were the latest for each library, for which we used the default settings. Since the versions and settings are not shown for MTMETRICS, there is little difference between the re-implemented metric results and the MTMETRICS results. The results for the integration

[2]http://www.itl.nist.gov/iad/mig/tests/metricsmatr/2010/results/metrics.html
[3]http://www.comp.nus.edu.sg/~nlp/sw/
[4]http://www1.ccls.columbia.edu/~SEPIA
[5]http://www.cs.umd.edu/~snover/tercom/
[6]http://web.archive.org/web/20140718062724/http://www.umiacs.umd.edu/~snover/terp/
[7]http://www.babblequest.com/badger2
[8]http://www.cs.cmu.edu/~alavie/METEOR/index.html
[9]ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a-20091001.tar.gz
[10]ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a-20091001.tar.gz

Method	Accuracy	F-score
Vector Based Similarity (baseline)	65.4%	75.3%
Mihalcea et al. [52]	70.3%	81.3%
Qiu et al. [67]	72.0%	81.6%
SimMat	72.7%	81.3%
Blacoe and Lapata [8]	73.0%	82.3%
Finch et al. [26]	75.0%	82.7%
Das and Smith [19]	76.1%	82.7%
Madnani et al. [48] (re-implemented)	76.6%	83.1%
Socher et al. [77]	76.8%	83.6%
Madnani et al. [48]	77.4%	**84.1%**
Combination	**77.6%**	83.9%

Table 5.2 Accuracy and F-score of our method (*SimMat*), previous methods, and combination of *SimMat* with eight MT metrics.

of the eight re-implemented metrics (accuracy=76.6%, F-score=83.1%) also differ from the MTMETRICS results (accuracy=77.4%, F-score=84.1%).

Comparison with previous methods

The results of our comparison with previous methods are summarized in Table 5.2. These methods were also evaluated using the MSRP corpus. Our proposed metric (*SimMat*) was evaluated using a threshold α of 0.2. This single metric outperformed many previous methods. The combination of *SimMat* with the eight MT metrics had the highest accuracy (77.6%).

5.1.3 Discussion

The strength of our method is that it is based on the common practices of people when they paraphrase sentences or phrases. These practices include reordering phrases and replacing words with similar words. Moreover, minor words are added to or removed from the paraphrased text that do not change the meaning. Since such practices are very popular, our paraphrase detection method is effective.

The example paraphrases used in this thesis were incorrectly detect by a state-of-the-art method integrating the eight machine translation metrics [48]. However, they were correctly

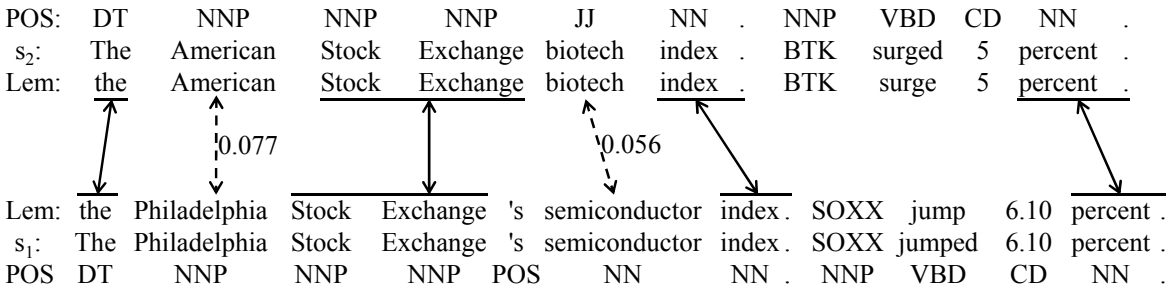


Fig. 5.9 Example of correct identification of non-paraphrased text with proposed combined method.

identified by our proposed combined method. In addition, our proposed method can correctly identify non-paraphrased text, as demonstrated by the example in Fig. 5.9.

In this case, only four identical phrases were matched. Moreover, the length of each identical phrase was small (maximum=2). The similarities of two similar words were very low (0.077 and 0.056). Therefore, this pair is considered to be a non-paraphrase pair.

We combined the proposed metric (*SimMat*) with eight standard MT metrics to create a state-of-the-art method. The results of integrating the eight re-implemented MT metrics (accuracy=76.6%, F-score=83.1%) were lower than with the state-of-the-art method (accuracy=77.4%, F-score=84.1%). This is because we used the default settings of the eight standard libraries. However, changing the setting refers to change their performance. Although of the lower of re-implement, the accuracy of our method (77.6%) was the highest for the most well-known standard MSRP corpus. The F-score (83.9%) was nearly similar with the state-of-the-art approach (84.1%).

5.2 Application

We used the proposed algorithms presented in Chapter 3, Chapter 4, and this chapter to build a web application for controlling the disclosure of information on Facebook. The system can identify private phrases, anonymize ones, and detect disclosure.

5.2.1 Identifying of private phrases

The user accesses the application with his/her existing account and composes a message. The application identifies user's private phrases in the composed message by using co-occurrence metric, as presented in Chapter 3. For example, two private phrases "Tokyo," and "Harvard University" of the composed message "After living in Tokyo, Mary studied at

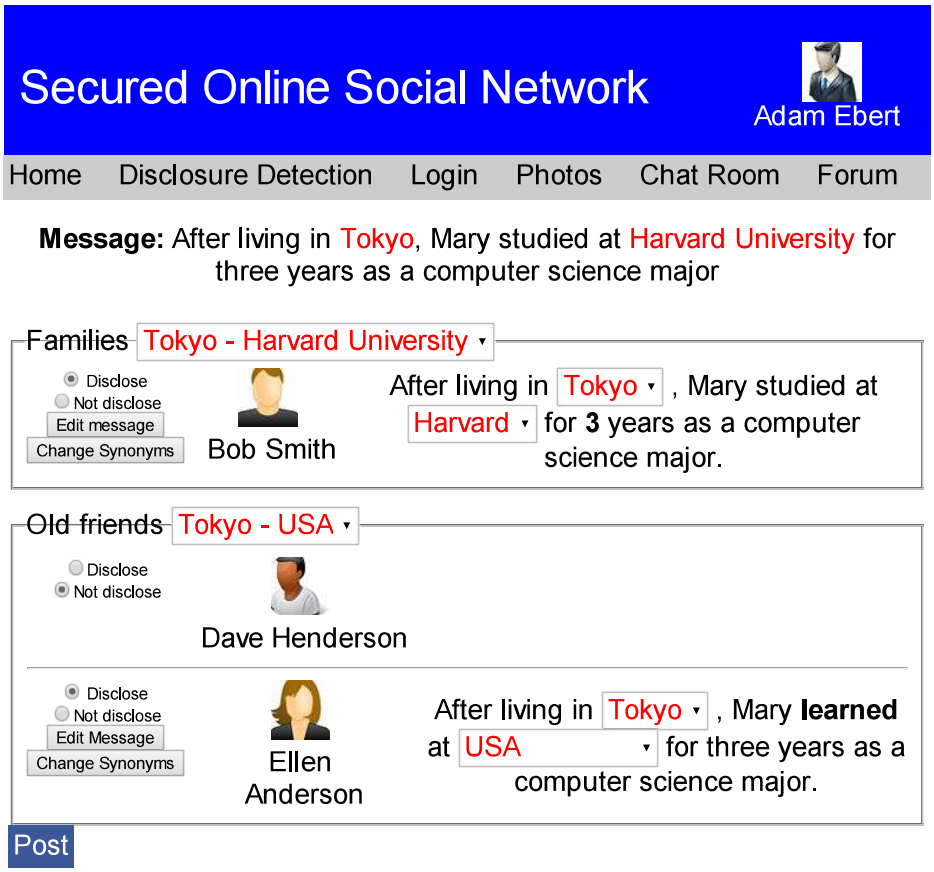


Fig. 5.10 Fingerprinted versions suggested for friends of user "Adam Ebert."

Harvard University for three years as a computer science major" are identified (in red), as shown in Fig. 5.10.

5.2.2 Anonymization of private phrases

The system anonymizes the private phrases by generalizations (Chapter 4) and creates a unique fingerprinted version of the message for each of the user's friends. Fig. 5.10 shows fingerprints suggested for friends of user "Adam Ebert."

The user can revise the fingerprints before the system posts the fingerprinted versions of the message. The user can change the generalizations used for private phrases and can request other synonyms for the fingerprints. The user can also edit other phrases after choosing the generalizations and synonyms. Finally, the user allows the application to post the fingerprinted versions on Facebook. Each friend then sees the appropriate fingerprinted version. Fig. 5.11 and Fig. 5.12 shows the Facebook pages of two friends, "Bob Smith" and "Ellen Anderson," correspondingly who see different fingerprinted versions of the message.

Fig. 5.11 A Facebook page of "Bob Smith."

Fig. 5.12 A Facebook page of "Ellen Anderson."

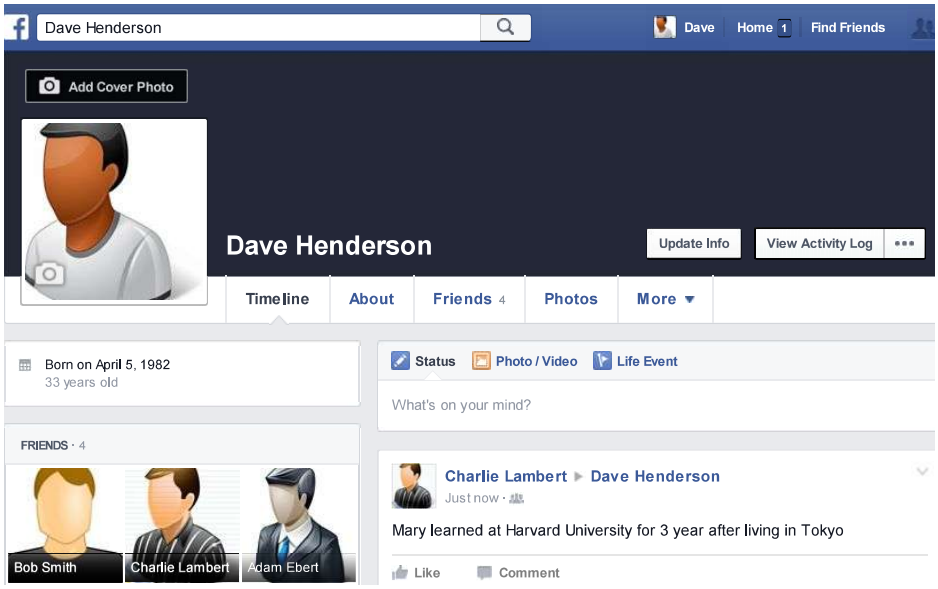


Fig. 5.13 A disclosure page.

5.2.3 Detection of disclosure

If a friend discloses a user's personal information obtained from a message posted by the user on Facebook, as illustrated in Fig. 5.13, the system automatically detects the disclosure using our paraphrase detection and notifies the user, as shown in Fig. 5.14. The example illustrates the need for fingerprinting. In this example, Bob Smith sends Charlie Lambert a copy of Adam Ebert's message via private e-mail. Charlie Lambert then modifies the message in an attempt to avoid detection and posts the modified message on Dave Henderson's wall. However, our system can still detect this disclosure and notify Adam Ebert of the disclosure. Our system thus detects disclosure and identifies the disclosers even if they use other means (such as phone, SMS message, e-mail, etc.) to transfer personal messages.

Testing using the 54,621 personal tweets showed that it takes about 15 seconds to create fingerprints for a message and about 2 seconds to detect whether a message posted by a friend discloses personal information about the user. Our system is thus practical for helping to protect the privacy of OSN users.

5.3 Summary

Our proposed similarity matching (*SimMat*) metric quantifies the similarity between two sentences and can be used to detect whether one is a paraphrase of the other. It is calculated using the matching of identical phrases and similar words. Phrase-by-phrase matching is

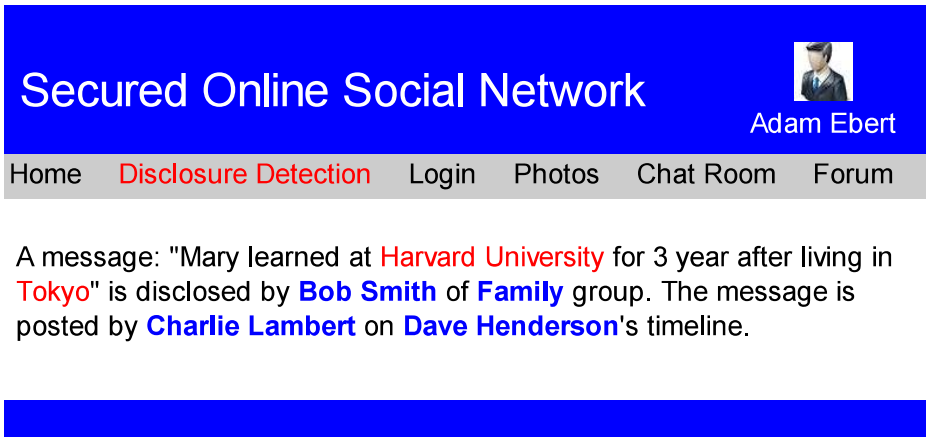


Fig. 5.14 Disclosure detection.

done using a heuristic algorithm that determines the longest duplicate phrase in each iteration. Word matching is done using the Kuhn-Munkres algorithm. WordNet is used for determining the similarity of two words. This similarity is used as the weights for the word-matching algorithm. Minor words, which are often added or removed from paraphrased text to improve naturalness, can create noise when detecting paraphrases. They are thus removed as doing so generally does not change the meaning. A brevity penalty metric is combined with the *SimMat* metric to quantify the effect of inserting and/or deleting words.

Evaluation using the Microsoft Research Paraphrase (MSRP) corpus showed that the *SimMat* metric detects paraphrases more effectively than previous methods. The *SimMat* metric was combined with eight machine translation metrics (MAXSIM, SEPIA, TER, TERp, METEOR, BADGER, BLEU, and NIST). Although the accuracy of the eight re-implemented metrics (accuracy=76.6%, F-score=83.1%) was lower than the published result (accuracy=77.4%, F-score=84.1%), their combination with the *SimMat* metric achieved the best accuracy (77.6%), which was higher than with the state-of-the-art approach (77.4%). Moreover, the F-score of the combination (83.9%) is nearly similar with the-state-of-the-art approach (84.1%). These results show that our method is promising approach to detecting paraphrasing.

The strength of our method is that it is based on the common practices people use to avoid paraphrase detection. These practices include cutting and pasting phrases and replacing words with similar words. They also include adding and removing minor words that do not change the meaning. These practices are really popular. Therefore, our method can correctly handle pairs that are misclassified by the state-of-the-art approach.

Future work includes quantifying the weights of words in matched phrases, determining the effect of a word's position in a sentence, and analyzing misclassified pairs to improve performance.

We demonstrate the practicality of our proposed algorithms in this thesis by creating a web application to control the posting of user messages on Facebook. After the user composes a message, the application suggests differently fingerprinted versions of the message for the user's different groups of friends. After the user accepts and/or modifies the different versions, the application posts them on Facebook. As a result, the user's friends see different versions of the message depending on the group of friends to which they belong. If any personal information about the user is disclosed on Facebook, the application detects the disclosure, identifies the friend responsible, and notifies the user of the disclosure and the person responsible.

Chapter 6

Conclusion and future work

6.1 Conclusion

Online social networks (OSNs) play a significant role in modern life. They help people to communicate more easily and quickly with each other. However, many users worry about their private messages being unintentionally or intentionally disclosed on the Internet by their friends or even by themselves. Therefore, our first objective is to identify private phrases in private messages. Our second objective is to anonymize the private phrases before they are posted on an OSN. And our third objective is to detect disclosers of private information and to notify the information owner.

6.1.1 Identification of private phrases

To enhance the privacy of private information in user profiles, we developed a method for identifying private phrases in private messages. The identification is done by comparing each noun phrase in a message with each phrase in the user's profile. The comparison is done using a co-occurrence metric [38]. The metric was calculated using the Wikipedia corpus, a huge corpus with high-quality information that is free to use while other corpora have limited queries (such as Google, Bing, and Yahoo). We also estimated a threshold for the co-occurrence metric. The threshold is the balance of the identification rate (precision=recall=76.39%) that ensures the best identification of private phrases.

Phrases indicating the user's location are a common type of private phrase. We developed a rule-based method for identifying private locational phrases in OSN messages. The rule is created by analyzing the relationships between the user's phrases and the locational phrases in the messages. This method discovers most private locational phrases in OSN messages. Its

accuracy of 84.95% is meaningfully better than those of four approaches based on machine learning.

6.1.2 Anonymization of private phrases

We also developed a method for anonymizing the private phrases by using generalizations. A metric is used to quantify the information loss due to each generalization to ensure that each group of the user's friends receives a version of the message with an appropriate level of privacy. The metric is calculated on the basis of the populations of each generalization. The populations are directly retrieved from the Info-box of Wikipedia. Comparison of the metric with previous metrics showed that it is the strongest.

Temporal information is a significant type of private information. We propose a method for creating anonymous fingerprings about temporal phrases to cover most of potential cases of OSN disclosure. The fingerprints not only anonymize temporal-related information but also can be used to identify a person who has disclosed information about the user. Moreover, we develop another method that can be used to anonymize time-related private information by removing the private temporal phrases. It makes sure that the private phrases can generally be deleted without damaging the grammatical structure of the sentences.

6.1.3 Detection of disclosure

Finally, we proposed a method using fingerprints to detect paraphrases. This method is used to detect disclosures of a user's private information. Its accuracy (77.6%) is slightly higher than that of the state-of-the-art method (77.4%).

We built a secure OSN to help users post private messages on Facebook. The system detects and anonymizes private phrases in the user's posts and helps user detect disclosure of their private information and identify the discloser.

6.2 Strength and limitation

Our methods can be easily applied to other languages because they focus only on lexical words in the messages. Moreover, they can be applied not only to OSNs but also to other areas (such as politics, news, and health). Testing showed that their results are better than those of state-of-the-art methods or quality baselines. Careful analysis of the privacy obtained with our methods showed their consistent with multiple attacks. The application we built for applying these methods to messages to be posted on Facebook showed that they are practical.

Our methods analyze the sensitiveness of a phrase in a message. However, the sensitiveness is probably inferred by combining information in multiple phrases. Attackers can gather information about a user on the Internet (such as the user's previous messages, homepage, and even messages sent by friends) and use it to infer private information. With hiding private information, a few anonymous messages are unnatural so that adversaries concern about that for attacking them. Our methods could not be used for standard formal messages (e.g., poetry, song lyrics).

6.3 Future work

We will work on overcoming the complex problems of private information that is inferred from various phrases or sources. We will also improve the naturalness of the anonymous fingerprints and will extend our methods so that they can be used for hiding private information in other types of digital media (such as images, videos, and audio). Future work also includes helping users manage their private information in different posts on the Internet.

References

[1] Ahmed Abbasi and Hsinchun Chen. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. ACM Transactions on Information Systems, 26(2):7:1–7:29, 2008.

[2] Alessandro Acquisti and Ralph Gross. Predicting social security numbers from public data. Proceedings of the National Academy of Sciences, 106(27):10975–10980, 2009.

[3] Timo Ahonen, Esa Rahtu, Ville Ojansivu, and J Heikkila. Recognition of blurred faces using local phase quantization. In Proceedings of the 19th International Conference on Pattern Recognition, pages 1–4. Institute of Electrical and Electronics Engineers, 2008.

[4] Einat Amitay, Nadav Har'El, Ron Sivan, and Aya Soffer. Web-a-where: geotagging web content. In Proceedings of the 27th ACM International Conference on Research and Development in Information Retrieval, pages 273–280. Association for Computing Machinery, 2004.

[5] Mikhail J Atallah, Victor Raskin, Christian F Hempelmann, Mercan Karahan, Radu Sion, Umut Topkara, and Katrina E Triezenberg. Natural language watermarking and tamperproofing. In Proceedings of the Information Hiding, pages 196–212. Springer Berlin Heidelberg, 2003.

[6] Mukhtaj S Barhm, Nidal Qwasmi, Faisal Z Qureshi, and Khalil El-Khatib. Negotiating privacy preferences in video surveillance systems. In Proceedings of the Modern Approaches in Applied Intelligence, pages 511–521. Springer Berlin Heidelberg, 2011.

[7] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In Proceedings of the 21st International Conference on Data Engineering, pages 217–228, 2005.

[8] William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 546–556. Association for Computational Linguistics, 2012.

[9] Thach V Bui, Binh Q Nguyen, Thuc D Nguyen, Noboru Sonehara, and Isao Echizen. Robust fingerprinting codes for database. In Proceedings of the Algorithms and Architectures for Parallel Processing, pages 167–176. Springer Berlin Heidelberg, 2013.

[10] Ji-Won Byun and Elisa Bertino. Micro-views, or on how to protect privacy while enhancing data usability: concepts and challenges. ACM Special Interest Group on Management of Data Conference Record, 35(1):9–13, 2006.

[11] Venkatesan T Chakaravarthy, Himanshu Gupta, Prasan Roy, and Mukesh K Mohania. Efficient techniques for document sanitization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, pages 843–852. Association for Computing Machinery, 2008.

[12] Yee Seng Chan and Hwee Tou Ng. Maxsim: A maximum similarity metric for machine translation evaluation. In Proceedings of the Association of Computational Linguistics, pages 55–62. Association for Computational Linguistics, 2008.

[13] Angel X. Chang and Christopher Manning. Sutime: A library for recognizing and normalizing time expressions. In Proceedings of the 8th International Conference on Language Resources and Evaluation, pages 3735–3740. European Language Resources Association, 2012.

[14] Ching-Yun Chang and Stephen Clark. Linguistic steganography using automatically generated paraphrases. In Proceedings of the Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 591–599. Association for Computational Linguistics, 2010.

[15] Ching-Yun Chang and Stephen Clark. Practical linguistic steganography using contextual synonym substitution and vertex colour coding. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1194–1203. Association for Computational Linguistics, 2010.

[16] Ching-Yun Chang and Stephen Clark. Adjective deletion for linguistic steganography and secret sharing. In Proceedings of the 24th International Conference on Computational Linguistics, pages 493–510, 2012.

[17] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pages 759–768. Association for Computing Machinery, 2010.

[18] Adrian Dabrowski, Edgar R Weippl, and Isao Echizen. Framework based on privacy policy hiding for preventing unauthorized face image processing. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics, pages 455–461, 2013.

[19] Dipanjan Das and Noah A Smith. Paraphrase identification as probabilistic quasi-synchronous recognition. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing, pages 468–476. Association for Computational Linguistics, 2009.

[20] Michael Denkowski and Alon Lavie. Extending the meteor machine translation evaluation metric to the phrase level. In Proceedings of the Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 250–253. Association for Computational Linguistics, 2010.

[21] Peter DeScioli, Robert Kurzban, Elizabeth N Koch, and David Liben-Nowell. Best friends alliances, friend ranking, and the myspace social network. Perspectives on Psychological Science, 6(1):6–8, 2011.

[22] EU Directive. 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Official Journal of the EC, 23(6), 1995.

[23] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proceedings of the 2nd International Conference on Human Language Technology Research, pages 138–145. Morgan Kaufmann Publishers Inc., 2002.

[24] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In Proceedings of the 20th International Conference on Computational Linguistics, page 350. Association for Computational Linguistics, 2004.

[25] Nicole B Ellison, Charles Steinfield, and Cliff Lampe. The benefits of facebook "friends:" social capital and college students' use of online social network sites. Journal of Computer-Mediated Communication, 12(4):1143–1168, 2007.

[26] Andrew Finch, Young-Sook Hwang, and Eiichiro Sumita. Using machine translation evaluation techniques to determine sentence-level semantic equivalence. In Proceedings of the Third International Workshop on Paraphrasing, pages 17–24, 2005.

[27] Clayton Fink, Christine D Piatko, James Mayfield, Tim Finin, and Justin Martineau. Geolocating blogs from their textual content. In Proceedings of the National Conference of the American Association for Artificial Intelligence, pages 25–26, 2009.

[28] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 363–370. Association for Computational Linguistics, 2005.

[29] Benjamin Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. ACM Computing Surveys, 42(4):1–53, 2010.

[30] Ralph Gross and Alessandro Acquisti. Information revelation and privacy in online social networks. In Proceedings of the ACM workshop on Privacy in the Electronic Society, pages 71–80. Association for Computing Machinery, 2005.

[31] Nizar Habash and Ahmed Elkholy. Sepia: surface span extension to syntactic dependency precision-based mt evaluation. In Proceedings of the NIST metrics for machine translation workshop at the association for machine translation, 2008.

[32] Bo Han, Paul Cook, and Timothy Baldwin. Automatically constructing a normalisation dictionary for microblogs. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 421–432. Association for Computational Linguistics, 2012.

[33] Bo Han, Paul Cook, and Timothy Baldwin. Automatically constructing a normalisation dictionary for microblogs. In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 421–432. Association for Computational Linguistics, 2012.

[34] A Harvey. Cv dazzle: Camouflage from computer vision, 2012.

[35] Anh-Tu Hoang, Hoang-Quoc Nguyen-Son, Minh-Triet Tran, and Isao Echizen. Detecting traitors in re-publishing updated datasets. In Proceedings of the Digital Forensics and Watermarking, pages 205–220. Springer Berlin Heidelberg, 2014.

[36] Vijay S. Iyengar. Transforming data to satisfy privacy constraints. In Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining, pages 279–288, 2002.

[37] Daniel Jurafsky and H James. Speech and language processing an introduction to natural language processing, computational linguistics, and speech, chapter 4, pages 83–122. Pearson Education, 2009.

[38] H. Kataoka, A. Utsumi, Y. Hirose, and H. Yoshiura. Disclosure control of natural language information to enable secure and enjoyable communication over the internet. In Proceedings of the 15th International Workshop on Security Protocols, volume 5964, pages 178–188, 2010.

[39] Peter Kieseberg, Sebastian Schrittwieser, Martin Mulazzani, Isao Echizen, and Edgar Weippl. An algorithm for collusion-resistant anonymization and fingerprinting of sensitive microdata. Electronic Markets, 24(2):113–124, 2014.

[40] Asanobu Kitamoto and Takeshi Sagara. Toponym-based geotagging for observing precipitation from social and scientific data streams. In Proceedings of the ACM Multimedia Workshop on Geotagging and Its Applications in Multimedia, pages 23–26. Association for Computing Machinery, 2012.

[41] Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pages 423–430. Association for Computational Linguistics, 2003.

[42] Dimitrios Kokkinakis and Anders Thurin. Anonymisation of swedish clinical data. In Proceedings of the Artificial Intelligence in Medicine, pages 237–241. Springer Berlin Heidelberg, 2007.

[43] Harold W Kuhn. The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97, 1955.

[44] Ieng-Fat Lam, Kuan-Ta Chen, and Ling-Jyh Chen. Involuntary information leakage in social network services. In Proceedings of the Advances in Information and Computer Security, pages 167–183. Springer Berlin Heidelberg, 2008.

[45] Michael Lebowitz. Generalization from natural language text*. Cognitive Science, 7 (1):1–40, 1983.

[46] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. Recognizing named entities in tweets. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 359–367. Association for Computational Linguistics, 2011.

[47] Shimon Machida, Shigeru Shimada, and Isao Echizen. Settings of access control by detecting privacy leaks in sns. In Proceedings of the 9th on Signal Image Technology and Internet Based Sytems, pages 660–666, 2013.

[48] Nitin Madnani, Joel Tetreault, and Martin Chodorow. Re-examining machine translation metrics for paraphrase identification. In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 182–190. Association for Computational Linguistics, 2012.

[49] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60, 2014.

[50] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, pages 55–60, 2014.

[51] Ben Medlock. An introduction to nlp-based textual anonymisation. In Proceedings of the 5th International Conference on Language Resources and Evaluation, 2006.

[52] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In Proceedings of the National Conference of the American Association for Artificial Intelligence, volume 6, pages 775–780, 2006.

[53] George A Miller. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41, 1995.

[54] James Munkres. Algorithms for the assignment and transportation problems. Journal of the Society for Industrial & Applied Mathematics, 5(1):32–38, 1957.

[55] Tran Hong Ngoc, Isao Echizen, Kamiyama Komei, and Hiroshi Yoshiura. New approach to quantification of privacy on social network sites. In Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, pages 556–564. Institute of Electrical and Electronics Engineers, 2010.

[56] Hoang-Quoc Nguyen-Son, Quoc-Binh Nguyen, Minh-Triet Tran, Dinh-Thuc Nguyen, Hiroshi Yoshiura, and Isao Echizen. New approach to anonymity of user information on social networking services. In The 6th International Symposium on Digital Forensics and Information Security Proceedings of the 7th FTRA International Conference on Future Information Technology, pages 731–739. Springer Berlin Heidelberg, 2012.

[57] Hoang-Quoc Nguyen-Son, Quoc-Binh Nguyen, Minh-Triet Tran, Dinh-Thuc Nguyen, Hiroshi Yoshiura, and Isao Echizen. Automatic anonymization of natural languages texts posted on social networking services and automatic detection of disclosure. In the 7th International Workshop on Frontiers in Availability, Reliability and Security Proceedings of the International Conference on Availability, Reliability and Security, pages 358–364. Institute of Electrical and Electronics Engineers, 2012.

[58] Hoang-Quoc Nguyen-Son, Minh-Triet Tran, Tien-Dung Tran, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen. Automatic anonymization of natural languages texts posted on social networking services and automatic detection of disclosure. In Proceedings of the 11th International Workshop on Digital-Forensics and Watermarking, pages 731–739. Springer Berlin Heidelberg, 2012.

[59] Hoang-Quoc Nguyen-Son, Tran Minh-Triet, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen. Anonymizing personal text messages posted in online social networks and detecting disclosures of personal information. IEICE Transactions on Information and Systems, 98(1):78–88, 2015.

[60] Taichi Noro, Takashi Inui, Hiroya Takamura, and Manabu Okumura. Time period identification of events in text. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pages 1153–1160. Association for Computational Linguistics, 2006.

[61] Iadh Ounis, Craig Macdonald, Jimmy Lin, and Ian Soboroff. Overview of the trec-2011 microblog track. In Proceeddings of the 20th Text REtrieval Conference, pages 1–5, 2011.

[62] Frank Pallas, Max-Robert Ulbricht, Lorena Jaume-Palasí, and Ulrike Höppner. Offline-tags: a novel privacy approach to online photo sharing. In Proceedings of the Extended Abstracts on Human Factors in Computing Systems, pages 2179–2184. Association for Computing Machinery, 2014.

[63] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 311–318. Association for Computational Linguistics, 2002.

[64] Steven Parker. Badger: A new machine translation metric. In Proceedings of the Metrics for Machine Translation Challenge, 2008.

[65] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In Demonstration papers at Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies, pages 38–41. Association for Computational Linguistics, 2004.

[66] John Platt et al. Fast training of support vector machines using sequential minimal optimization. Advances in kernel methods—support vector learning, 3:185–208, 1999.

[67] Long Qiu, Min-Yen Kan, and Tat-Seng Chua. Paraphrase recognition via dissimilarity significance classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 18–26. Association for Computational Linguistics, 2006.

[68] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1524–1534. Association for Computational Linguistics, 2011.

[69] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of database systems, pages 188–188. Association for Computing Machinery, 1998.

[70] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In Proceedings of the ACM SIGMOD-SIGACT-SIGART Conference on Principles of Database Systems, volume 98, pages 188–201, 1998.

[71] David Sanchez, Montserrat Batet, and Alexandre Viejo. Automatic general-purpose sanitization of textual documents. IEEE Transactions on Information Forensics and Security, 8(6):853–862, 2013.

[72] Jeremy Schiff, Marci Meingast, Deirdre K Mulligan, Shankar Sastry, and Ken Goldberg. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In Proceedings of the Protecting Privacy in Video Surveillance, pages 65–89. Springer Berlin Heidelberg, 2009.

[73] Sebastian Schrittwieser, Peter Kieseberg, Isao Echizen, Sven Wohlgemuth, Noboru Sonehara, and Edgar Weippl. An algorithm for k-anonymity-based fingerprinting. In Proceedings of the Digital Forensics and Watermarking, pages 439–452. Springer Berlin Heidelberg, 2012.

[74] Nakatani Shuyo. Language Detection Library for Java. http://code.google.com/p/language-detection/, 2010.

[75] Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In Proceedings of Association for Machine Translation in the Americas, pages 223–231, 2006.

[76] Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. Machine Translation, 23(2-3):117–127, 2009.

[77] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Proceedings of the Advances in Neural Information Processing Systems, pages 801–809, 2011.

[78] Jannik Strötgen and Michael Gertz. Heideltime: High quality rule-based extraction and normalization of temporal expressions. In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 321–324. Association for Computational Linguistics, 2010.

[79] Fred Stutzman, Ralph Gross, and Alessandro Acquisti. Silent listeners: The evolution of privacy and disclosure on facebook. Journal of privacy and confidentiality, 4(2): 7–41, 2013.

[80] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In Proceedings of the 16th international conference on World Wide Web, pages 697–706. Association for Computing Machinery, 2007.

[81] Mercan Topkara, Umut Topkara, and Mikhail J Atallah. Words are not enough: sentence level natural language watermarking. In Proceedings of the 4th ACM International Workshop on Contents Protection and Security, pages 37–46. Association for Computing Machinery, 2006.

[82] Naushad UzZaman and James F. Allen. Event and temporal expression extraction from raw text: first step towards a temporally aware system. International Journal of Semantic Computing, 4(4):487–508, 2010.

[83] Jian Xu, Wei Wang, Jian Pei, Xiaoyuan Wang, Baile Shi, and Ada Wai-Chee Fu. Utility-based anonymization using local recoding. In Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining, pages 785–790, 2006.

[84] Takayuki Yamada, Seiichi Gohshi, and Isao Echizen. Countermeasure of re-recording prevention against attack with short wavelength pass filter. In Proceedings of the 18th International Conference on Image Processing, pages 2753–2756. Institute of Electrical and Electronics Engineers, 2011.

[85] Takayuki Yamada, Seiichi Gohshi, and Isao Echizen. icabinet: Stand-alone implementation of a method for preventing illegal recording of displayed content by adding invisible noise signals. In Proceedings of the 19th ACM international conference on Multimedia, pages 771–772. Association for Computing Machinery, 2011.

[86] Takayuki Yamada, Seiichi Gohshi, and Isao Echizen. Use of invisible noise signals to prevent privacy invasion through face recognition from camera images. In Proceedings of the 20th ACM international conference on Multimedia, pages 1315–1316. Association for Computing Machinery, 2012.

[87] Takayuki Yamada, Seiichi Gohshi, and Isao Echizen. Privacy visor: Method for preventing face image detection by using differences in human and device sensitivity. In Proceedings of the Communications and Multimedia Security, pages 152–161. Springer Berlin Heidelberg, 2013.

[88] Yue Zhang, Graeme Blackwood, and Stephen Clark. Syntax-based word ordering incorporating a large-scale language model. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pages 736–746. Association for Computational Linguistics, 2012.

[89] Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. Journal of the American Society for Information Science and Technology, 57(3):378–393, 2006.

[90] Xueling Zheng, Liusheng Huang, Zhili Chen, Zhenshan Yu, and Wei Yang. Hiding information by context-based synonym substitution. In Proceedings of the Digital Forensics and Watermarking, pages 162–169. Springer Berlin Heidelberg, 2009.

Publication

Journals

1. Hoang-Quoc Nguyen-Son, Minh-Triet Tran, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen, "Anonymizing Personal Text Messages Posted in Online Social Networks and Detecting Disclosures of Personal Information," *IEICE Transactions on Information and Systems*, Volume 98, Issue 1, pp.78-88 (January 2015)

Conferences

1. Hoang-Quoc Nguyen-Son, Yusuke Miyao, and Isao Echizen, "Paraphrase Detection Based on Identical Phrase and Similar Word Matching," *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pp. 508-516, ACL, China, (October 2015)

2. Hoang-Quoc Nguyen-Son, Minh-Triet Tran, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen, "A Rule-Based Approach for Detecting Location Leaks of Short Text Messages," Workshop on Privacy by Transparency in Data-Centric Services (PTDCS), *Proceedings of the 18th International Conference on Business Information Systems (BIS)*, pp. 1-12, LNCS, Springer, Poland, (June 2015)

3. Hoang-Quoc Nguyen-Son, Minh-Triet Tran, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen, "A System for Anonymizing Temporal Phrases of Message Posted in Online Social Networks and for Detecting Disclosure," the 4th International Workshop on Resilience and IT-Risk in Social Infrastructures (RISI), *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, pp. 455-460, Switzerland, (September 2014)

4. Hoang-Quoc Nguyen-Son, Anh-Tu Hoang, Minh-Triet Tran, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen, "Anonymizing Temporal Phrases in Natural Language

Text to be Posted on Social Networking Services," *Proceedings of the 12th International Workshop on Digital-Forensics and Watermarking (IWDW)*, LNCS, pp. 437-451 , New Zealand (October 2013)

5. Anh-Tu Hoang, Hoang-Quoc Nguyen-Son, Minh-Triet Tran and Isao Echizen, "Detecting Traitors in Re-Publishing Updated Datasets," *Proceedings of the 12th International Workshop on Digital-Forensics and Watermarking (IWDW)*, LNCS, pp. 205-220, New Zealand (October 2013)

6. Hoang-Quoc Nguyen-Son, Minh-Triet Tran, Tien-Dung Tran, Hiroshi Yoshiura, Sonehara Noboru, and Isao Echizen, "Automatic Anonymous Fingerprinting of Text Posted on Social Networking Services," *Proceedings of the 11th International Workshop on Digital-Forensics and Watermarking (IWDW)*, LNCS, pp. 410-424, China (October 2012)

7. Hoang-Quoc Nguyen-Son, Quoc-Binh Nguyen, Minh-Triet Tran, Dinh-Thuc Nguyen, Hiroshi Yoshiura and Isao Echizen, "Automatic Anonymization of Natural Languages Texts Posted on Social Networking Services and Automatic Detection of Disclosure" the 7th International Workshop on Frontiers in Availability, Reliability and Security (FARES), *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, Czech Republic, pp. 358-364 (August 2012)

8. Hoang-Quoc Nguyen-Son, Quoc-Binh Nguyen, Minh-Triet Tran, Dinh-Thuc Nguyen, Hiroshi Yoshiura and Isao Echizen, "New Approach to Anonymity of User Information on Social Networking Services", The 6th International Symposium on Digital Forensics and Information Security (DFIS), *Proceedings of the 7th FTRA International Conference on Future Information Technology (FutureTech)*, Lecture Notes in Electrical Engineering, vol. 164, pp. 731-739, Canada (June 2012)

9. Nguyen Son Hoang Quoc, Le Thanh Tam, Nguyen Hoang Long, Tran Minh Triet, "Digital identity management system with identity information stored in mobile devices," *Proceedings of the 3th Information and Communication Technology in Faculty of Information (ICTFIT)*, pp. 81-90, Vietnam (November 2010).

Presentations

1. Hoang-Quoc Nguyen-Son, Hiroshi Yoshiura Noboru Sonehara, and Isao Echizen, "A Recommendation System for Anonymous Fingerprinting of Text Posted on Social

Networks", The Enriched Multimedia (EMM), *IEICE Technical Report*, vol. 113, no. 66, pp. 31-36, Japan (May 2013)

2. Hoang-Quoc Nguyen-Son, Quoc-Binh Nguyen, Minh-Triet Tran, Dinh-Thuc Nguyen, Hiroshi Yoshiura and Isao Echizen, "Automatic Anonymous Fingerprinting of Text Posted on Social Networking Services", *the 29th Symposium on Cryptography and Information Security (SCIS)*, Japan (February 2012)

3. Hoang-Quoc Nguyen-Son and Isao Echizen, "Automatic Anonymous Fingerprinting of Text Posted on Online Social Networking," *the Information System for Social Innovation (ISSI)*, pp. 53-58, Japan (February 2013)

4. Hoang-Quoc Nguyen-Son and Isao Echizen, "Automatic Anonymous Fingerprinting of Text Posted on Online Social Networking," *Proceedings of the Information System for Social Innovation (ISSI)*, pp. 29-30, Japan (February 2014)

Posters

1. Hoang-Quoc Nguyen-Son, "Automatic Anonymous Fingerprinting of Text Posted on Social Networking Services, " *the International Symposium on Global Knowledge Circulation – Designing Integrated Social Infrastructure*, pp. 30, Japan (December, 2012)

2. Hoang-Quoc Nguyen-Son and Isao Echizen, "A method for Anonymizing Sensitive Information about a User and for Detecting Revelations on Social Networking Sites," *National Institute of Informatics OPEN HOUSE*, Japan (June, 2013)

3. Hoang-Quoc Nguyen-Son, Anh-Tu Hoang, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen, "Anonymous Sensitive Information of Natural Language Text Posted on Social Networks and Detection of Disclosure," *the Big Data x Big Brother Workshop 2013*, Japan (July, 2013)

4. Hoang-Quoc Nguyen-Son, Hiroshi Yoshiura, Noboru Sonehara, and Isao Echizen, "A System for Anonymizing Personal Text Messages Posted in Online Social Networks and for Detecting Disclosures," *Information System for Social Innovation (ISSI)*, Japan (February, 2015)