IEICE TRANS. INF. & SYST., VOL.E98–D, NO.1 JANUARY 2015

PAPER Special Section on Enriched Multimedia

Anonymizing Personal Text Messages Posted in Online Social Networks and Detecting Disclosures of Personal Information

Hoang-Quoc NGUYEN-SON[†a)], **Minh-Triet TRAN**[††], ***Nonmembers***, **Hiroshi YOSHIURA**[†††], **Noboru SONEHARA**[††††], ***and*** **Isao ECHIZEN**[†,††††], ***Members***

SUMMARY While online social networking is a popular way for people to share information, it carries the risk of unintentionally disclosing personal information. One way to reduce this risk is to anonymize personal information in messages before they are posted. Furthermore, if personal information is somehow disclosed, the person who disclosed it should be identifiable. Several methods developed for anonymizing personal information in natural language text simply remove sensitive phrases, making the anonymized text message unnatural. Other methods change the message by using synonymization or structural alteration to create fingerprints for detecting disclosure, but they do not support the creation of a sufficient number of fingerprints for friends of an online social network user. We have developed a system for anonymizing personal information in text messages that generalizes sensitive phrases. It also creates a sufficient number of fingerprints of a message by using synonyms so that, if personal information is revealed online, the person who revealed it can be identified. A distribution metric is used to ensure that the degree of anonymization is appropriate for each group of friends. A threshold is used to improve the naturalness of the fingerprinted messages so that they do not catch the attention of attackers. Evaluation using about 55,000 personal tweets in English demonstrated that our system creates sufficiently natural fingerprinted messages for friends and groups of friends. The practicality of the system was demonstrated by creating a web application for controlling messages posted on Facebook.

key words: *fingerprint, anonymized text message, disclosure detection, online social network*

1. Introduction

Users often share information through online social networks (OSNs) (such as Facebook, Twitter, and Google+). However, they or their friends often disclose the users' personal information, both intentionally and unintentionally. For example, Stutzman et al. examined 5,076 Facebook accounts and were able to identify the real name for 89% of them, the birthday for 88%, and the current residence for 51% [1]. Such personal information could be used to determine a user's social security number*. Therefore, users may feel unsafe when sharing personal information in an OSN. The risk of such revelations can be reduced by anonymizing the personal information in messages before they are posted. Moreover, if personal information is somehow disclosed, the person who disclosed it should be identifiable.

Most previous methods for anonymizing personal information in natural language text simply remove sensitive phrases [2]. Others replace them with an appropriate categorical word or phrase (*location*, *person*, *organization*, etc.) [3]. We previously reported a method that uses generalizations of sensitive phrases to anonymize personal information [4]. However, sensitive words and phrases, such as "*flu*" and "*H5N1*" in the general message "Unlike other types of *flu*, *H5N1* usually does not spread between people" are also anonymized although this general message does not disclose personal information. Anonymization of general messages should be avoided.

One way to identify a person who has disclosed personal information is to "fingerprint" 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 sensitive information. Messages can be fingerprinted, for example, by reordering their structure [5], using paraphrasing [6], or by synonymizing [7]. However, these methods cannot create a sufficient number of unique fingerprints for all the friends of a typical OSN user.

Our contributions in this paper are as follows:

- We report a system for anonymizing personal information and detecting disclosure in OSNs. The system anonymizes the personal information by generalizing sensitive 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 disclosures.
- 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 co-occurrence metric used to detect sensitive phrases. This prevents attackers from replacing sensitive phrases with similar phrases to avoid detection.
- We identify a classifier to determine whether a composed message is either a personal message or a general message to ensure that only personal messages are anonymized.

Manuscript received March 30, 2014.
Manuscript revised August 12, 2014.

[†]The authors are with the Graduate University for Advanced Studies, Kanagawa-ken, 240–0193 Japan.
[††]The author is with University of Science, Hochiminh, Vietnam.
[†††]The author is with The University of Electro-Communications, Chofu-shi, 182–8585 Japan.
[††††]The authors are with National Institute of Informatics, Tokyo, 101–8430 Japan.
a) E-mail: nshquoc@nii.ac.jp

DOI: 10.1587/transinf.2014MUP0016

*https://medium.com/cyber-security/24eb09e026dd

- 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 system creates a sufficient number of fingerprints for friends and that the fingerprinted messages are natural.

We evaluated our system by using about 16 million tweets taken from the TREC Tweets2011 Dataset [8]. From them we extracted about 55,000 personal messages in English that contained sensitive phrases falling into seven attribute types (hometown, education, work, religion, politics, sports, and personal interests). These are the main attributes common to major OSNs (e.g., Facebook and Google+). The system created an average of 16.59 generalizations and 140.91 fingerprints per tweet. This demonstrates the practicality of our system given that the average number of friends per user in Facebook, the largest OSN, at the beginning of 2014 was 130[†]. Moreover, the number of fingerprints is significantly higher than that of a state-of-the-art fingerprint algorithm using only synonyms [7] (21.29 on average).

We demonstrated the practicality of our system 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.

This paper is organized as follows: Sect. 2 describes related work. Section 3 presents our proposed system. Section 4 shows the web application created using the system, and Sect. 5 describes our evaluation. Section 6 discusses the results, and Sect. 7 summarizes the key points and mentions future work.

2. Related Work

The two main objectives of the system reported in this paper are to anonymize personal information and detect disclosure of personal information.

2.1 Anonymizing Personal Information

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 sensitive phrases [2]. Others replace sensitive information with appropriate categorical words or phrases (such as *location*, *person*, and *organization*) [3]. These methods use name entity recognition to detect entities in messages and anonymize them. However, some OSN messages containing sensitive phrases do not disclose personal information about the user. For example, the general message "*Tokyo* is the capital of *Japan*" does not disclose personal information. Therefore, we classify a message containing sensitive phrases as either a personal message or a general message. *Personal messages* disclose personal information about the user while *general messages* do not. We anonymize sensitive phrases only in personal messages.

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

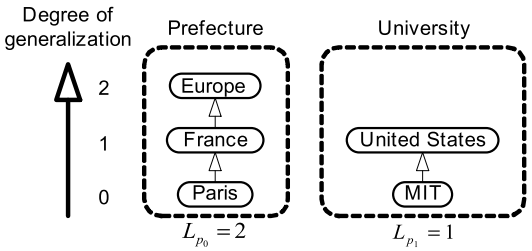

Fig. 1 Generalization schemes for two quasi-identifiers.

Our previous method anonymizes sensitive information by generalization [4], but the anonymized messages are unnatural. Attackers recognize such unnaturalness and focus on changing sensitive information to avoid disclosure detection. We have improved the naturalness of anonymized messages by using a frequency metric.

2.2 Metric for Quantifying Loss due to Anonymization

Anonymizing sensitive phrases by generalization results in information loss. Generalization schemes for two quasi-identifiers, "Prefecture" and "University," are diagrammed in Fig. 1. 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 sensitive phrases $P = \{p_i\}$ is to use the Samarati metric (Sam) [9], which is based on the degree of generalization of the i-th phrase l_{p_i}, as shown in Eq. 1. For example, if the message contains two sensitive 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."

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

The precision metric (Pre) [9] 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 sensi-

tive phrase p_i, as shown in Eq. 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 [10]. Therefore, "MIT" and "Paris" should have different metric values for information loss.

$$Pre(P) = \sum_{i=0}^{N-1} \frac{l_{p_i}}{l_{P_i}} \tag{2}$$

Ngoc et al. proposed a metric based on probability and entropy [11] 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†. However, this metric simply quantifies information loss within the scope of universities in Japan. A metric is needed that automatically quantifies all sensitive phrases on the basis of actual data.

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

$$Dis(P) = \sum_{i=0}^{N-1} \frac{log(|p_i|)}{log(|P_i|)}, \tag{3}$$

where $|x|$ is the population of sensitive phrase x. *Dis* is explained in more detail in the next section.

2.3 Detecting Disclosure of Personal Information

Most methods for detecting disclosure of personal information use fingerprinting. Fingerprinting a message differently for each friend receiving it enables the person who discloses sensitive 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) [12]. Others use semantic transformation based on word sense disambiguation, semantic role parsing, or anaphora resolution to create fingerprints [13]. The payload of each method is about 0.5 fingerprints per message. The method proposed by Zheng et al. [7] 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.

Our system uses the best generalizations of sensitive phrases so that the generalized message is more natural. It creates 140.91 fingerprints on average by using the best synonyms for some phrases in the generalized message, a sufficient number for most OSN users.

†http://www3.ibac.co.jp/univ1/mst/info/univinfo_50.jsp

3. Proposed System

Our proposed system has two main processes: anonymization of personal information and detection of disclosure, as illustrated in Fig. 2, which shows the case of a user with m friends in n groups.

In the anonymization of personal information process, the system receives an input message t (such as a blog, comment, or status) from the user. The system then automatically anonymizes the personal information, creates differently fingerprinted versions of the message $F = \{f_0^{(0)}, f_1^{(0)}, \ldots, f_{m-1}^{(n-1)}\}$, and suggests one for each group of friends. The user then accepts and/or revises the different versions, and confirms for the system the final fingerprinted messages $\bar{F} = \{\bar{f}_0^{(0)}, \bar{f}_1^{(0)}, \ldots, \bar{f}_{m-1}^{(n-1)}\}$ to be posted. Finally, the system posts the fingerprinted messages so that each friend sees the appropriate version.

In the detection of disclosure process, if any of the user's friends discloses the user's personal information, the system analyzes the fingerprint to identify the discloser and sends a notification to the user.

3.1 Anonymization of Personal Information

Throughout this paper, 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.*" Algorithm 1 describes the steps in the anonymization of personal information process. The following subsections explain each function in detail.

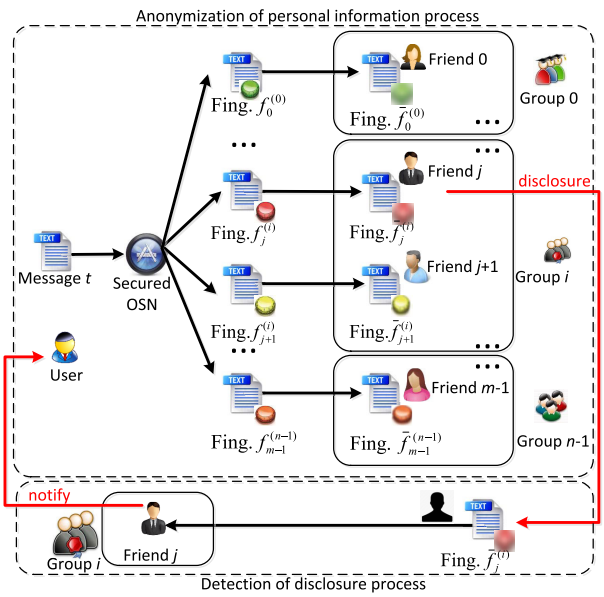

Fig. 2 Overview of proposed system.

Algorithm 1 Anonymization of personal information.

```
1: function AnonymizeInformation(input message t)
2:     P ← DetectSensitivePhrases(t, user's profile A);
3:     if P is not null then
4:         flag ← IsPersonalMessage(t);
5:         if flag is true then
6:             G ← CreateGeneralizationSchemas(P);
7:             InfoLoss ← QuantifyInfoLossOfGeneralizations (G);
8:             F ← CreateFingerprints(InfoLoss, t);
9:             DisplayFingerprintedMessages(F);
10:            SaveFingerprints(F);
11:        else
12:            DisplayMessage(t); // display the same t to all friends
13:        end if
14:    else
15:        DisplayMessage(t); // display the same t to all friends
16:    end if
17: end function
```

3.1.1 DetectSensitivePhrases Function

The data in a user's personal profile comprise seven attribute types (work, education, religion, etc.) and are all noun phrases. Therefore, we use noun phrase chunking library[†] 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 sensitive phrases in input message t are detected 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 [14].

The co-occurrence of two phrases A and B ($Co(A, B)$), Eq. 4 is the number of pages retrieved using a search engine from a huge dataset (such as Google or Wikipedia) containing both A and B ($Fr(A \cap B)$) divided by the number containing A or B ($Fr(A \cup B)$). We use Wikipedia for creating a search engine here. Two example co-occurrence metrics are $Co(\text{Shinjuku, Tokyo}) = \frac{1{,}578}{60{,}084} = 0.0263$ and $Co(\text{Shinjuku, Harvard University}) = \frac{28}{40{,}219} = 0.0007$. The result of co-occurrence analysis reveals that "Tokyo" is more similar to "Shinjuku" than "Harvard University."

$$Co(A, B) = \frac{Fr(A \cap B)}{Fr(A \cup B)} \tag{4}$$

Table 1 shows the results of detection. If the value of the co-occurrence metric is greater than the threshold α, the phrase is considered sensitive. On the basis of the 1,589 different sensitive phrases described in the evaluation section, we set α to 0.0169. By using α, we detect two sensitive phrases, p_0 = "Harvard University" and p_1 = "Tokyo," as shown in Eq. 5. In this example, even if a friend changes a sensitive phrase directly by using a synonym phrase like "Harvard University" or indirectly by using a similar phrase like "Tokyo," the system detects the modified phrase.

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

Table 1 Sensitive phrase detection.

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

$$\begin{aligned} P &= \text{DetectSensitivePhrases}(t, A) = \{p_0, p_1\} \\ &= \{\text{"Harvard University", "Tokyo"}\} \end{aligned} \tag{5}$$

3.1.2 IsPersonalMessage Function

Since messages posted in an OSN may include sensitive phrases that do not disclose personal information about the user, messages containing sensitive phrases are classified as either personal or general. *Personal messages* disclose personal information about the user while *general messages* do not. If the IsPersonalMessage function determines that the message is general, the system posts the same version of the message for all friends. Otherwise, the message is anonymized and fingerprinted.

In an evaluation (described in the next section), we found that sequential minimal optimization [15] is the best approach to creating classifier β used here. This classifier is used to determine whether input message t is a personal or general message by using Eq. 6. In example input message t, the result of classification is *true*. This means that t is a personal message. The system thus uses t to anonymize the personal information in subsequent steps.

$$flag = \text{IsPersonalMessage}(t) = true \tag{6}$$

3.1.3 CreateGeneralizationSchemas Function

Our system creates generalizations for each sensitive phrase in P by using generalization schemas (Eq. 7). The holonym relationships in the Wordnet lexical database [16] are used to create these schemas. For example, according to Wordnet, "Cambridge" is a direct generalization of "Harvard University." Figure 3 shows the results of generalization for two sensitive phrases: $G^{(0)}$ ="University" and $G^{(1)}$ ="Prefecture."

$$G = \text{CreateGeneralizationSchemas}(P) = \{g_j^{(i)}\} \tag{7}$$

From the about 16 million tweets in the TREC Tweets2011 dataset [8], we extracted 75,464 sensitive phrases exceeding co-occurrence threshold α. Of these sensitive phrases, 98.47% are covered by Wordnet. This shows that Wordnet covers most cases of sensitive phrases. For the few remaining cases, we use the sensitive phrase as the sole level of generalization.

3.1.4 QuantifyInfoLossOf Generalizations Function

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*), we extract the population of generalizations shown in Fig. 3 from the *HasPopulation* attributes of YAGO [10]. 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 sensitive phrases. The *Dis* for the generalizations of two sensitive phrases in P is described in Eq. 8. Table 2 shows the values of the precision metric, Samarati metric, and distribution metric for all possible generalizations of schemas G.

$$Dis(P) = \sum_{i=0}^{N-1} \frac{log(|p_i|)}{log(|P_i|)} = Dis(\text{Harvard U.}) + Dis(\text{Tokyo})$$
$$= \frac{log(21,000)}{log(317,672,000)} + \frac{log(13,230,000)}{log(4,299,000,000)} = 1.25 \quad (8)$$

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. 9, 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 sensitive phrases in P is described in Eq. 10. A special case of the $InfoLoss$ metric is presented in Sect. 6.2.

$$InfoLoss(g) = \sum_{i=0}^{N-1} InfoLoss(g_i) \quad (9)$$

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

$$InfoLoss(P) = Dis(\text{Harvard U.}) + Dis(\text{Tokyo}) = 1.25 \quad (10)$$

3.1.5 CreateFingerprints Function

This function 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. 11. 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[†] (we consider $ln0$ to be equal to 0). This score is an extension of the substitution metric proposed by Change et al. [17]. Evaluation using 1113 OSN message revealed that the threshold γ when using this score is 80.55. Use of this value ensures that our system 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 3.

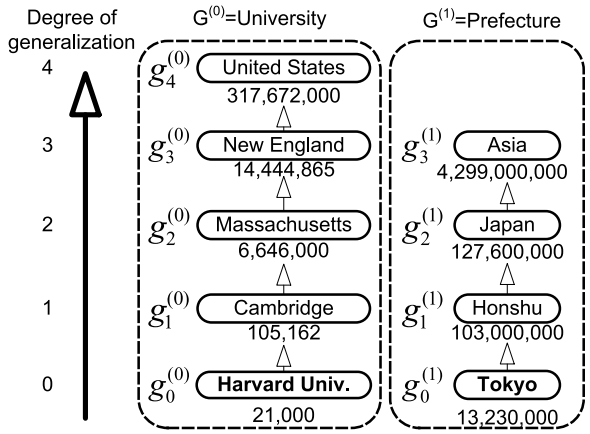

Fig. 3 Generalization schemas for two quasi-identifiers.

Table 2 Quantify possible generalizations.

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

Table 3 Check naturalness of message though replacement.

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	$fr(s) = ln(0) = 0$
$Fre(p,t) = Fre(\text{"}\mathbf{3}\text{"}, t) = 91.94 > \gamma(= 80.55)$		

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

Table 4 Assign generalizations for each group.

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

$$Fre(p,t) = \sum_{n=2}^{5} \sum_{pos=p_{index}-n+1}^{p_{index}} ln(fr(subphrase(t,pos,n))) \quad (11)$$

The appropriate degrees of generalization are shown in Table 4. 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 [18] to determine the proper levels of groups.

Each generalization is used to replace corresponding sensitive phrases in t of each group. A message is modified using synonyms to create a fingerprint for each friend in a group. The sysnonyms are obtained using Wordnet. The best fingerprints are checked by using frequency threshold γ. For example, our system suggested the following fingerprint for **"Friend 1"** in the "**Acquaintances**" group. "*My hometown is **Honshu**. My favorite food is sushi. After graduating from Tokyo University, I studied at **Harvard University** for **3** years as a computer science major*."

3.1.6 DisplayFingerprintedMessages Function

The system displays to the user the suggested fingerprints. The user can revise the fingerprints before posting the modified versions of the message. For example, the user can replace anonymized sensitive phrases with other suggested ones. The user can also instruct the system to generate other synonyms or edit the other phrases before posting the modified versions. Finally, the system stores the final fingerprints for subsequent disclosure detection and posts the fingerprinted messages.

3.2 Detection of Disclosure

The detection of disclosed personal information and identification of the person who disclosed it are done using Algorithm 2. For example, if a message t' ="*This is information about Adam. He is from Honshu and has studied at Harvard University for 3 years*" about a user (such as Adam Ebert) is posted in an OSN, the system automatically detects disclosed message t'. The system extracts the sensitive phrases using the DetectSensitivePhrases function described above. The result is set P' of detected sensitive phrases: p'_0 ="Honshu" and p'_1 = "Harvard University." The classifier β checks t' to determine whether it is the user's personal message or a general message.

Algorithm 2 Detection of disclosure.

```
1: function DetectDisclosure(input message t')
2:     P' ← DetectSensitivePhrases(t', user's profile A);
3:     if P' is not null then
4:         flag ← IsPersonalMessage(t');
5:         if flag is true then
6:             person ← DetectDisclosure(P', t', F);
7:             if person is not null then
8:                 Notify(person); // Send information about discloser to
   user
9:             end if
10:        end if
11:    end if
12: end function
```

In the DetectDisclosure function, the system uses co-occurrence threshold α to compare the sensitive phrases in P' with the stored generalizations created during the fingerprint creation process to determine to which group of friends the discloser belongs. The saved synonyms are then used to identify the person in the group who disclosed the personal information. In the example here, the system identifies "Friend 1" in the "Acquaintances" group as the discloser. It then sends a notification about the disclosure to the user. The user may then unfriend the person or move the friend to another group with a lower disclosure level.

4. Implementation

We used our system to build a web application for controlling the disclosure of information on Facebook. It anonymizes personal information and detects disclosures.

4.1 Anonymization of Personal Information

The user accesses the application with his/her existing account and composes a message. The application anonymizes the user's personal information and creates a unique fingerprinted version of the message for each of the user's friends. Figure 4 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 sensitive 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. Figure 5 shows the Facebook pages of two friends, "Bob Smith"

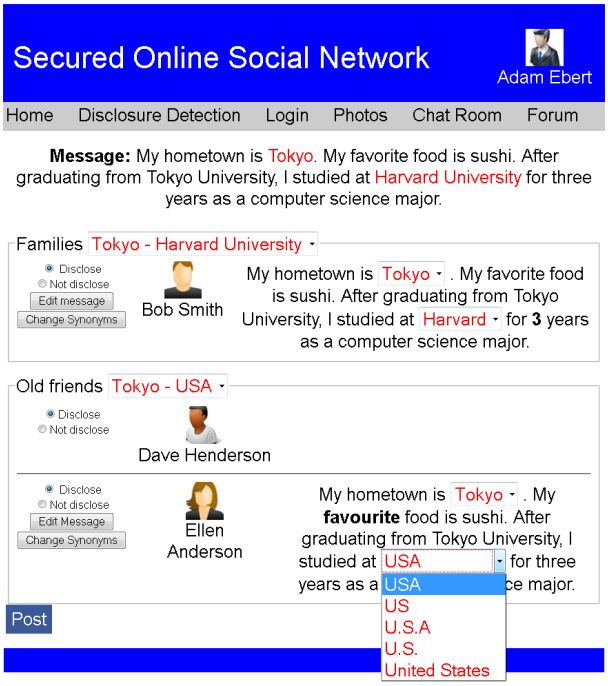

Fig. 4 Fingerprinted versions suggested for friends of user "Adam Ebert."

Fig. 5 Facebook pages of "Bob Smith" and "Ellen Anderson."

and "Ellen Anderson," who see different fingerprinted versions of the message.

4.2 Detection of Disclosure

If a friend discloses a user's personal information obtained from a message posted by the user on Facebook, the system automatically detects the disclosure and notifies the user, as shown in Fig. 6. 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

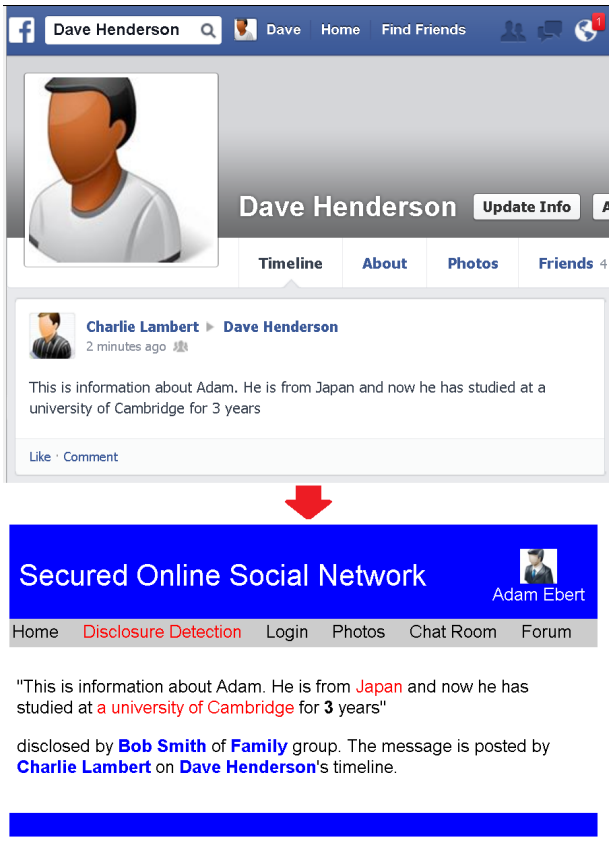

Fig. 6 Disclosure detection.

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 disclosures 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. Evaluation

From the about 16 million tweets in the TREC Tweets2011 Dataset [8], we extracted the ones in English using a language detection tool [19]. We then normalized the over 4 million extracted tweets (i.e., misspellings were corrected) using lexical normalization [20]. From the normalized tweets, we estimated the threshold for the co-occurrence metric of sensitive phrase detection, determined the best classifier for distinguishing between general and personal messages, and estimated the frequency metric used for naturalness checking. Finally, we used two thresholds and the classifier to calculate the number of possible generalizations for groups and possible synonyms for friends.

Table 5 Classifier creation results.

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**	$F1$
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%
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%
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%
NaiveBayesMulti	**82.3%**	**80.2%**	**79.9%**	74.5%	72.0%	71.5%	79.5%	79.2%	79.2%	79.1%	76.0%	76.5%
RandomCommittee	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%
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%
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%
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%

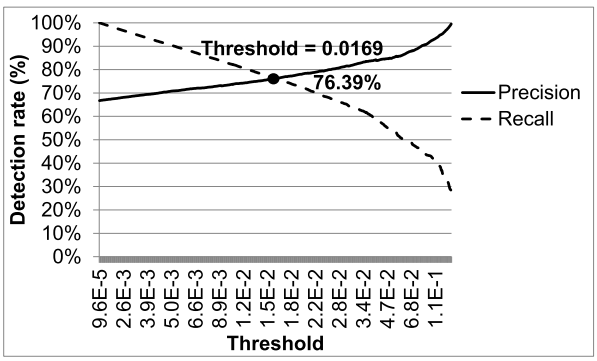

Fig. 7 Co-occurrence threshold.

5.1 Estimating Threshold for Co-occurrence Metric of Sensitive Phrases Detection

We used a name entity recognize algorithm [21] to extract 1589 different locations in the normalized tweets. The locations were compared with a list of countries[†] to find the best matches using the co-occurrence metric. The precision and recall for the 1589 locations are plotted in Fig. 7. We used 0.0169 as the co-occurrence threshold α in order to balance precision with recall.

5.2 Distinguishing between General and Personal Messages

Messages containing sensitive phrases were detected by comparing each phrase in the normalized tweets with certain phrases in a corpus of cities for hometown[††], universities, and colleges for education[†††], careers[††††], sports[†††††], religions[††††††], politics[*], and interests[**]. Co-occurrence threshold α was used to quantify each comparison. Finally, we extracted 137,628 tweets containing sensitive phrases as a dataset for evaluation.

We manually labeled 3,000 random sensitive tweets and ran 11 algorithms with 10-fold cross validation, as shown in Table 5. 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 (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 system 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. It had an $F1$ score of 92% and is used in this paper. β was used to extract 54,621 personal tweets from the 137,628 ones used for estimating the threshold of the frequency metric in the next subsection.

The best performances of the algorithms are shown in bold in Table 5. We did not experiment with the (1-gram+···+5-gram) model because the (1-gram+···+4-gram) model did not find any better solutions. Moreover, some algorithms with the (1-gram+···+5-gram) model took a long time to create classifiers. For example, the multinomial logistic regression (Logistic) algorithm with the (1-gram+···+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+···+5-gram) model.

[†]http://www.internetworldstats.com/list2.htm
[††]http://www.maxmind.com/en/worldcities
[†††]http://www.odditysoftware.com/page-datasales161.htm
[††††]http://www.careerdirections.ie/ListJobs.aspx
[†††††]http://listofsports.com/
[††††††]http://www.guavastudios.com/religion-list.htm
[*]http://www.gksoft.com/govt/en/parties.html
[**]http://www.hobby-hour.com/hobby_list.php

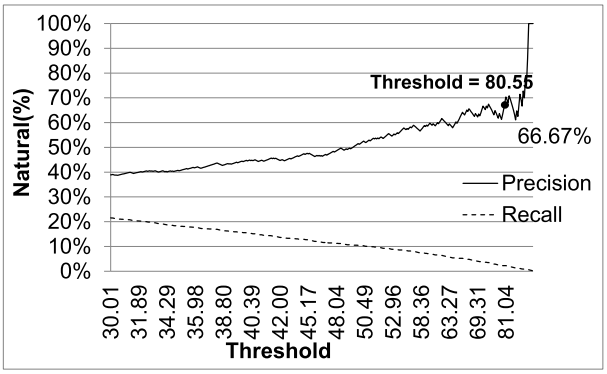

Fig. 8 Frequency threshold.

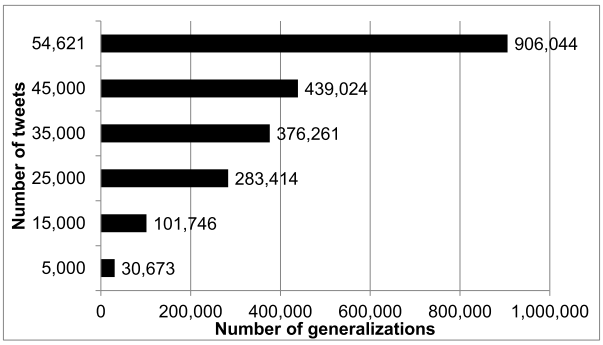

Fig. 9 Number of possible generalizations for groups.

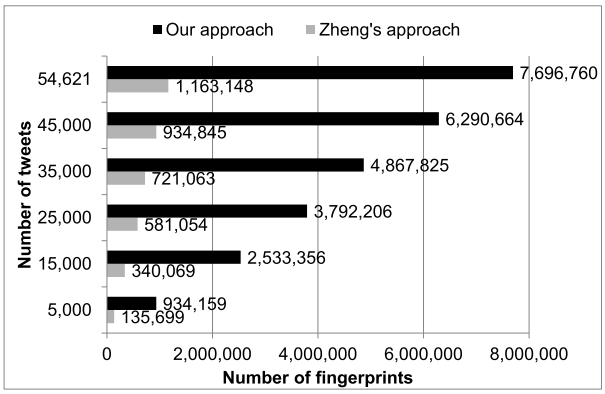

Fig. 10 Number of possible fingerprints for friends.

5.3 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 by using generalizations for the sensitive 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. 8. 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.

5.4 Number of Possible Generalizations for Groups

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

$$\bar{T} = \prod_{i=0}^{N-1} |T_i|, \tag{12}$$

where N is the total number of sensitive phrases, and $|T_i|$ is the number of generalizations of sensitive 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. 9. We created 906,004 generalizations from the 54,621 personal tweets, an average of 16.59 generalizations per tweet. These results show that our system 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. 9, many tweets in 9,612 tweets (from 45,000 to 54,621) are long diary blogs. Each blog contains more than seven sensitive phrases. Moreover, some sensitive phrases are repeated several times in the blog. Each blog creates more than 2,000 generalizations by using our system and is retweeted a few times. Therefore, these 9,612 tweets create more generalizations than other ones.

5.5 Number of Possible Fingerprints for Friends

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

$$\bar{F} = \sum_{i=0}^{T-1} \prod_{j=0}^{n_i-1} |S_j| \tag{13}$$

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 [7] to create fingerprints. The second was to use generalizations for sensitive phrases and synonyms for other phrases (our system).

As shown in Fig. 10, 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[†]. Therefore, our

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

approach creates a sufficient number of fingerprints for the average Facebook user.

6. Discussion

6.1 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 sensitive phrase to avoid disclosure detection, the use of the co-occurrence metric to detect sensitive phrases thwarts their efforts. The co-occurrence metric directly detects the exact sensitive phrase if the attacker uses a synonym or indirectly detects it if the attacker uses a similar phrase.

6.2 Limitation

In our system, the distribution metric supports quantifying information loss for sensitive phrases related to location (home, education, etc.) that have information about population in YAGO. However, YAGO does not have a population attribute for other sensitive 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 sensitive phrases about work ("student") and hometown ("Tokyo") are shown in Fig. 11. The $InfoLoss$ for them is calculated using

$$InfoLoss(P) = Dis(\text{Tokyo}) + Pre(\text{student}) = \frac{log(13,230,000)}{log(4,299,000,000)} + \frac{0}{5} = 0.74. \quad (14)$$

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

6.3 Usability and Privacy

Byun and Bertino [22] suggested that groups with higher usability and security should receive a version with a lower degree of anonymization. A distribution metric is used in our system to ensure that the degree of anonymization is appropriate for each group of friends.

For example, for the message "I'm from Tokyo," our system automatically creates fingerprints with different degrees of anonymization ("Tokyo" for *Best friends*, "Japan" for *Acquaintances*, and "Asia" for *Public* group), as shown in Fig. 12. In this example, friends in the *Best friends* group with the highest usability and security receive the lowest degree of anonymization (Tokyo) while friends in the *Public* group receive the highest one (Asia).

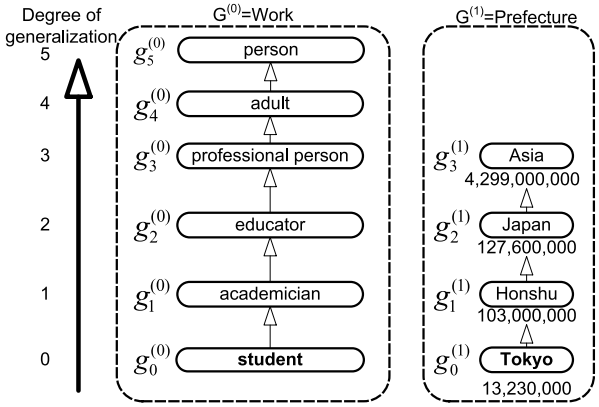

Fig. 11 Generalizations for two sensitive phrases.

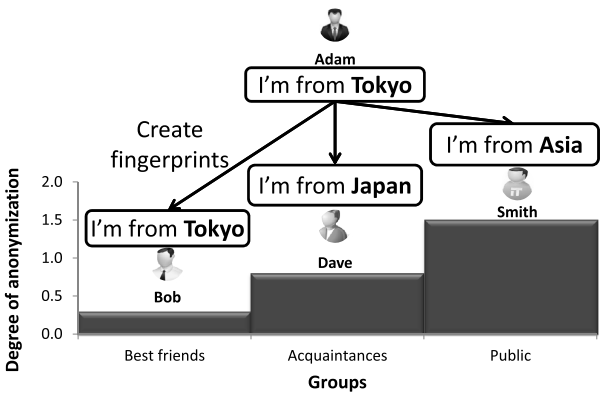

Fig. 12 An example of usability and privacy of our approach.

7. Conclusion

Our system anonymizes sensitive information 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 system can detect which friend has disclosed sensitive information about the user.

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 detect sensitive 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 system 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. It also includes anonymizing time-related

phrases in messages to prevent crimes (such as theft and kidnapping).

References

[1] F. Stutzman, R. Gross, and A. Acquisti, "Silent listeners: The evolution of privacy and disclosure on facebook," J. Priv. and Conf., vol.4, no.2, pp.7–41, 2012.
[2] D. Kokkinakis and A. Thurin, "Anonymisation of swedish clinical data," Proc. 11th AI in Medic., pp.237–241, 2007.
[3] B. Medlock, "An introduction to nlp-based textual anonymisation," Proc. 5th Conf. Lang. Res. and Eval., 2006.
[4] H.Q. Nguyen-Son, M.T. Tran, D. Tien, H. Yoshiura, N. Sonehara, and I. Echizen, "Automatic anonymous fingerprinting of text posted on social networking services," Proc. 11th Inter. Worksh. Digit. Forens. and Waterm., pp.410–424, 2013.
[5] Y. Zhang, G. Blackwood, and S. Clark, "Syntax-based word ordering incorporating a large-scale language model," Proc. 13th Conf. Euro. Assoc. for Comput. Ling., pp.736–746, 2012.
[6] C.Y. Chang and S. Clark, "Linguistic steganography using automatically generated paraphrases," Proc. 11th NAACL HLT, pp.591–599, 2010.
[7] X. Zheng, L. Huang, Z. Chen, Z. Yu, and W. Yang, "Hiding information by context-based synonym substitution," Proc. 8th Inter. Worksh. Digit. Forens. and Waterm., pp.162–169, 2009.
[8] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff, "Overview of the trec-2011 microblog track," Proc. 20th Tex. Retriev. Conf., 2011.
[9] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," Proc. 17th ACM SIGACT-SIGMOD-SIGART, p.188, 1998.
[10] F. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A core of semantic knowledge," Proc. 16th Inter. Conf. WWW, pp.697–706, 2007.
[11] T.H. Ngoc, I. Echizen, K. Komei, and H. Yoshiura, "New approach to quantification of privacy on social network sites," Proc. 24th Adv. Infor. Netw. and Appl., pp.556–564, 2010.
[12] M. Topkara, U. Topkara, and M.J. Atallah, "Words are not enough: Sentence level natural language watermarking," Proc. 4th ACM Worsh. Cont. Prot. and Sec., pp.37–46, 2006.
[13] M.J. Atallah, V. Raskin, C.F. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K.E. Triezenberg, "Natural language watermarking and tamperproofing," Proc. 5th Infor. Hid., pp.196–212, 2002.
[14] H. Kataoka, A. Utsumi, Y. Hirose, and H. Yoshiura, "Disclosure control of natural language information to enable secure and enjoyable communication over the internet," Proc. 15th Inter. Worksh. Sec. Prot., pp.178–188, 2010.
[15] J.C. Platt, "Fast training of support vector machines using sequential minimal optimization," Adv. in Ker. Meth., pp.185–208, 1999.
[16] G.A. Miller, "Wordnet: A lexical database for english," Commun. ACM, vol.38, no.11, pp.39–41, 1995.
[17] C.Y. Chang and S. Clark, "Practical linguistic steganography using contextual synonym substitution and vertex colour coding," Proc. Empirical Methods in NLP, pp.1194–1203, 2010.
[18] S. Machida, S. Shimada, and I. Echizen, "Settings of access control by detecting privacy leaks in sns," Proc. 9th Signal Image Technology and Internet Based Sytems, pp.660–666, 2013.
[19] N. Shuyo, "Language detection library for java," 2010.
[20] B. Han, P. Cook, and T. Baldwin, "Automatically constructing a normalisation dictionary for microblogs," Proc. Joint Conf. Emp. Meth. in Nat. Lang. Proc. and Comput. Nat. Lang. Learn., pp.421–432, 2012.
[21] X. Liu, S. Zhang, F. Wei, and M. Zhou, "Recognizing named entities in tweets," Proc. 49th ACL HLT, pp.359–367, 2011.
[22] J.W. Byun and E. Bertino, "Micro-views, or on how to protect privacy while enhancing data usability: Concepts and challenges," ACM SIGMOD Record, vol.35, no.1, pp.9–13, 2006.

Hoang-Quoc Nguyen-son received B.Sc. and M.Sc. degrees in computer science from the University of Science, VNU-HCM, in 2008 and 2011. He is currently pursuing a Ph.D. degree in computer science at The Graduate University for Advanced Studies (SOKENDAI), Japan. His current research interests include data hiding, anonymity, and privacy for natural language text.

Minh-Triet Tran obtained B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Science, VNU-HCM, in 2001, 2005, and 2009, respectively. He is currently Deputy Head of the Software Engineering Department, Faculty of Information Technology, University of Science, VNU-HCM. His research interests include cryptography and security, multimedia, human-computer interaction, and software engineering.

Hiroshi Yoshiura is a professor at The University of Electro-Communications, Tokyo, Japan. His current areas of research include security, privacy enhancement, security protocols, network security, copy protection, risk management, and watermarking.

Noboru Sonehara is a professor in the Information and Society Research Division, National Institute of Informatics, Tokyo, Japan. He received B.Sc. and M.Sc. degrees from Shinshu University, Japan, in 1976 and 1978, respectively. He received a PhD from Shinshu University in 1994. He has been Director of the Information and Society Research Division since 2006. His current areas of research include ICT security, privacy, trust, risk, resilience, and e-authentication platform design.

Isao Echizen is a professor in the Information and Society Research Division, National Institute of Informatics, Tokyo, Japan. He received M.Sc. and Ph.D. degrees in applied physics from the Tokyo Institute of Technology, Tokyo, Japan, in 1995 and 1997, respectively. His research interests include media security technology, privacy and business processes, anonymity systems, and anonymity metrics.