A k-anonymized Text Generation Method

Yu Suzuki, Koichiro Yoshino, Satoshi Nakamura

Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara, Japan
{ysuzuki,koichiro,s-nakamura}@is.naist.jp

Abstract. In this paper, we propose a method for automatically generating k-anonymized texts from texts which include sensitive information. Many texts are posted on social media, but these texts sometimes include sensitive information, such as living places, phone numbers, and SSNs. Even if sensitive information is removed from the texts, readers still be able to estimate the sensitive information from the anonymized texts, because the readers can guess sensitive information using remained information. To solve this problem, we propose a method for anonymizing texts using k-anonimization based techniques. This anonymization process is time consuming, we cannot identify appropriate anonymized strings in real time. Therefore, we proposed a method for generating an anonymization dictionary, and anonymize texts using the anonymization dictionary. In our experiments, we confirmed that our proposed method can anonymize texts in a practical time.

Keywords: k-anonymization, text processing

1 Introduction

Text data mining is one important technology for discovering patterns in a large size of textual dataset. However, if a dataset contains sensitive information, the dataset must not be processed by the other persons or organizations, because sensitive information must not be informed to the other persons. It is said that we can reuse the dataset for data mining if we sanitize a dataset by removing all sensitive information. However, even if we remove sensitive information, observers may estimate the sensitive information from the remaining strings, then the observers may expose the sensitive information. To solve this issue, we should remain characters as many as possible for mining texts, but we must remove not only all sensitive information but also their related information.

There are two kinds of sensitive information: *identifier* and *quasi-identifier*. *Identifier* is information which can identify a specific person by only the information. For example, persons' names, phone numbers, and social security numbers (SSNs) should be treated as identifiers. We must remove identifiers completely if they appear in the target texts. For example, if a sentence "Suzuki is a faculty member of NAIST. His research interests include data engineering." we should remove the person's name "Suzuki," and rewrite the sentence to "* is a faculty member of NAIST." where "*" is a placeholder of the identifier.

Quasi-identifier is information which cannot identify a specific person, but which can identify with the other information. In the above sentence, “faculty member,” “NAIST,” and “data engineering” should be considered as quasi-identifiers. We cannot identify the specific person if one of the three strings is appeared in one sentence, but we can identify if the three strings appeared in the sentence.

The goal of this research is to modify target texts to sanitized texts which cannot identify persons but can identify attributes. In our study, we set the name of persons as identifiers, and the attributes related to the persons as quasi-identifiers. The main idea is based on k-anonymity[7], which is proposed by Sweeney for anonymizing structured data. In this technique, if more than k records are considered as the same, the records are called k-anonymized. In our method, we use this technique for anonymizing quasi-identifiers in the text.

2 Related Work

One simple solution for text anonymization is only remove sensitive information from texts[2]. We call this sensitive information as *identifier*. However, if these identifiers are removed from texts, the observers will be able to estimate sensitive information. For example, when a purchasing history is given, we can estimate customers as female if the customers bought lady’s clothes and cosmetics. To prevent this estimation, we should detect *quasi-identifiers*, which are not identifiers but observers can estimate identifiers by the information, and partly delete.

Many research about anonymization are done for structured data. First, k-anonymity is proposed by Sweeney[7] for anonymizing structured data. l-diversity[4] and t-closeness[3] are proposed for protecting privacy, there are many kinds of algorithm for protecting privacy information. However, this method is for structured data, then it is not discovered whether these techniques can apply to text data.

On the other hand, cryptography-based methods are also proposed[1]. In this method, the system encrypts target data, and apply data mining techniques for the encrypted data. The advantage of this method is the persons who performs data mining cannot understand the contents of the target data. This method is also used for structured data, we cannot simply apply these techniques to text data.

Nguyen-Son et al.[6] proposed a method for anonymizing texts by generalyzing sensitive phrases. For example, if there is a sensitive phrase “He lives in *Paris*,” the system generalize a term *Paris* to *France* or *Europe*, and make a sanitized phrase “He lives in *France*.” In this study, they assume that a list of sensitive information is given, but actually these data is not always given. In our study, we assume that the candidates of privacy or sensitive data is given, which is more reasonable settings.

We proposed a method for generating anonymization dictionary rapidly[5] in the past. In this method, we used several heuristics and the set theory in order

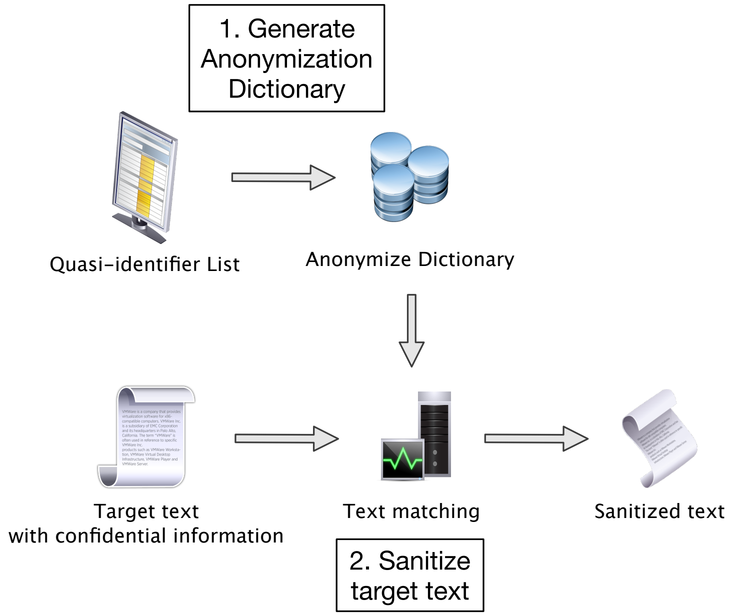


Fig. 1. Overview of our proposed system

to build a dictionary at a high speed. However, in this paper, we do not use these techniques because we earn enough speed by keeping all the data in the memory.

3 *k*-anonymization of texts

In this section, we describe how to generate k-anonymized texts from texts which include sensitive information. Figure 1 shows the overview of our proposed method.

Input of our system is a set of identifiers I, a set of quasi-identifiers Q, a parameter k which represents a strength of anonymity, and a target text which has sensitive information T. Output of our system is a sanitized text which has no sensitive information.

In our proposed method, we did the following steps to sanitize texts as follows:

1. Generate anonymization dictionary D from a list of quasi-identifiers
2. Sanitize target texts
   (a) Remove all terms which consist of I
   (b) Replace terms which consist of Q with the sanitized text
3. Output sanitized texts

3.1 Generation of Anonymization Dictionary

In this section, we describe how to generate an anonymization dictionary from a quasi-identifier list. Table 1 shows an example of an anonymization dictionary. In this table, there are five attributes: term, anonymized term (anonymized), k, number of remaining characters (remain), and number of characters.

A quasi-identifier list $Q = \{q_1, q_2, \cdots, q_N\}$ is given by users where q_i is a quasi-identifier. First, we generate all possible candidates of anonymized strings from q_i. We define the anonymized strings as the strings which are partly concealed by '*'. For example, if q_i is "cat", we generate a candidate list which consist of ca*, c*t, *at, *t, *a*, and c* where * means one or more deleted characters. We do not consider one asterisk (* only) and original string (cat) as a candidate list. We define $\boldsymbol{q}'_i = \{q'_{i,1}, q'_{i,2}, \cdots, q'_{i,n(q_i)}\}$ as a set of anonymized strings which corresponds to q_i. Here, we find that $n(q_i)$, a number of anonymized strings in $\boldsymbol{q}'_i$, is less than or equals to $2^{l(q_i)} - 2$ where $l(q_i)$ is a number of characters in q_i.

Next, we calculate k which is a number of terms that match the anonymized strings. We generate regular expression which corresponds to $q'_{i,j}$, and count the number of terms that the regular expression matches. For example, if the regular expression is 'ca*', the strings 'car', 'cam', and 'can' matches. If 'ca*' matches 10 strings in Q, the value of k which corresponds to 'ca*' is 10. Of cause, anonymized strings must match the original strings (ex. 'ca*' must match 'cat'). Therefore, k is always grater than 1 and less than the number of strings in Q.

Finally, we store $\boldsymbol{q}'_i$ to the anonymization dictionary D if k is more than two. We do not store columns if $k = 1$, because $k = 1$ means that we can identify original string from the anonymized strings, then the anonymized string is useless for anonymization.

Table 1. Anonymization dictionary of a quasi-identifier "JAPAN"

term	anonymized	k	remain	length
JAPAN	JAP*N	2	4	5
JAPAN	J*P*N	3	3	5
JAPAN	JA*AN	3	4	5
JAPAN	J*AN	7	3	5
JAPAN	JA*N	8	3	5
	⋮			
JAPAN	J*	5,281	1	5
JAPAN	*P*	8,484	1	5
JAPAN	*A*	17,395	1	5

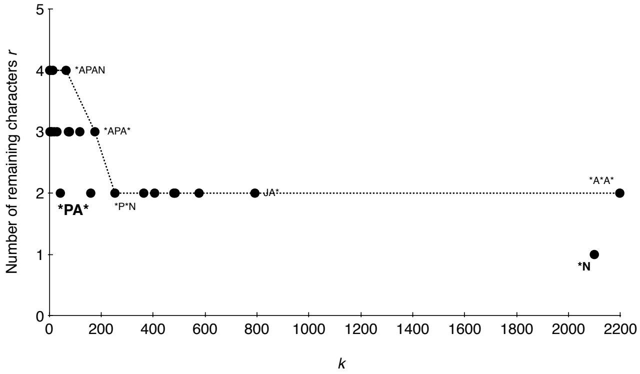


Fig. 2. The relationship between k and number of remaining characters

3.2 Applying Anonymization Dictionary to Text

Next, we apply the anonymization dictionary to the target texts with sensitive information. We used k which identifies a strength of anonymization, and the anonymization dictionary D which is described at section 3.1.

First, we pick up a substring s_i from i-th to $L + i$-th characters of the target text T, where L is the substring which has the maximum number of characters in Q, and $|L|$ is the number of characters in L. Then, we generate a set of substrings $S = \{s_{0,|L|}, s_{1,|L|+1}, \cdots, s_{|T|-1,|T|}\}$. For example, if the target text is "`I␣have␣a␣cat`" and $|L|$ is set to five, S is {"`I␣hav`", "`␣have`", "`have␣`", "`ave␣a`",$\cdots$, "`a␣cat`", "`␣cat`", "`cat`", "`at`", "`t`"}.

Then, we retrieve strings q_i that include $s_{i,|L|+i}$ at the beginning of q_i, and q_i and $s_{i,|L|+i}$ has the longest common sequence, from Q. For example, if $s =$ `cat`, we retrieve rows with term "cat." If $s =$ `have␣a␣cat`, we retrieve "have a cat," but we do not retrieve "have" because the length of strings are shorter than the retrieved strings. We also do not retrieve "cat" because the beginning of these strings are different.

Next, we select an anonymized string from D. The selected candidates of anonymized strings should have the value k more than the value the user specified. From the candidates, we select the anonymized string with k has the most lowest value. If there are multiple candidates, we select the anonymized string which has the large number of remained characters. After these process, we randomly select one anonymized string if there are multiple candidates.

For example, we assume that there is a string "Japan" in the target text, and we find that there is the term "Japan" in D. Table 1 shows the original string, the anonymized string, k, number of remained characters, and the length of the original string. If the user set the value of k to 2, the first row "JAP*N" is selected. If the user set the value of k to 3, there are two candidates "J*P*N" (second row) and "JA*AN" (third row). Then, we select "JA*AN" from the candidates because the length of remained character of “JA*AN” is higher than that of “J*P*N.”

Finally, we replace the retrieved string with the anonymized string. For example, if there is a string “I live in Japan.” and a user set k to 3, the system replaces the selected string “Japan” to “JA*AN.” Therefore, the output string is “I live in JA*AN.” If a user set k to $5,000$, the output string is “I live in J*.” Therefore, readers cannot distinguish whether s/he lives in Japan, Jamaica, Jakarta, or the other places.

Figure 2 shows the relationship between k and number of remaining characters r. Points in this figure correspond to the rows in Table 1. As we mentioned in the paragraph above, an anonymized string will never selected if there are more than one anonymized strings which have grater k values and grater number of remaining characters. For example, there is a point where $k = 10$, $r = 180$, and anonymized string is “*PA*.” This anonymized string is never selected because there are many strings which have larger value of k and r. As a result, the candidates which are on dotted line on Fig. 2 are selected, but the other candidates are never selected. Therefore, we remove candidates which are not on line.

4 Case study

To confirm the accuracy of our proposed method, we did our experimental evaluation using Wikipedia data. First, we prepared all titles in Japanese Wikipedia[1] for Japanese version and English Wikipedia[2] for English version, and use this list as a quasi-identifiers. The number of titles in Japanese Wikipedia is $1,697,542$ and that in English Wikipedia is $13,124,734$.

We constructed a Web-based anonymization system[3]. This system which corresponds to text anonymization method, which is described in section 3.2, is constructed using Ruby on Rails 5 and Oracle Database 12c. For constructing anonymization dictionary, we also constructed a system using Java 1.8. It takes about one week for generating this anonymization dictionary using 10 servers about a total of $1,000$ cores, 2TB memories total. We could not calculate all strings in the data because of time, because if a number of characters are large, it takes a long time for generating data for anonymization dictionary. How to construct this anonymization dictionary in a practical time should be a future work.

Fig. 3 shows the result of the section 1., first paragraph of this paper using our proposed system. We used English Wikipedia dataset for this anonymization. We set k to 2. From this text, 297 of 689 characters (43.1%) remain. Therefore, we found that it is too difficult to read, because there is almost no information from this text. This because, alphabet has only 26 characters, several sentences unexpectedly match the strings in the anonymization dictionary.

[1] https://dumps.wikimedia.org/jawiki/20170501/jawiki-20170501-all-titles-in-ns0.gz

[2] https://dumps.wikimedia.org/enwiki/20170520/enwiki-20170520-all-titles-in-ns0.gz

[3] http://bigdata.naist.jp/anonymizer/

Text data mining is one of important technologies for discovering patterns in large size of textual datasets. If the datasets consist of sensitive information, the datasets must not be processed by the other persons or organizations. However, if we completely sanitize these datasets by removing all sensitive information, we can send the datasets to the other persons. On the other hand, if we remove all information related to the sensitive information from the texts, we cannot extract any information from the texts. Therefore, we should remain characters as many as possible, but we must remove all sensitive information. We call this text modifying process as sanitizing.

Original

*ext d*a m**g * * of *port*t t**o**s *r d***r*g ***ns * *r* *ze of text*l d***s. *f * d***s *n** of ***enti* *****n, * d***s *u* *t * *oc**d * * o*r **ons or **i*t*ns. *owe*r, if we **l*e* **tize ** d***s * **** *l ***enti* *****n, we *n *nd * d***s * * o*r **ons. *n * o*r h*d, if we *** *l *****n **ted * * ***enti* *****n ** * tex*, we *n*t ex**t *y *****n ** * tex*. *here**, we *ould *** ***t*s * *ny * po*i*e, *t we *u* *** *l ***enti* *****n. *e *ll ** text *dify*g *oc*s * **tiz*g.

Anonymized

Fig. 3. k-anonymized text of the section 1., first paragraph of this paper

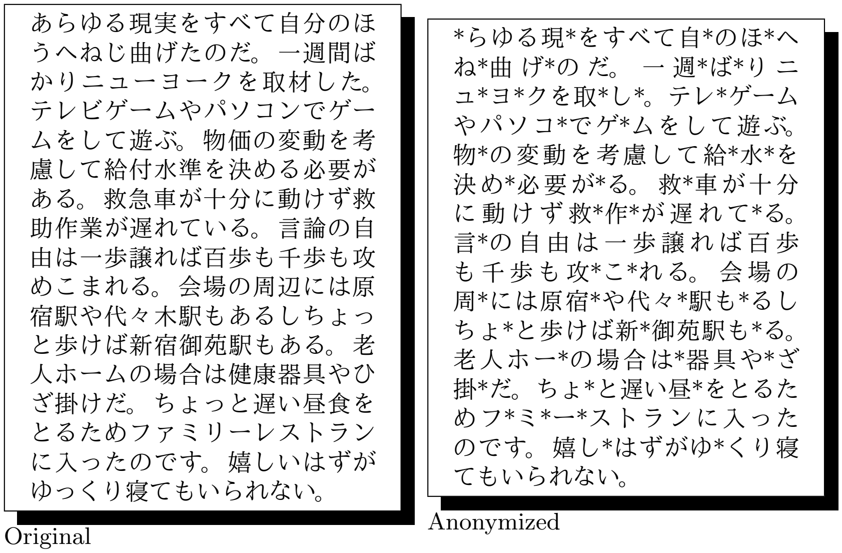
We set $k = 2$, and use Japanese anonymization dictionary. We did our experiments for Japanese texts[4]. Fig. 4 191 of 237 characters (80.6%) remain. From this anonymized text, we partly get information which is also get from the original text. This is because, Japanese sentences consist of many kinds of characters, unexpected sentences do not match the strings in the anonymization dictionary.

The evaluation measure of text anonymization method is important. In our experimental evaluation, we count how many characters remain for measuring accuracy of text anonymization method. However, we found that what kind of rules can be extracted from the anonymized data, and did the anonymization method can remove sensitive strings practically? should also be evaluation measures. We consider these evaluation measures as future works.

5 Conclusion

In this paper, we proposed a method for anonymizing texts which include sensitive information. We use k-anonymization based techniques for anonymization. First, we generate an anonymization dictionary using quasi-identifier list. Then, we generate an anonymized text using this anonymization dictionary.

[4] ATR 503 sentences http://research.nii.ac.jp/src/ATR503.html



あらゆる現実をすべて自分のほうへねじ曲げたのだ。一週間ばかりニューヨークを取材した。テレビゲームやパソコンでゲームをして遊ぶ。物価の変動を考慮して給付水準を決める必要がある。救急車が十分に動けず救助作業が遅れている。言論の自由は一歩譲れば百歩も千歩も攻めこまれる。会場の周辺には原宿駅や代々木駅もあるしちょっと歩けば新宿御苑駅もある。老人ホームの場合は健康器具やひざ掛けだ。ちょっと遅い昼食をとるためファミリーレストランに入ったのです。嬉しいはずがゆっくり寝てもいられない。

Original

*らゆる現*をすべて自*のほ*へね*曲げ*のだ。一週*ば*りニュ*ヨ*クを取*し*。テレ*ゲームやパソコ*でゲ*ムをして遊ぶ。物*の変動を考慮して給*水*を決め*必要が*る。救*車が十分に動けず救*作*が遅れて*る。言*の自由は一歩譲れば百歩も千歩も攻*こ*れる。会場の周*には原宿*や代々*駅も*るしちょ*と歩けば新*御苑駅も*る。老人ホー*の場合は*器具や*ざ掛*だ。ちょ*と遅い昼*をとるためフ*ミ*ー*ストランに入ったのです。嬉し*はずがゆ*くり寝てもいられない。

Anonymized

Fig. 4. *k*-anonymized text written in Japanese

In our method, we convert strings which are in quasi-identifier list to k-anonymized strings. If strings are k-anonymized, some characters are replaced to "*", then there are more than k kinds of quasi-identifiers which correspond to the replaced strings.

In the experimental evaluation, we constructed the anonymization dictionary, and also constructed a Web application. We input English and Japanese texts to the system, and observed the outputs of our proposed system. From the output texts, we found that our proposed system is effective if the number of variations of characters are large, like Japanese, but the texts are almost broken if the number of variations of characters are small, like English.

In future work, we should construct a method for generating quasi-identifiers from given texts. In this paper, we use Wikipedia titles as quasi-identifiers, because these strings have many proper nouns. However, in our experiment, we discovered that our proposed system conceal many general nouns, then users are hard to read the anonymized texts. To solve this problem, we should develop a method which can remain more strings to be able to read the strings and remove sensitive information.

Moreover, we should construct a method to set an appropriate value of k automatically. If we set extremely small value to k, sensitive information may be appeared to the anonymized texts, but if we set large value to k, the readability of the anonymized texts should be low, and the information which can be extracted from the anonymized texts will reduce.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. pp. 439–450. SIGMOD '00, ACM, New York, NY, USA (2000), http://doi.acm.org/10.1145/342009.335438
2. Kokkinakis, D., Thurin, A.: Anonymisation of Swedish Clinical Data, pp. 237–241. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
3. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering. pp. 106–115 (April 2007)
4. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data 1(1) (Mar 2007), http://doi.acm.org/10.1145/1217299.1217302
5. Maeda, W., Suzuki, Y., Nakamura, S.: Fast text anonymization using k-anonyminity. In: Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services. pp. 340–344. iiWAS '16, ACM, New York, NY, USA (2016), http://doi.acm.org/10.1145/3011141.3011217
6. NGUYEN-SON, H.Q., TRAN, M.T., YOSHIURA, H., SONEHARA, N., ECHIZEN, I.: Anonymizing personal text messages posted in online social networks and detecting disclosures of personal information. IEICE Transactions on Information and Systems E98.D(1), 78–88 (2015)
7. Sweeney, L.: K-anonymity: A model for protecting privacy. International Journal of Uncertain. Fuzziness Knowledge-Based Systems 10(5), 557–570 (2002)