Fast Text Anonymization using k-anonyminity

Wakana Maeda, Yu Suzuki, Satoshi Nakamura
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 6300192, Japan
maeda.wakana.mo7@is.naist.jp

ABSTRACT

In this paper, we propose a method for anonymizing unstructured texts using a quasi-identifier list. In our method, the system redacts from some parts of quasi-identifiers in the texts to the alternate characters such as "∗", in order to prevent re-identification of information which should be kept in secrecy. However, this method has a room for an improvement for keeping the information on the original text as is. If the system anonymizes the texts and keeps the original texts as much as possible, the accuracy of the outputs by data mining techniques for the anonymized texts should be useful. Our method anonymizes quasi-identifiers to remain substrings which do not contribute to re-identification, in order to keep the information on the original texts as is.

Concretely, the system identifies the substrings which should be redacted to satisfy the following two conditions: 1) Any terms in the quasi-identifier list satisfies k-anonymity by redacting characters. 2) The number of redacted characters is minimized. From the quasi-identifier list, we construct the anonymization dictionary which records the two number in advance; the number of quasi-identifiers which are anonymized in the same way, and a number of redacted characters of the anonymized quasi-identifier. However, this construction step is time consuming, because the system needs to retrieve a huge number of patterns. To solve this problem, we propose an acceleration method for constructing the anonymization dictionary using several heuristics and the set theory.

1. INTRODUCTION

Recently, bigdata, data collections with high velocity, high volume, and high variety, are used for decision making, insight discovery and process optimization. When these data collections include the information which should be kept in secrecy, we should anonymize the information. The main targets of existing anonymization techniques are the structured data. However, there are a lot of textual data which are not structured. Therefore, we should develop a method for anonymizing textual data.

There are two types of information which should be kept in secrecy: identifier and quasi-identifier. Identifier is a piece of information which can identify a specific entity; such as name, phone number and social security number. Quasi-identifier is a piece of information which cannot identify a specific entity by only using the information, but can identify by combining the other information; such as sex, nationality and birth place. When we anonymize texts, we should remove strings if the strings are identifiers. However, we do not have to remove the quasi-identifiers if they do not correspond to the other information resources. If we accurately convert quasi-identifier to k-anonymized strings, we can appropriately anonymize the texts, and also can suppress a reduction of information in the texts.

Redaction is one of major text anonymization methods for electronic health record in the United States[4]. This method replaces detected terms which should be anonymized in text to alternate characters such as "∗". This is effective against identifiers. However, the problem is that the number of remaining characters decreases excessively if we anonymize quasi-identifiers in the texts using redaction.

The aims of this study are as follows: 1) We prevent from identifying the quasi-identifier in the texts. 2) We preserve original texts as much as possible.

We proposed a method that redacts substrings of quasi-identifiers based on k-anonymity[8]; that is, a quasi-identifier is called anonymized when a quasi-identifiers which is redacted in the same way exists least than k times in the quasi-identifier list. In addition, we redact substrings of the quasi-identifier to minimize the number of redacted characters.

Therefore, our method needs to grasp two numbers; the number of quasi-identifier which is redacted in the same way, and a number of redacted characters about an anonymized quasi-identifier. The system previously needs to grasp the two numbers when the system selects redacted quasi-identifiers to fulfill the threshold. So our method keeps the two numbers into the anonymization dictionary. The advantage of using the anonymization dictionary is that we can observe the trade-off relation between the two numbers. Based on the results, we can select effective value of k. When we construct the anonymization dictionary, we faced the problem that it takes much time to calculate the number of quasi-identifiers which are redacted in the same way. The reason is because the system needs to search the quasi-identifier list using all anonymyzation patterns whose substrings redacted. Therefore, the problem is that retrieval targets includes unrelated terms. Moreover, another problem is that we should

iiWAS '16, November 28 - 30, 2016, Singapore, Singapore

ISBN 978-1-4503-4807-2/16/11…$15.00

DOI: http://dx.doi.org/10.1145/3011141.3011217

calculate result for too many anonymyzation patterns if the length of a quasi-identifier becomes longer.

In this study, we propose a method for reduce the processing time by reducing retrieval targets based on set theory.

Our contributions in this paper are as follows:

- The system extracts effective retrieval targets using a paired information; a character and a position where the character appears. See section 3.1.1.
- The system reduces retrieval targets using the intersection of two sets of anonymyzation patterns; If the system retrieves anonymyzation patterns when $n-1$ characters of a quasi-identifier are redacted, the system uses two anonymyzation patterns when n characters of the quasi-identifier are redacted. See section 3.1.3.

2. RELATED WORK

Methods for anonymizing unstructured data are categorized three types: pattern matching based method[5], machine learning based method[9], and their combined method[2]. Meystre et al. [4] reported that the pattern matching method is the major method for anonymization of texts.

As an anonymous indicator, k-anonymity[8] is generally used when we construct an anonymization system for structured data. An entity is called k-anonymized when the same entity appears in a dataset at least k times. One of methods to achieve k-anonymity for structured data is the combining generalization and suppression[7]. Generalization involves replacing a value with a less specific but semantically consistent value; the original ZIP codes 02138, 02139 can be generalized into 0213*. Suppression involves not releasing a value at all; the original ZIP codes 02138, 02139 can be suppressed to *****. Other k-anonymization algorithm is thought of the k-anonymity problem as the k-member clustering problem[1]. Generalization indicates one anonymization method for texts that replaces a term for generic concept.[6]. When we use generalization for text data, we previously needs knowledge of relationships between term and term concept.

On the other hand, removing term is also proposed[10]. However, we want not to remove terms as much as possible for minimizing the loss of information. So, one possible idea is to remove only substrings of a term in the list not to remove the whole term. The method for anonymizing unstructured texts using the remove of substrings, positional sampling method is proposed[3]. This method removes all characters except those corresponding to 1-bits of the seed. For example, where the seed equals $\{1, 0, 1, 1\}$ and input the text is "the third time", 2-th character where the seed value is 0 of the term and spaces are removed; "h" of "the", "h" of "third" and "i" of "time" are removed. Output text is "tetirtme." This method can retain string matching statistics and prevent from reconstructing the original string from the anonymized string. However, anonymized text data using this method is not readable for human, so analysis except using string matching is difficult.

3. OUR PROPOSED METHOD

The goal of this method is to anonymize unstructured texts. Concretely, the aims are as follows: 1) We prevent from identifying the quasi-identifier in the texts. 2) We preserve original texts as much as possible. To accomplish these aims, we prepared the text d in the document collection C, a quasi-identifier list W which has many quasi-identifiers shown in table 1, and a parameter k which indicates the risk threshold of re-identification. If k has higher value, the risk of re-identification reduces. For simplicity, we assume that there is no identifier in C, because we focus on anonymizing quasi-identifiers.

At section 3.1, we explain a naive method for anonymizing unstructured texts d using a quasi-identifier list W and a parameter k. Our method redacts substrings of quasi-identifiers based on k-anonymity[8]; that is, a quasi-identifier is called k-anonymized when a quasi-identifier wich is redacted in the same way exists least k times in W. In addition, we select substrings of the quasi-identifiers to minimize the number of redacted characters. Therefore, our method needs to grasp two numbers; the number of quasi-identifier which is anonymized in the same way, and a number of redacted characters about an anonymized terms. The system previously needs to grasp the two numbers when the system selects anonymized terms to fulfill the threshold. So our method keeps the two numbers into the anonymization dictionary D. The advantage of using D is that we can observe the trade-off relation between the two numbers. If users change the value of k, the number of redacted characters changes. Then, the users find the appropriate value of k by browsing the anonymized texts.

However, the process of constructing D is time consuming The reason is because the system needs to search W using all anonymyzation patterns whose substrings redacted when the system grasps the number of anonymized terms in the same way. Therefore, we need search the number of terms which satisfies the regular expressions of their substrings. For example, if we have a term "crew," we should calculate how many terms which match the following 14 patterns: "∗rew," "c∗ew," "cr∗w," "cre∗," "∗∗ew," "c∗∗w," "cr∗∗," "∗r∗w," "∗re∗," "c∗e∗," "∗ ∗ ∗w," "c∗ ∗ ∗," "∗r∗∗," and "∗∗e∗," where "∗" is an alternate character. If the length of a term becomes longer, we should calculate result count for too many patterns. To reduce this computational cost, we use both a heuristic rule and a set theory based method for reducing target terms. First, we look up the terms which have the same length and, at least one character at the same position. For example, if we have a term "crew," we retrieve terms which length is 4, then the terms "draw," "docs," and "cram" are retrieved. Then, we pick up the terms which the first character is "c," second one is "r," third one is "e," or fourth one is "w." After this process, we pick up "draw" and "cram" from the retrieved terms, and remove "docs."

Next, we used set theory for reducing retrieval target terms in W. The system reduces retrieval targets using the intersection of two sets of anonymyzation patterns ; If the system retrieves anonymyzation patterns when $n-1$ characters of a quasi-identifier are redacted, the system uses two sets of anonymyzation patterns when n characters of the quasi-identifier are redacted. Moreover, the system reduces the frequency of search times when the system calculates the number of ;by breaking process when the number of elements included in an intersection of sets of $n-1$ anonymyzation patterns is less than the threshold.

3.1 k-anonymization of substrings using pattern matching

Figure.1 shows the overview of our proposed method. Our

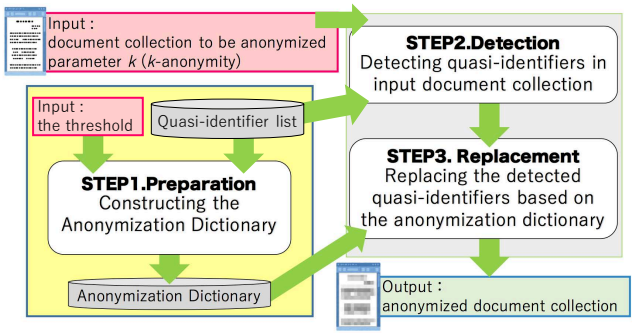

Figure 1: Overview of our proposed method

proposed method consists of 3 steps.

1. We construct the anonymization dictionary based on the quasi-identifier list.
2. We detect quasi-identifiers in input the collection.
3. The system replaces the quasi-identifiers which detected by Step 2 for anonymized strings using the anonymization dictionary based on parameter k and to minimize the number of redacted characters.

Step 1. Construction of the anonymization dictionary

We construct an anonymization dictionary D, which is a table from original strings to anonymized strings shown in table 2, from the quasi-identifier list W whose quasi-identifiers. `term_id` is an ID which corresponds to an original string in W shown in table 1. We define w_{id} as the quasi-identifier with `term_id` in W. $\hat{k}$ is the number of anonymized terms in the same way; that is the number of quasi-identifiers which satisfy the regular expressions of their substrings in W. r is the number of characters which are replaced with the alternate characters. `a_string` is an anonymized term which correspond to the term in W. S_{id} is a set of `a_string`s which corresponds to w_{id}.

At first, the system finds S_{id} that the system redacts substrings of each w_{id} excluding redacted all characters pattern. For example, the system generates S_{id} whose the anonymizied string; w_{id} is replaced substrings for alternate character such as "*". Where the string length of w_{id} is l, the number of listed pattern is 2^{l-1} excluding redacted all characters. The system calculates $\hat{k}$ to all these patterns. After that, the system records $\hat{k}$ and r of each `a_string` in S_{id} on D.

Step 2. Detection of quasi-identigiers

In this step, the system detects the quasi-identifiers in C based on W. The system extracts n-gram phrase to 2-gram phrase from each sentence in d. Then, the system checks whether an extracted phrase is included in W or not. If the phrase is included, the system detects the phrase as the quasi-identifier.

Table 1: Word table W

`term_id`	`term`
1	crew
2	draw
3	crawl
⋮	⋮

Step 3. Replacement of the detected quasi-identifiers

In this step, the quasi-identifiers which is detected in Step2 are replaced based on D which is constructed in Step1. Which `a_string` in S_{id} to use for anonymizing w_{id} is determined based on k and r. First, the system extracts `a_string` whose $\hat{k}$ is more than k. If multiple `a_string`s exist, the system selects one whose r is minimized. The system replaces the redacted character of selected pattern for alternate character such as "$*$". The system performs these processings for all sentences and all documents, and outputs anonymized document collection.

3.2 Acceleration of the anonymization dictionary

Where the string length of w_{id} is l, constructing D needs pattern matching 2^{l-1} times on W in order to calculate $\hat{k}$ on W. When we use pattern matching, the system retrieves using regular expressions to W. Where the number of the term in W is M, the number of caluculation is $2^{l-1}M$. Too large l and M causes unrealistic processing time in constructing D.

As mentioned above, constructing D has a frequency of calculation problem and a processing time problem. We propose a method for reducing retrieval target terms and the processing time using set theory.

We set W, and input parameter $K(\geq 2)$ as minimum k-anonymity.

1. For Step1-1, we generate a paired information; a character and a position where the character appears.
2. For Step1-2, based on the paired information generated Step1-1, we extract effective pattern matching targets.
3. For Step1-3, the system reduces retrieval targets using the intersection of two sets of anonymyzation pattern; If the system retrieves anonymyzation patterns when $n-1$ characters of w_{id} are redacted, the system uses two sets of anonymyzation patterns when n characters of w_{id} are redacted.

Table 2: Anonymization Dictionary D

`term_id`	$\hat{k}$	r	`a_string`
1	2	1	cr∗w
1	2	1	∗rew
1	3	1	cre∗
1	4	2	c∗∗w
1	7	2	∗r∗w
1	8	2	∗∗ew
1	13	2	c∗e∗
1	28	2	∗re∗
1	35	2	cr∗∗
1	92	3	∗ ∗ ∗w
1	400	3	∗∗e∗
1	419	3	∗r∗∗
1	628	3	c∗ ∗ ∗
⋮	⋮	⋮	⋮

4. For Step1-4, the system reduces the frequency of search times by breaking process; when the number of elements included in an intersection of sets of $n-1$ anonymyzation patterns is less than the threshold K, the system breaks the process for calculating $\hat{k}$. If the number of elements is K or more, the system decrements n by 1. If n equals 1, the system breaks process. If not, the system returns to Step1-2.

Step 1-1. Generation of paired information

For each quasi-identifier in W, the system generates a paired information; a character and a position where the character appears. We define $\{c, i\}_j$ as paired information; a combination of i-th position of character c in w_j and the character c.

For example, where w_j="crew", $\{c, i\}_j$=$\{$"c'', 0$\}$, $\{$"r'', 1$\}$, $\{$"e'', 2$\}$, $\{$"w'', 3$\}$. Let $|w_j|$ be the length of w_j. Where $|w_j| = |$"$crew''| = 4$, the system extracts 4 paired information. We previously get paired information in this way.

Step 1-2. Extraction of effective pattern-matching targets using the paired information

In this step, the system extracts effective pattern-matching targets using paired information by step1-1. We define $T_{|w_id|,c,i}$ as a set of quasi-identifiers whose the length is $|w_{id}|$ and the paid information is $\{c, i\}$.

For example, where w_j="crew", the system extracts $T_{4,c,0}$, $T_{4,r,1}$, $T_{4,e,2}$ and $T_{4,w,3}$. The system does not extract w ="crawl" despite having {"c",0}, because $|w| \neq 4$.

As the elements of $T_{4,c,0}$, the system extracts the terms whose lengths are equal to 4 and the 0-th character is "c" such as "clew" and "cell." The number of elements in $T_{4,c,0}$ equals the count of hit terms in W for the candidate anonymized strings "c$***$" which is redacted except 0-th character "c".

Step 1-3. Calculation of the number of retrieved targets using set theory

The system needs to grasp the number of quasi-identifier which is anonymized in the same way, however it takes much time to calculate the number. The reason is because the system needs to search the quasi-identifier list using all anonymyzation patterns whose substrings redacted. Therefore, the problem is that retrieval targets includes unrelated terms.

Our proposed method reduces retrieval targets using the intersection of two sets of anonymyzation pattern; If the system retrieves anonymyzation patterns when $n-1$ characters of w_{id} are redacted, the system uses two sets of anonymyzation patterns when n characters of w_{id} are redacted.

For example, when we calculate the number of retrieved targets $\hat{k}$ for query "cr$**$," that is the number of elements in $T_{4,cr,01}$, we can calculate $\hat{k}$ by the intersection of $T_{4,c,0}$ and $T_{4,r,1}$.

The system proceeds to the next step, after the system calculates $\hat{k}$ for all anonymyzation patterns when $n-1$ characters of w_{id} is redacted.

Step 1-4. End condition

If the maximum $\hat{k}$ for anonymyzation patterns when $n-1$ characters of w_{id} is redacted is less than K, the system breaks the process.

For example, where K is 3, the system breaks the process if the maximum number of elements in $T_{4,cr,01}$, $T_{4,ca,02}$, $T_{4,cw,03}$, $T_{4,ra,12}$, $T_{4,rw,13}$ and $T_{4,aw,23}$ is 2. In this way, the system does not need to find anonymyzation pattern when 3 characters is redacted. If the maximum number of elements is K or more, the system decrements n by 1. And if n equals to 1, the system breaks process, and then the process returns to Step 1-2.

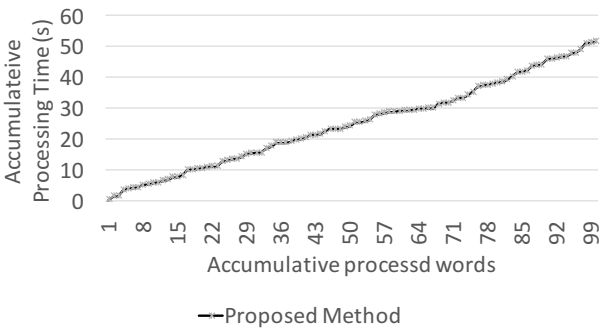

Figure 2: Accumulative processing time

4. EXPERIMENTAL EVALUATION

In this chapter, we verify whether our ideas accelerate the processing time of constructing the anonymization dictionary or not. Therefore we should evaluate the sum total of the processing time of extracting effective pattern-matching targets and the number of quasi-identifiers which are anonymized in the same way. However, due to the number of pages circumstances we only the former.

This time, we evaluate the processing time of extracting effective pattern-matching targets in constructing the anonymization dictionary.

4.1 Experimental Setup

We used the Japanese Wikipedia page title list (until January 1, 2016) which consists of $1,617,400$ page titles as the quasi-identifier list W. Then, we randomly pick up 100 titles whose length is between 3 and 15. These data are used as experimental data. We extract only effective pattern-matching targets for each experimental data in W.

4.2 Experimental Results and Discussions

Figure.2 shows the accumulative processing time vs. accumulative number of terms. In detail, mean processing time of our proposed method is 0.52 seconds, standard deviation is 0.47 seconds.

We consider the relationship between the length of strings and the processing time. Figure.3 shows a result. The processing time in proposed method does not necessarily depend on the length of string. For example, the processing time for "警察庁広域重要指定 113 号事件" (a case in Japan) is 0.07 seconds, and for "コンテンポラリー・アーティスト"(Contemporary Artist) is 1.88 seconds (the maximum processing time.) Proposed method repeatedly extracts total 9,047 terms for "コンテンポラリー・アーティスト" and total 794 terms for "警察庁広域重要指定 113 号事件." The latter has less terms than the former, so we consider that proposed method ran faster for the latter. On the other hand, for "スーパーサッカー (SUPER SOCCER: a TV program in Japan)," whose the string length 8 less than 15, the proposed method repeatedly ex-

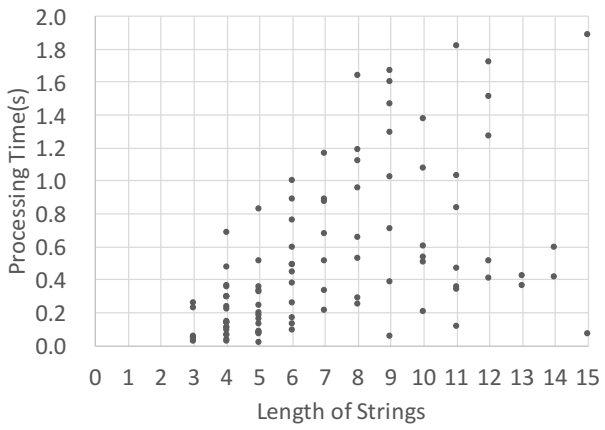

Figure 3: Processing time vs. length of strings

tracts total 25,701 terms. Due to that, the processing time is 1.63 seconds. It takes time in comparison to other terms.

Moreover, we describe the difference between "森小路'(a place name in Japan)' and "転型期不況 (the economic recession in japan from a period of high growth to one of stable growth.)" The processing time for the former is 0.04 seconds and for the latter is 0.014 seconds. The system repeatedly extracts total 1,172 terms for "森小路" as effective pattern-matching targets. On the other hand, the system repeatedly extracts total 276 terms for "転型期不況" less than for "森小路." We consider that the system ran faster for "転型期不況" whose the number of extraction is less than that of "森小路" regardless of the length of string.

These things suggest that the processing time in proposed method is related to the appearance frequency of a character.

5. CONCLUSION

We propose a method for anonymizing the quasi-identifier. Our method anonymizes quasi-identifiers to remain substrings which do not contribute to re-identification, in order to keep the information on the original texts as is.

The system selects the substrings which should be redacted to satisfy the following two conditions: 1) Any terms in the quasi-identifier list satisfies k-anonymity by redacting characters. 2) The number of redacted characters is minimized. However, the system cannot detect which substring to redact ideally if the system does not retrieve all patterns. Therefore, we constructed the anonymization dictionary which records the two number; the number of quasi-identifiers which are anonymized in the same way, and the number of redacted characters of the anonymized quasi-identifier in advance. Moreover we propose an acceleration method for constructing the anonymization dictionary using heuristics and set theory.

In future work, we should measure the processing time of construction of the anonymization dictionary and observe a trade-off relation between the two number. In addition, we should confirm whether our method can suppress the risks of re-identification or not by subject experiments. Then, based on the result of subject experiments, we will discover the mechanisms of predicting the original quasi-identifier from the anonymized quasi-identifier using contexts. Moreover, we will improve our method based on the discovered mechanisms.

6. REFERENCES

[1] J.-W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*, pages 188–200. Springer, 2007.

[2] K. Hara. Applying a svm based chunker and a text classifier to the deid challenge. In *i2b2 Workshop on challenges in natural language processing for clinical data*, pages 10–11. Am Med Inform Assoc, 2006.

[3] S.-H. Kim, D.-G. Kwon, and H.-G. Cho. Privacy-enhanced string matching with wordwise positional sampling. In *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, page 33. ACM, 2014.

[4] S. M. Meystre, F. J. Friedlin, B. R. South, S. Shen, and M. H. Samore. Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Medical Research Methodology*, 10(1):1–16, 2010.

[5] I. Neamatullah, M. M. Douglass, H. L. Li-wei, A. Reisner, M. Villarroel, W. J. Long, P. Szolovits, G. B. Moody, R. G. Mark, and G. D. Clifford. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8(1):1, 2008.

[6] H.-Q. Nguyen-Son, T. Minh-Triet, H. Yoshiura, N. Sonehara, and I. 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.

[7] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.

[8] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[9] G. Szarvas, R. Farkas, and R. Busa-Fekete. State-of-the-art anonymization of medical records using an iterative machine learning framework. *Journal of the American Medical Informatics Association*, 14(5):574–580, 2007.

[10] T. Venkatesan, H. Gupta, P. Roy, and M. K. Mohania. Efficient techniques for document sanitization. In *ACM Conference on Information and Knowledge Management*. Citeseer, 2008.