

t-Plausibility: Semantic Preserving Text Sanitization

Wei Jiang
Mummoorthy Murugesan
Chris Clifton
Luo Si

CSD TR #08-024
September 2008

t-Plausibility: Semantic Preserving Text Sanitization

Wei Jiang¹, Mummoorthy Murugesan², Chris Clifton³, Luo Si⁴

¹*Dept. of Computer Science, Missouri University of Science and Technology
500 W. 15th St, Rolla, MO 65409, USA. wjiang@mst.edu*

²⁻⁴*Dept. of Computer Science, Purdue University
305 N. University Street, West Lafayette, IN 47907, USA
{mmuruges, clifton, lsi}@cs.purdue.edu*

ABSTRACT

Text documents play significant roles in decision making and scientific research. Under federal regulations, documents (e.g., pathology records) containing personally identifiable information cannot be shared freely, unless properly sanitized. Generally speaking, document sanitization consists of finding and hiding personally identifiable information. The first task has received much attention from the research community, but the main strategy for the second task has been to simply remove personal identifiers. It is not hard to see that if important information (e.g., diagnoses and personal medical histories) is completely removed from pathology records, these records are no longer readable, and even worse, they no longer contain sufficient information for research purposes.

Observe that the sensitive information “tuberculosis” can be replaced with the less sensitive term “infectious disease”. That is, instead of simply removing sensitive terms, these terms can be hidden by more general but semantically related terms to protect sensitive information, without unnecessarily degrading the amount of information contained in the document. Based on this observation, the main contribution of this paper is to provide a novel information theoretic approach to text sanitization, develop efficient heuristics to sanitize text documents, and analyze possible attacks preventable under the proposed model.

1. INTRODUCTION

Medical documents, such as pathology records, play significant roles in detecting, verifying and monitoring new diagnostic examinations and treatment methodologies. However, under federal regulations, e.g., the Health Insurance Portability and Accountability Act (HIPAA) [6], because these records often contain sensitive or confidential information, they cannot be distributed freely. As a consequence, they cannot be used for medical research, e.g., to discover cures for life threatening diseases, unless properly sanitized.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGMOD '09 Providence, RI, USA

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

In general, document sanitization consists of two main tasks: (1) Identifying personally identifiable information, e.g., as defined by the HIPAA safe harbor rules, and (2) “hiding” the discovered identifiers. Unfortunately, medically relevant terms can often be identifying, for example conditions related to the disease (such as weight, which can assist in identification.) To truly sanitize documents requires hiding such relatively unique information, which likely goes beyond obvious identifiers.

The first task has received much attention from the research community, and many commercial products have been developed to detect personal identifiable attributes. As for the second task, the main approach adopted by current text sanitization techniques is to simply remove personal identifiers (names, dates, locations, diagnoses, etc.) to prevent re-identification of text documents. It is not hard to see that if diagnoses and personal medical histories are completely removed from pathology records, these records are no longer readable, and even worse, they no longer contain sufficient information to allow biomedical researchers to develop treatments for fatal diseases. This can be illustrated by the following example.

Suppose a phrase “Uses marijuana for pain” is contained in a medical report. The traditional techniques can sanitize this phrase by “blacking out” sensitive information, such as the drug used or diagnosis, turn the phrase into the meaningless “uses — for —”. This can cause sanitized texts to be no longer readable, and hence, document utility is unnecessarily degraded. More specifically, let d refer to the sample text in Figure 1(a), where **Sacramento**, **marijuana**, **lumbar pain** and **liver cancer** are the sensitive terms. Let d^* refer to the sanitized text in Figure 1(b), which is the result of removing sensitive words from d . Clearly, d^* is useless for analyzing disease epidemics. Let d^\dagger refer to the sanitized text in Figure 1(c), where sensitive words are replaced by more general terms (using the hypernym trees presented in Figure 2, where a word w in a given tree has a broader meaning than its children). d^\dagger contains much more information than d^* . However, it still protects sensitive information (removing specific identifying information as well as the sensitivity of the type of drug used) and preserves linguistic structure.

Observe that the sensitive information “tuberculosis” can be replaced with the less sensitive term “infectious disease”. That is, instead of simply removing sensitive terms, these terms can be hidden by more general but semantically related terms to protect sensitive information without unnecessarily degrading document utility (the amount of infor-

mation). Based on this observation, the overall objectives of this paper are: (1) provide an information theoretic approach to text sanitization, (2) develop efficient algorithms to sanitize text documents based on the proposed information theoretic measure, and (3) analyze possible attacks that the proposed text sanitization approach can prevent, from the perspective of existing privacy protection models.

A **Sacramento** resident purchased **marijuana** for the **lumbar pain** caused by **liver cancer**.

(a) Sample text

A ~~Sacramento~~ resident purchased ~~marijuana~~ for the ~~lumbar pain~~ caused by ~~liver cancer~~.

(b) Sanitized text

A **state capital** resident purchased **drug** for the **pain** caused by **carcinoma**.

(c) Semantic preserving sanitized text

Figure 1: A sample text and its sanitized versions

1.1 Problem Overview

To avoid unnecessary distortion, our view of text document security is as follows: given a threshold t and the set of word ontologies (e.g., hypernym trees), a sanitized text should be a plausible result of at least t base text documents. From this point of view, we will develop information theoretic measures and algorithms to sanitize text as shown in Figure 1(c). We make the following assumptions: Given a document and an (possibly domain specific) ontology, we can identify a set of sensitive words related to the document (e.g., using techniques developed in [1, 11, 14] or domain specific knowledge). In our problem domain, since sensitive words are the focal point, without loss of generality, we assume a piece of text d only contains sensitive values.

The rest of the paper is organized as follows: Section 2 summarizes relevant current work and points out the differences; Section 3 proposes an information theoretic measure according to the concept of t -plausibility and analyzes the hardness of t -plausibility based text sanitization problem; Section 4 proposes uniform t -plausibility measure and effective and efficient text sanitization heuristics; Section 5 discusses how t -plausibility prevents possible attacks by comparing existing privacy protection models and issues regarding sanitization of large documents; Section 6 validates our analyses via experimental results and Section 7 concludes the paper with future research directions.

2. RELATED WORK

The existing work most related to ours is data anonymization and text de-identification. The most common data anonymization technique is k -anonymity proposed in [12, 15]. Since then, there has been extensive work done related to anonymizing structured information, i.e., datasets of at

least k tuples in relational format. The proposed work here focuses on sanitizing a single text-based document without assuming access to a collection of related documents. Clearly the presence of multiple identical text documents about different individuals is unlikely; and modifying similar documents so they are identical while still preserving some semblance of semantics and readability requires a much deeper understanding of documents than Natural Language Processing is currently capable of. Therefore, applying k -anonymity directly is not feasible.

2.1 Text Anonymization

Much text anonymization work has mainly concentrated on de-identification of medical documents. The Scrub system [14] finds and replaces patterns of identifying information such as name, location, Social Security Number, medical terms, age, date, etc. The replacement strategy is to change the identified word to another word of similar type (e.g., an identified name is replaced with a fake name), and it is not clear whether the semantics of the reports themselves reveal the individuals. Similarly, [16] provides a six-step anonymization scheme that finds and replaces identifying words with pseudonyms from patient records in Norwegian.

In [4], the authors present schemes for removing protected health information (PHI) from free-text nursing notes. Since the text in this case does not follow the syntax of natural language construction, it poses challenges in recognition of PHI. The solution consists of techniques such as pattern matching, lexical matching, and heuristics to find the PHI from nursing notes. Moving away from purely syntactic based recognition of identifying information, MEDTAG [11], a specialized medical semantic lexicon, is used for finding personally identifying information in patient records. This system uses the semantic tags for disambiguation of words, and also relies on manually written disambiguation rules to differentiate between words that have different contextual meanings. The identifiers are then removed from the medical records.

To our knowledge, there exists very little work that addresses the general problem of text sanitization. A two-phase scheme that employs both sanitization and anonymization was proposed in [13]. The sanitization step uses automatic named entity extraction methods to tag the terms, and then replaces them with dummy values. This is equivalent to replacing the selected terms with their corresponding categories. The anonymization phase is defined based on k -anonymity and only applied on quasi-identifying words (i.e., words presumed to be combinable with certain external knowledge to possibly identify an individual). In [1], an ontological representation of text document is used to find and remove sensitive sentences. The pre-defined contextual restrictions act as the inputs, and also guide the sanitization procedure. This results in the sensitive sentences being removed from the document. Assuming the existence of an external database containing demographic information, suppression-based methods were introduced in [3] to sanitized documents such that the resulting documents cannot be linked to less than k records in the external database.

To summarize, existing work mainly focuses on how to identify sensitive words, and either remove them or replace them with pseudonyms. The replacement strategies lack a theoretic foundation, and consequently, without a formal measurement, it is difficult to judge the quality of sanitized documents. The work proposed here provides a theoretic

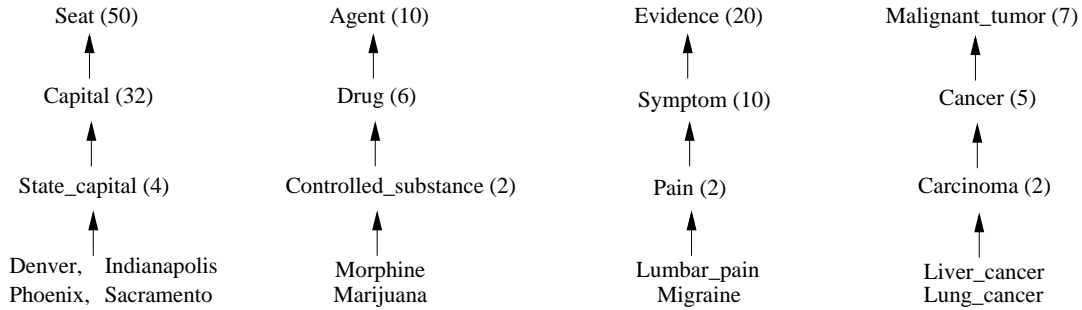


Figure 2: Word ontologies

measure on the quality of sanitized documents from a privacy protection point of view. These measures provide formal reasonings on how and why a more general term is chosen to replace sensitive information in a given document. In addition to the given document, the only information available to us is related word ontologies, or hypernym trees; and we do not use domain-specific information extraction techniques. We use WordNet [10] in our examples/experiments to retrieve word ontologies and generate hypernym trees.

3. T -PLAUSIBILITY TEXT SANITIZATION

Before presenting the concept of t -plausibility sanitization on text (t -PAT), we first introduce key notations and terminologies in Section 3.1. The formal definition of t -PAT is presented in Section 3.2. Then we prove that t -PAT is NP-hard in Section 3.3. At the end of this section, we present a pruning-based algorithm to find the optimal solution according to the definition of t -PAT.

3.1 Basic Notations and Terminologies

For the remaining of this paper, the terms sanitization (sanitized) and generalization (generalized) are interchangeable. The term “base text” refers a text that has not been sanitized in any way. Let d be a base text, \bar{d} be a sanitized text and $d[i]$ (or $\bar{d}[i]$) denote the i^{th} term in d (or \bar{d}) (a term is a word, or phrase recognized by the ontology; where we use “word” it could also be such a short phrase.) Because we merely consider sensitive words, most often, d represents the set of sensitive words in the original text. For example, suppose the text in Figure 1(a) is the original text, then we have $d = [\text{Sacramento, marijuana, lumbar_pain, liver_cancer}]$.

Definition 1 (Generalizable \triangleleft). We say that d is generalizable to \bar{d} (denoted as $d \triangleleft \bar{d}$) if $|d| = |\bar{d}|$ and $d[i] \triangleleft \bar{d}[i]$ for $1 \leq i \leq m$.

Since we only consider the base texts generalizable to some sanitized text, we always assume that $|d| = |\bar{d}|$. Next, we list additional notations adopted throughout the paper.

- $o = \{o_1, \dots, o_m\}$: Word ontology set represented as a set of word ontologies related to each word in d .
- $d = \{w_1, \dots, w_m\}$: A base text represented as a set of terms, where $|d| = m$, $w_i \in o_i$ for $1 \leq i \leq m$. w_i is equivalent to $d[i]$.
- $\bar{d} = \{\bar{w}_1, \dots, \bar{w}_m\}$: A generalized text represented as a set of terms, where $|\bar{d}| = m$, $\bar{w}_i \in o_i$ and $w_i \triangleleft \bar{w}_i$. \bar{w}_i is equivalent to $\bar{d}[i]$.

In most situations, a set is considered as an ordered set. For instance $o[i]$ is the i^{th} ontology of the i^{th} word in d (i.e., $d[i]$). Similarly, $d[i]$ can only be generalized to $\bar{d}[i]$. Figure 2 contains four word hypernym trees used extensively hereafter. The base values (sensitive values from the original text) are at the bottom of the ontologies. Values in the non-leaf nodes can be generalized to any values on its path to the root. For example, given the first ontology in Figure 2, Sacramento can be generalized to State_capital, Capital and so on. State_capital can be generalized to Capital, Seat, etc.

The integer value, termed as *volume*, in the parentheses indicates how many base values can be generalized to its related value. For instance, the value 32 associated with Capital (i.e., the volume of Capital) indicates that there are thirty-two base values (including the four values shown at leaves) that can be generalized to Capital. Note that the ontologies shown here are not the complete ontology trees, and some branches are omitted. The partial ontologies are extracted from WordNet, but for practicality of illustration, we do not use complete ontologies. Based on the previously introduced notations, we define the following functions:

1. $W(\bar{w}_i, d, \bar{d}, o) \rightarrow \{w_i^1, \dots, w_i^{k_i}\}$, the word domain function ($W(\bar{w}_i)$ for short):

- Pre-condition: $\bar{w}_i = \bar{d}[i]$ and $d \triangleleft \bar{d}$ according to o .
- Post-condition: $d[i] \in \{w_i^1, \dots, w_i^{k_i}\}$, $w_i^j \triangleleft \bar{w}_i$ for $1 \leq j \leq k_i$ and $w_i^1, \dots, w_i^{k_i}$ are base values in o_i .¹

Given a generalized word \bar{w}_i , it returns a set of all possible words at the same level of $d[i]$ that can be generalized to \bar{w}_i according to o .

2. $P_o(w'_i, \bar{w}_i, d, \bar{d}, o) \rightarrow Prob(w'_i \triangleleft \bar{w}_i)$, the local probability function ($P_o(w'_i, \bar{w}_i)$ for short):

- Pre-condition: $w'_i \in W(\bar{w}_i, d, \bar{d}, o)$
- Post-condition: The probability that given \bar{w}_i (or $d[i]$), w'_i is the original word.

Given a word w'_i in the domain of \bar{w}_i , it returns the probability that \bar{w}_i is generalized from w'_i .

3. $D(d, \bar{d}, o) \rightarrow \{W(\bar{w}_1, d, \bar{d}, o) \times \dots \times W(\bar{w}_m, d, \bar{d}, o)\}$, the text domain function ($D(\bar{d})$ for short):

- Pre-condition: $d \triangleleft \bar{d}$ according to o .

¹Base values are defined as the values from the based level of the ontology, and base level is defined by that of w_i in o_i .

- Post-condition: $\mathcal{D} = \{d_1, d_2, \dots, d_k\}$ and $d \in \mathcal{D}$, where $k = \prod_{i=1}^m k_i$ and $k_i = |W(\bar{w}_i, d, \bar{d}, o)|$.

Given a text d , its generalized counterpart \bar{d} and a set of word ontologies, the function returns a set of all possible texts that can be generalized to \bar{d} according to o . We call such set as the domain of d .

4. $P(d', d, \bar{d}, o) \rightarrow \text{Prob}(d' \triangleleft \bar{d})$, the global plausibility function ($P(d', \bar{d})$ for short):
 - Pre-condition: $d' \in D(d, \bar{d}, o)$
 - Post-condition: The probability that d is generalized from d' .

Given a text d' in the domain of d , the function returns the probability that d' can be generalized to \bar{d} . That is, P returns the probability that d' is the original text.

Example 1. Refer to Figure 2, if $\bar{w}_i = \text{controlled_substance}$, then $W(\bar{w}_i)$ returns {Ecstasy, Marijuana}. Assuming uniform distribution in $W(\bar{w}_i)$ and $w'_i = \text{marijuana}$, $P_o(w'_i, \bar{w}_i) = \frac{1}{|W(\bar{w}_i)|} = \frac{1}{2}$. Let $d = \{\text{marijuana}, \text{lumbar_pain}\}$ and $\bar{d} = \{\text{controlled_substance}, \text{pain}\}$, then $D(d, \bar{d}, o)$ returns $d_1 = \{\text{marijuana}, \text{lumbar_pain}\}$, $d_2 = \{\text{marijuana}, \text{migraine}\}$, $d_3 = \{\text{ecstasy}, \text{lumbar_pain}\}$ and $d_4 = \{\text{ecstasy}, \text{migraine}\}$. If we assume uniform distribution in both $W(\bar{d}[1])$ and $W(\bar{d}[2])$, for $1 \leq i \leq 4$, $P(d_i, \bar{d}) = \frac{1}{4}$. \square

3.2 Plausibility Sanitization on Text

Based on the previously introduced notations and terminologies, here we formally define our text sanitization problem. Define the sanitization function f as

$$f : d, t, o \rightarrow \bar{d}$$

which takes a text d , security parameter or threshold t and a set of ontologies o as the input and outputs \bar{d} . The security parameter basically restricts the set of possible outputs and is defined as follows:

Definition 2 (t -Plausibility). \bar{d} is t -plausible if at least t base texts (including d) can be generalized to \bar{d} based on o .

This definition simply says that a sanitized text \bar{d} can be associated with at least t texts, and any one of them could be the original text d . For instance, Let \bar{d} be the text in Figure 1(c). Based on the word ontologies in Figure 2, $|D(\bar{d})| = 96$, and we say that \bar{d} can be associated with 96 texts. If $t \leq 96$, \bar{d} satisfies the t -plausibility condition. When a text is sanitized properly, we should not be able to uniquely identify the original text. To prevent unique identification, there should exist more than one text that could be the base text. These texts are called plausible texts. The parameter t is defined as a lower bound on the number of plausible texts related to a given generalized text. t can also be considered as the degree of privacy that a sanitized text needs to guarantee.

Based on t , we define the text sanitization problem as an optimization problem. Since our intuition relies on the concept of t -plausibility, we term the text sanitization problem as t -Plausibility Sanitization on Text (t -PAT).

Definition 3 (t -PAT). Let t be a threshold, o be a set of word ontologies and d be a base text. The t -PAT problem is to find a sanitized text \bar{d} , such that \bar{d} is t -plausible and $|D(d, \bar{d}, o)|$ is minimal.

According to Definition 3, the t -PAT problem is to find a sanitization \bar{d} of d , such that $|D(d, \bar{d}, o)|$, is equal to t or the least upper bound of t . We next show that this text sanitization problem is NP-hard.

3.3 Hardness of t -PAT

Theorem 1. t -PAT defined in Definition 3 is NP-Hard.

Proof. The reduction is from the subset product problem, which is defined as follows: Given a set of integers I and a positive integer p , is there any non-empty subset $I' \subseteq I$ such that the product of numbers in I' equals p ? This problem is proven to be NP-Complete [5]. We now show a reduction from the subset product problem to t -PAT. Assume that there exists an algorithm \mathcal{A} that solves t -PAT in polynomial time. For each w_i ($1 \leq i \leq m$), define a set M_i containing the volumes of the terms along the path from w_i to \bar{w}_i , where $\bar{w}_i \in o_i$ and $w_i \triangleleft \bar{w}_i$. The number of possible \bar{w}_i is limited by the depth of the ontology. For example, refer to Figure 2, let w_i be the word *Sacramento* and \bar{w}_i be the word *capital*, then $M_i = \{1, 4, 8\}$. The input to \mathcal{A} are the sets M_1, \dots, M_m and the plausibility parameter t . The solution of t -PAT is a set of m numbers $\{n_1, \dots, n_m\}$ such that $n_i \in M_i$, and $\prod_{i=1}^m n_i$ is equal to the least upper bound of t .

The subset product problem can be solved using \mathcal{A} as follows. The input to the subset product problem is the set of integers $I = \{a_1, \dots, a_m\}$ and the product p . Construct m sets by creating $M_i = \{a_i, 1\}$ for $1 \leq i \leq m$ and invoke \mathcal{A} with inputs M_1, \dots, M_m and p . \mathcal{A} returns a set of m numbers $\{n_1, \dots, n_m\}$. If $\prod_{i=1}^m n_i = p$, the subset product has an answer. This can be obtained by looking at $\{n_1, \dots, n_m\}$ returned by \mathcal{A} . If $n_i = 1$, a_i is not included in the subset. On the other hand, if $n_i = a_i$, the subset contains the element a_i . Suppose $\prod_{i=1}^m n_i > p$, then there does not exist any subset in I such that the product of the subset is p . Otherwise, \mathcal{A} would have returned such a subset. Since the input and output transformations can be performed in polynomial time, we can conclude that t -PAT is NP-hard. \square

3.4 Exhaustive Search with Pruning Strategy

To solve t -PAT, we can simply enumerates all possible solutions and picks the best one. This can be easily accomplished by a recursive formulation. However, the exhaustive search is inefficient and intractable for large values of m . We present a pruning strategy that limits the search space to improve search efficiency. ESearch_Prune (Algorithm 1) is a recursive procedure to generate combinations of generalizations of a set of words $d = \{w_1, \dots, w_m\}$ with the given ontology $o = \{o_1, \dots, o_m\}$. The procedure takes a set \bar{d} (current generalization up to i^{th} word), the index i , the best value for t -PAT found so far as t_c and its corresponding generalization \bar{d}_c . When $i < m$, \bar{d} is a partial generalization on d . If $|D(\bar{d})| > t_c$ then any superset \bar{d}' of \bar{d} will be such that $|D(\bar{d}')| > t_c$. This observation guides the pruning process.

At step 2 of algorithm 1, h_i denotes the height of the hypernym tree o_i of word w_i , and \bar{w}_i^{+j} indicates the j^{th} generalization (or hypernym) of w_i on o_i in ascending order from w_i to the root of the tree. $w_i = \bar{w}_i^{+0}$ is a special case. If $i < m$, for each generalization of w_i from \bar{w}_i^{+j} to $\bar{w}_i^{+h_i}$, ESearch_Prune is called again with $i + 1$. Note that \bar{w}_i^{+j} is selected in an ascending order such that $\bar{w}_i^{+j} \triangleleft \bar{w}_i^{+(j+1)}$. The recursion terminates when i equals m . When this occurs,

the set \bar{d} is used to calculate $|D(\bar{d})|$. If this $|D(\bar{d})|$ is less than t_c , but greater than or equal to t then \bar{d} and $|D(\bar{d})|$ are returned as the best solutions. Otherwise, t_c and \bar{d}_c are returned. Algorithm 1 lists the steps of a pruning based recursive procedure. It is invoked as $\text{ESearch_Prune}(\emptyset, 1, \infty, \emptyset)$ and the returned values are the solutions to t -PAT.

Algorithm 1 $\text{ESearch_Prune}(\bar{d}, i, t_c, \bar{d}_c)$ - The Exhaustive Search with Pruning for t -PAT

Require: \bar{d} a set containing i generalized words, i an index, t_c the current least upper bound on t , \bar{d}_c a generalization of d whose $|D(\bar{d}_c)| = t_c$ and d, o, t are implicit parameters

- 1: **if** $i < m$ **then**
- 2: **for** $j = 0$ to h_i **do**
- 3: **if** $|D(\bar{d} \cup \{\bar{w}_i^{+j}\})| > t_c$ **then**
- 4: return (t_c, \bar{d}_c)
- 5: **end if**
- 6: $(t_c, \bar{d}_c) \leftarrow \text{ESearch_Prune}(\bar{d} \cup \{\bar{w}_i^{+j}\}, i + 1, t_c, \bar{d}_c)$
- 7: **if** $t_c = t$ **then**
- 8: return (t_c, \bar{d}_c)
- 9: **end if**
- 10: **end for**
- 11: **else**
- 12: **if** $t \leq |D(\bar{d})| < t_c$ **then**
- 13: return $(|D(\bar{d})|, \bar{d})$
- 14: **end if**
- 15: **end if**
- 16: return (t_c, \bar{d}_c)

4. T-PAT REVISITED

The optimal solution to the t -PAT problem defined in Definition 3 may not be the *best* solution in practice because it does not consider privacy protection of individual sensitive words. It is possible that an optimal solution comes from heavily generalizing only a few sensitive words. This can be illustrated by the following example.

Example 2. Refer to Figure 1 and Figure 2. Let d be the text in Figure 1(b). Suppose $t = 32$, then the optimal solution \bar{d} based on Definition 3 is:

A **capital** resident purchased **marijuana**
for the **lumbar pain** caused by **liver cancer**.

The volume of capital is 32 (there are 32 base values generalizable to capital.) This implies that there are 32 possible texts can be associated to \bar{d} . However, from a privacy preserving point of view, this \bar{d} does not protect privacy as well as the following generalized text:

A **state capital** resident purchased **drug**
for the **pain** caused by **carcinoma**.

A solution is to require that every sensitive word be protected equally. If most of the sensitive words are not generalized, then \bar{d} contains too much sensitive information. \square

As shown in the example, in practice, not only do we need to measure the quality of a generalized text \bar{d} using the threshold t , but also we need to consider how \bar{d} can preserve the privacy of every sensitive word. To achieve this goal, we next present an information theoretic measure based on the uniform plausibility assumption - every sensitive word needs to be protected equally.

4.1 Uniform t -Plausibility and an Information Theoretic Measure

Uniform plausibility implies that each sensitive word need to be protected unbiasedly. Under this uniform plausibility requirement, we can avoid situations where some words are generalized too much and other words are not generalized at all. To materialize uniform plausibility, we utilize the expected uncertainty of individual sensitive words as a measure. We use entropy to model this uncertainty and to accomplish uniform plausibility. Details are given next.

Let m be the number of words that need to be generalized in d , H be an entropy function and α be a system parameter governing the tradeoff between global optimality and uniform generalization. For $1 \leq i, j \leq m$, the cost function $\mathcal{C}(\bar{d}, t)$ is defined as:

$$\frac{\alpha}{m^2} (H(\bar{d}) - \log t)^2 + \frac{1 - \alpha}{m} \sum_{i=1}^m \left(H(\bar{w}_i) - \frac{\log t}{m} \right)^2 \quad (1)$$

The intuition behind is that the first term defines a global measure: how close the generalized text \bar{d} is to the expected uncertainty defined by t . The second term defines a local measure to achieve the uniform uncertainty (leading to uniform plausibility) among all sensitive words. $\frac{\log t}{m}$ is the expected entropy of each sensitive word when the text is properly generalized. Intuitively, the lower \mathcal{C} , the better each sensitive word is protected. Note that the denominators m^2 and m are used as scaling factors so that the two terms are relatively at the same scale. Detailed discussion on this issue is presented in Section 4.5.

Next we show how to calculate the entropies of \bar{w}_i and \bar{d} . $H(\bar{w}_i)$ can be calculate as:

$$H(\bar{w}_i) = - \sum_{j=1}^{k_i} P_o(w_i^j, \bar{w}_i) \log P_o(w_i^j, \bar{w}_i) \quad (2)$$

where $k_i = |W(\bar{w}_i)|$ is the number of words that can be generalized to \bar{w}_i . W is the word domain function and P_o is the local plausibility function. Both functions are defined in Section 3.1. Similarly, $H(\bar{d})$ can be calculated as follows:

$$H(\bar{d}) = - \sum_{i=1}^k P(d_i, \bar{d}) \log P(d_i, \bar{d}) \quad (3)$$

where $k = |D(\bar{d})|$. D is the text domain function and P is the global plausibility function. Both functions are defined in Section 3.1. If we assume that each word is independent. $P(d_i, \bar{d})$ can be calculated as follows:

$$P(d_i, \bar{d}) = \prod_{j=1}^m P_o(d_i[j], \bar{d}[j]) \quad (4)$$

Example 3. Let \bar{w}_i be the state_capital in Figure 2, and assume uniform distribution in $W(\bar{w}_i)$. We can compute $H(\bar{w}_i) = - \sum_{j=1}^4 \frac{1}{4} \log \frac{1}{4} = 2$. Let \bar{d} be the sanitized text in Figure 1(c) (i.e., $\bar{d} = \{\text{state_capital, drug, pain, carcinoma}\}$). Let d be the original text in Figure 1(b) (i.e., $d = \{\text{Sacramento, marijuana, lumbar_pain, liver_cancer}\}$). Assume uniform distribution in each \bar{w}_i (or $d[i]$). Then $P(d_i, \bar{d}) = \frac{1}{4} \cdot \left(\frac{1}{2}\right)^3 = \frac{1}{32}$, and since $P(d_i, \bar{d}) = \frac{1}{32}$ for $1 \leq i \leq 32$, $H(\bar{d}) = - \sum_{i=1}^{32} \frac{1}{32} \log \frac{1}{32} = 5$.

Let $\alpha = 0.5$ and $t = 32$. If $\bar{d} = \{\text{capital, marijuana,}$

lumbar_pain, liver_cancer},

$$\begin{aligned} \mathcal{C}(\bar{d}, t) &= \frac{1}{8} \sum_{i=1}^4 \left(H(\bar{w}_i) - \frac{5}{4} \right)^2 \\ &= \frac{1}{8} \left(\frac{15}{4} \right)^2 + \frac{3}{8} \left(-\frac{5}{4} \right)^2 \approx 2.33 \end{aligned}$$

Nevertheless, if $\bar{d}' = \{\text{state_capital, drug, pain, carcinoma}\}$,

$$\begin{aligned} \mathcal{C}(\bar{d}', t) &= \frac{1}{8} \sum_{i=1}^4 \left(H(\bar{w}_i) - \frac{5}{4} \right)^2 \\ &= \frac{1}{8} \left(\frac{3}{4} \right)^2 + \frac{3}{8} \left(-\frac{1}{4} \right)^2 \approx 0.09 \end{aligned}$$

Clearly, $\mathcal{C}(\bar{d}', t)$ is a much smaller cost than $\mathcal{C}(\bar{d}, t)$. This matches the intuition behind Equation 1 and implies that \bar{d}' is a better sanitized text than \bar{d} from a privacy protection perspective. Indeed, we can observe that \bar{d}' achieves uniform plausibility better than \bar{d} . \square

As mentioned before, the optimal solutions presented in Section 3.4 do not take into account the concept of uniform plausibility. In other words, if \bar{d} is optimal according to Definition 3, not all words in \bar{d} are equally protected. This was shown in Example 2. Whether or not uniform plausibility is achievable depends on the structure of hypernym trees. At least by minimizing \mathcal{C} , we can achieve some degree of uniform plausibility. Our objective here is defined by the following definition:

Definition 4 (Uniform t-PAT). Given a text d , a set of hypernym trees o (related to d), α value and a threshold t , find a \bar{d} of d , such that $H(\bar{d}) \geq \log t$ and $\mathcal{C}(\bar{d}, t)$ is minimized.

4.2 Search for Minimizing $\mathcal{C}(\bar{d}, t)$ with Pruning

The exhaustive search approach enumerates all the possible solutions and picks the generalization \bar{d} that minimizes the cost metric \mathcal{C} under the condition that $H(\bar{d}) \geq \log t$. We now present a pruning strategy that improves the efficiency of pure exhaustive search. Key steps are provided in Algorithm 2. Similar to Algorithm 1, the recursive procedure builds all possible combinations of generalizations of the words $d = \{w_1, \dots, w_m\}$ based on the set of hypernym trees $o = \{o_1, \dots, o_m\}$. The procedure takes a set \bar{d} (current generalization up to i^{th} word), the index i , the current minimum cost t_c and the generalization \bar{d}_c whose cost is equal to t_c . The end of recursion is reached when i equals m . When this occurs, the cost $\mathcal{C}(\bar{d}, t)$ of such generalization, as defined by \bar{d} , is computed. This cost is compared against the current minimum cost (t_c), and if the new cost is lower than t_c and $H(\bar{d}) \geq \log t$, the current generalization \bar{d} and its cost are returned.

When $i < m$, we apply the pruning criteria to decide whether further generalization would give us a better solution with a cost less than t_c . Let us assume that the cost metric (Equation 1) is represented as $\mathcal{C} = \mathcal{C}_1 + \mathcal{C}_2$, where \mathcal{C}_1 and \mathcal{C}_2 are the global and local measures respectively. Let \bar{d} be a partial anonymization on d . This implies that $|\bar{d}| < m$. The entropy of \bar{d} is calculated as $H(\bar{d})$ by the Equation 3. Let \bar{d}' be a generalization and also be a superset of \bar{d} . Thus we have $\bar{d} \subset \bar{d}'$, $|\bar{d}| < |\bar{d}'| \leq m$ and $H(\bar{d}') \geq H(\bar{d})$. We observe the following properties on the cost function.

Observation 1 (Monotonicity property of \mathcal{C}_2). Given $\bar{d} \subset \bar{d}'$, we have the following condition on the the second term of the cost metric: $\mathcal{C}_2(\bar{d}', t) \geq \mathcal{C}_2(\bar{d}, t)$.

Proof. Write $\mathcal{C}_2(\bar{d}', t) = \mathcal{C}_2(\bar{d}, t) + \delta$. Since $(H(\bar{w}_i) - \frac{\log t}{m})^2$ cannot be negative, we have $\delta \geq 0$. This implies that $\mathcal{C}_2(\bar{d}', t) \geq \mathcal{C}_2(\bar{d}, t)$. \square

This observation can be used to stop considering the supersets of a partial anonymization \bar{d} when $\mathcal{C}_2(\bar{d}, t)$ is greater or equal to the minimum cost (t_c) found so far. Given that $\mathcal{C}_1(\bar{S}, t) \geq 0$, if $\mathcal{C}_2(\bar{d}) \geq t_c$ then the cost of a complete generalization consisting of \bar{d} must exceed t_c .

Observation 2 (Monotonic property of Cost metric). If $H(\bar{d}) \geq \log t$, then the cost function $\mathcal{C}(\bar{d}', t) \geq \mathcal{C}(\bar{d}, t)$.

Proof. This basically says that the cost metric of a superset (\bar{d}') of a generalization (\bar{d}) is greater than or equal to the cost metric of its subset \bar{d} when $H(\bar{d}) \geq \log t$. We establish this by considering the first and second terms of \mathcal{C} individually.

- The first term of the cost metric for \bar{d} is defined as $\mathcal{C}_1 = (H(\bar{d}) - \log t)^2$ (ignoring the coefficient). Since $H(\bar{d}) \geq \log t$ and $H(\bar{d}') \geq H(\bar{d})$, we have $\mathcal{C}_1(\bar{d}', t) \geq \mathcal{C}_1(\bar{d}, t)$.
- From Observation 1, we know that $\mathcal{C}_2(\bar{d}', t) \geq \mathcal{C}_2(\bar{d}, t)$.

Given the above properties of individual terms, it is clear that $\mathcal{C}(\bar{d}', t) \geq \mathcal{C}(\bar{d}, t)$. \square

This observation is used to stop considering the supersets of a partial anonymization \bar{d} when $\mathcal{C}(\bar{d}, t)$ is greater than the current minimum cost t_c (given $H(\bar{d}) \geq \log t$).

The pruning strategies avoid generating combinations clearly worse than the best solution. Algorithm 2 lists the steps of a pruning based recursive procedure. This procedure is invoked as `Uniform_ESearch_Prune($\emptyset, 1, \infty, \emptyset$)` and the returned values minimize the cost function \mathcal{C} . The pruning condition at steps 3-5 is based on Observation 2, and the pruning condition at steps 7-9 is based on Observation 1. The notation h_i and \bar{w}_i^{+j} introduced in Section 3.4 indicate the height of o_i and the j^{th} generalization of w_i .

4.3 Proposed Heuristics

Exhaustive search based algorithms are not practical in the average case. Thus, in this section, we propose three heuristics to generate sanitized texts that possess the property of uniform plausibility. Although the proposed heuristics are simple, they are intuitive, and most importantly, we also provide an upper bound on the worst case scenario according to the cost function \mathcal{C} . Since to use the cost function \mathcal{C} , we need to know the $P_o(w'_i, \bar{w}_i)$ value for each w'_i in $W(\bar{w}_i)$, for the rest of this section, we assume that words are uniformly distributed in each $W(\bar{w}_i)$.² Let \bar{w}_i^- and \bar{w}_i^+ be the immediate hyponym and hypernym of \bar{w}_i on the hyponym tree respectively.

LUB_Search (Algorithm 3) consists of two main steps: finding an upper bound on \mathcal{C} , and performing greedy search to improve the upper bound cost. Steps 1-3 of Algorithm 3 find a generalized text \bar{d} such that the optimal cost is always less than or equal to $\mathcal{C}(\bar{d}, t)$. Steps 5-7 check the

²In later sections, we will propose a way to assign non-uniform probability for each word in $W(\bar{w}_i)$.

Algorithm 2 Uniform_ESearch_Prune($\bar{d}, i, t_c, \bar{d}_c$) - Search for minimizing \mathcal{C} with Pruning

Require: \bar{d} a partial generalization on d containing i words, i an index, t_c current minimum cost, \bar{d}_c a partial generalization whose cost is t_c and d, t, o implicit parameters

- 1: **if** $i < m$ **then**
- 2: **if** $H(\bar{d}, t) \geq \log t$ **then**
- 3: **if** $\mathcal{C}(\bar{d}, t) > t_c$ **then**
- 4: return (t_c, \bar{d}_c)
- 5: **end if**
- 6: **end if**
- 7: **if** $\mathcal{C}_2(\bar{d}, t) > t_c$ **then**
- 8: return (t_c, \bar{d}_c)
- 9: **end if**
- 10: **for** $j = 0$ to h_i **do**
- 11: $(t_c, \bar{d}_c) \leftarrow$ Uniform_ESearch_Prune($\bar{d} \cup \bar{w}_i^{+j}, i + 1, t_c, \bar{d}_c$)
- 12: **end for**
- 13: **else**
- 14: **if** $\mathcal{C}(\bar{d}, t) < t_c$ **and** $H(\bar{d}, t) \geq \log t$ **then**
- 15: return $(\mathcal{C}(\bar{d}, t), \bar{d})$
- 16: **end if**
- 17: **end if**
- 18: return (t_c, \bar{d}_c)

Algorithm 3 LUB_Search(d, t, o, δ) - The Least Upper Bound Search for uniform t -PAT

Require: A base document d , a threshold t , a set of hypernym trees o and δ a greedy search threshold that limits the maximum number of deviations allowed

- 1: **for all** $w_i \in d$ **do**
- 2: Find a \bar{w}_i and $H(\bar{w}_i) = \lceil \frac{\log t}{m} \rceil$
- 3: **end for**
- 4: $c \leftarrow \mathcal{C}(\bar{d}, t)$
- 5: **if** $c = 0 \vee \delta = 0$ **then**
- 6: return \bar{d}
- 7: **end if**
- 8: $(c, \bar{d}) \leftarrow$ MStep_Greedy_Search($\delta, \bar{d}, c, \bar{d}$)
- 9: return \bar{d}

condition $\mathcal{C}(\bar{d}, t) = 0$. If the condition holds, we know that \bar{d} is the best possible solution, and no further computation is needed. The condition $\delta = 0$ (δ is the maximum number of deviations allowed in greedy search) indicates that no greedy search is performed and the current \bar{d} is returned. If neither condition holds, the algorithm will continue to the greedy search phase. The procedure MStep_Greedy_Search at step 8 of Algorithm 3 returns a generalized text that is either the same as \bar{d} or a better generalized document according to Equation 1.

The key steps of MStep_Greedy_Search are provided in Algorithm 4. In this procedure, the search space is constrained around the generalized text \bar{d} with the number of steps confined by the parameter δ . Our intuition is that the optimal solution is not far from \bar{d} (computed at steps 1-3). For instance, if $\delta = 1$, MStep_Greedy_Search exhaustively checks all possible one-step deviations \bar{d}_c of \bar{d} , and returns the one with the lowest cost and at the same time the condition $H(\bar{d}_c) \geq \frac{\log t}{m}$ must hold. Let \bar{w}_i^+ and \bar{w}_i^- be the immediate hypernym and hyponym of \bar{w}_i on the hypernym tree o_i respectively. One-step deviation on \bar{d} means that we

Algorithm 4 MStep_Greedy_Search($\delta, \bar{d}, t_c, \bar{d}_c$)

Require: δ a limit on the maximum number of deviations allowed on \bar{d} , \bar{d} a text or a generalized text in the form of $\{\bar{w}_1, \dots, \bar{w}_m\}$, t_c indicating the best current value of \mathcal{C} , and \bar{d}_c the generalized text whose cost is t_c

- 1: **for** $i = 1$ to $|\bar{d}|$ **do**
- 2: $\bar{d}' \leftarrow \bar{d}_{-i} \cup \{\bar{w}_i^+\}$
- 3: **if** $\mathcal{C}(\bar{d}', t) < t_c$ and $H(\bar{d}') > \log t$ **then**
- 4: $t_c \leftarrow \mathcal{C}(\bar{d}', t)$
- 5: $\bar{d}_c \leftarrow \bar{d}'$
- 6: **end if**
- 7: $\bar{d}' \leftarrow \bar{d}_{-i} \cup \{\bar{w}_i^-\}$
- 8: **if** $\mathcal{C}(\bar{d}', t) < t_c$ and $H(\bar{d}') > \log t$ **then**
- 9: $t_c \leftarrow \mathcal{C}(\bar{d}', t)$
- 10: $\bar{d}_c \leftarrow \bar{d}'$
- 11: **end if**
- 12: **end for**
- 13: **if** $\delta = 1$ **then**
- 14: return (t_c, \bar{d}_c)
- 15: **end if**
- 16: **for** $i = 1$ to $|\bar{d}|$ **do**
- 17: $(t_c, \bar{d}_c) \leftarrow$ MStep_Greedy_Search($\delta - 1, \bar{d}_{-i} \cup \{\bar{w}_i^+\}, t_c, \bar{d}_c$)
- 18: $(t_c, \bar{d}_c) \leftarrow$ MStep_Greedy_Search($\delta - 1, \bar{d}_{-i} \cup \{\bar{w}_i^-\}, t_c, \bar{d}_c$)
- 19: **end for**
- 20: return (t_c, \bar{d}_c)

can generate two new generalized texts by simply changing one of \bar{w}_i (or $\bar{d}[i]$) values to \bar{w}_i^+ or \bar{w}_i^- . Let \bar{d}_{-i} denote $\bar{d} - \{\bar{w}_i\}$; that is,

$$\bar{d}_{-i} = \{\bar{w}_1, \dots, \bar{w}_{i-1}, \bar{w}_{i+1}, \dots, \bar{w}_m\}$$

Both $\bar{d}_{-i} \cup \{\bar{w}_i^+\}$ and $\bar{d}_{-i} \cup \{\bar{w}_i^-\}$ are one-step deviations of \bar{d} . Base on the above description, when $\delta = 1$, the greedy strategy MStep_Greedy_Search generates $2m$ deviations of \bar{d} , and returns the best one.

On the other hand, when $\delta = 2$, the MStep_Greedy_Search algorithm exhaustively checks all possible two-step deviations \bar{d}_c of \bar{d} , and returns the one with the lowest cost. Similar to the one-step deviation, through two-step deviation we can generate a new generalized text from \bar{d} by making at most two changes. For instance, we can generalize or de-generalize³ \bar{w}_i twice, or we can generalize or de-generalize \bar{w}_i and \bar{w}_j once each. The outcome of two-step deviation include those of one-step deviation. For the case of $\delta = 2$, MStep_Greedy_Search generates $2m^2$ candidates. In general, MStep_Greedy_Search generates

$$\sum_{j=1}^{\delta} 2^j \cdot \binom{m}{j}, \text{ for } 1 \leq \delta \leq m$$

deviations of \bar{d} . The parameter δ determines the additional cost for finding possibly a better solution than \bar{d} . The size of δ is determined by the user.

Steps 1-12 of MStep_Greedy_Search check all one-step deviations of \bar{d} and returns the best one if $\delta = 1$. When $\delta > 1$, steps 16-19 will be executed to generate deviations of more than one steps. For succinctness, some steps are omitted

³generalize means moving up in the hypernym tree and de-generalize means moving down in the hypernym tree

Algorithm 5 One_Step_Alternative_Search(\bar{d}, d, t)

Require: \bar{d} a generalized text related to d and t is the privacy threshold

- 1: $m \leftarrow |\bar{d}|$
- 2: $t_c \leftarrow \mathcal{C}(\bar{d}, t)$
- 3: **for** $i = 1$ to m **do**
- 4: **for** $j = 1$ to m **do**
- 5: $\bar{d}' \leftarrow \bar{d}_{-j} \cup \{\bar{w}_i^+\}$
- 6: **if** $\mathcal{C}(\bar{d}', t) < t_c$ and $H(\bar{d}') > \log t$ **then**
- 7: $t_c \leftarrow \mathcal{C}(\bar{d}', t)$
- 8: $\bar{d}_c \leftarrow \bar{d}'$
- 9: **end if**
- 10: $\bar{d}' \leftarrow \bar{d}_{-j} \cup \{\bar{w}_i^-\}$
- 11: **if** $\mathcal{C}(\bar{d}', t) < t_c$ and $H(\bar{d}') > \log t$ **then**
- 12: $t_c \leftarrow \mathcal{C}(\bar{d}', t)$
- 13: $\bar{d}_c \leftarrow \bar{d}'$
- 14: **end if**
- 15: **end for**
- 16: $\bar{d} = \bar{d}_c$
- 17: **end for**
- 18: **return** (t_c, \bar{d}_c)

before the recursive calls. For instance, we need to check if \bar{w}_i is on top or on the bottom of the hypernym tree, and hypernym trees also need to be modified. We address these issues in the implementation.

When δ is small, one disadvantage of MStep_Greedy_Search is that many possible close deviations of \bar{d} are not searched. Since the optimal solution is expected to be very close to the upper bound in space, we introduce another heuristic, One_Step_Alternative_Search, that searches for the best solution closely around \bar{d} . This heuristic can replace the MStep_Greedy_Search strategy directly in Algorithm 3. Main steps of One_Step_Alternative_Search are given in Algorithm 5. During each iteration (the outer for-loop), $2m$ possible one-step derivations from \bar{d} are generated, and the one with the best cost is chosen to replace \bar{d} before next iteration.

The two greedy search strategies are interchangeable in the LUB_Search. Regardless of which strategy is adopted, the LUB_Search algorithm always guarantees the following:

Theorem 2 (Upper Bound on the LUB_Search algorithm). Suppose that $\bar{d} = \text{LUB_Search}(d, t, o)$. Then $H(\bar{d}) \geq \log t$ and $\mathcal{C}(\bar{d}, t) - C^* \leq \sigma^2$, where C^* is any minimal attainable cost according to Equation 1 and σ is a limiting factor on the size of hypernym tree.

Before proving Theorem 2, we need the following lemmas about the entropy of \bar{w}_i and the entropy of \bar{d} .

Lemma 1. If words in $W(\bar{w}_i)$ are uniformly distributed, then $H(\bar{w}_i) = \log |W(\bar{w}_i)|$.

Proof. Since we assume words in $W(\bar{w}_i)$ are uniformly distributed, for any word $w_i^j \in W(\bar{w}_i)$, $P_o(w_i^j, \bar{w}_i) = \frac{1}{|W(\bar{w}_i)|}$.

Let $k_i = |W(\bar{w}_i)|$, and $H(\bar{w}_i)$ can be rewritten as follows:

$$\begin{aligned} H(\bar{w}_i) &= - \sum_{j=1}^{k_i} P_o(w_i^j, \bar{w}_i) \log P_o(w_i^j, \bar{w}_i) \\ &= - \sum_{j=1}^{k_i} \frac{1}{|W(\bar{w}_i)|} \log \frac{1}{|W(\bar{w}_i)|} \\ &= - \log \frac{1}{|W(\bar{w}_i)|} = \log |W(\bar{w}_i)| \end{aligned}$$

□

Lemma 2. If words in $W(\bar{w}_i, o)$ are uniformly distributed and $\bar{d} = \text{LUB_Search}(d, t, o)$, then $H(\bar{d}) = \sum_{i=1}^m H(\bar{w}_i)$.

Proof. Assuming uniform distribution in $W(\bar{w}_i)$ and from previous analysis, we can rewrite $P(d_i, \bar{d})$ as $P(d_i, \bar{d}) = \prod_{j=1}^m \frac{1}{|W(\bar{w}_j)|}$. Let $k = \prod_{i=1}^m k_i$, and $H(\bar{d})$ is computed as:

$$\begin{aligned} H(\bar{d}) &= - \sum_{i=1}^k P(d_i, \bar{d}) \log P(d_i, \bar{d}) \\ &= - \sum_{i=1}^k \left\{ \prod_{j=1}^m \frac{1}{|W(\bar{w}_j)|} \left(\log \prod_{j=1}^m \frac{1}{|W(\bar{w}_j)|} \right) \right\} \\ &= - \frac{k}{\prod_{j=1}^m |W(\bar{w}_j)|} \log \prod_{j=1}^m \frac{1}{|W(\bar{w}_j)|} \\ &= - \log \prod_{j=1}^m \frac{1}{|W(\bar{w}_j)|} = - \sum_{j=1}^m \log \frac{1}{|W(\bar{w}_j)|} \\ &= \sum_{j=1}^m \log |W(\bar{w}_j)| = \sum_{j=1}^m H(\bar{w}_j) \end{aligned}$$

□

Now, we can prove Theorem 2 using Lemma 1 and Lemma 2. First we prove the condition $H(\bar{d}) \geq \log t$ and then we prove the second condition $\mathcal{C}(\bar{d}, t) - C \leq \sigma^2$.

Theorem 2. Refer to Algorithm 3. Steps 11-17 attempt to find a generalized text that has lower cost than \bar{d} (computed at steps 1-3). In the worst case scenario, assume the \bar{d} is returned from LUB_Search(d, t, o). Then we have that for any $1 \leq i \leq m$, $H(\bar{w}_i) \geq \frac{\log t}{m}$. This implies $H(\bar{w}_i) - \frac{\log t}{m} \geq 0$. Then the following inequality holds:

$$\begin{aligned} \sum_{i=1}^m \left(H(\bar{w}_i) - \frac{\log t}{m} \right) &\geq 0 \\ \sum_{i=1}^m H(\bar{w}_i) - \sum_{i=1}^m \frac{\log t}{m} &\geq 0 \\ \sum_{i=1}^m H(\bar{w}_i) &\geq \log t \end{aligned}$$

According to Lemma 2, $H(\bar{d}) = \sum_{i=1}^m H(\bar{w}_i)$. Therefore, $H(\bar{d}) \geq \log t$. Now let C^* be a minimal attainable cost according to Equation 1. Because we want to derive an upper bound on $\mathcal{C}(\bar{d}, t)$, let $C^*(\bar{d}^*, t) = 0$ and $H(\bar{w}_i) > \frac{\log t}{m}$. $C^*(\bar{d}^*, t)$ can be written as:

$$\frac{\alpha}{m^2} (H(\bar{d}^*) - \log t)^2 + \frac{1-\alpha}{m} \sum_{i=1}^m \left(H(\bar{w}_i^*) - \frac{\log t}{m} \right)^2 \quad (5)$$

Let assume the number of words that can be generalized to \bar{w}_i grows exponentially with base 2^σ as the height of the hypernym tree. (σ is the limiting factor and in practice, $\sigma = 3$ is sufficient.) For instance, assuming \bar{w}_i is at level h of the hypernym tree o_i , then the size of $W(\bar{w}_i)$ is about $2^{\sigma h}$. On the other hand, if \bar{w}'_i is at level $h + 1$ of o_i , then the size of $W(\bar{w}'_i)$ is about $2^{\sigma(h+1)}$. Again if we assume uniform distribution in both $W(\bar{w}_i)$ and $W(\bar{w}'_i)$, $H(\bar{w}'_i) = H(\bar{w}_i) + \sigma$. Because each \bar{w}_i in \bar{d} is at most one level above \bar{w}^*_i in \bar{d}^* , $H(\bar{w}_i) = H(\bar{w}^*_i) + \sigma$ in the worst case. Based on Lemma 2 and Equation 1, the first component of $\mathcal{C}(\bar{d}, t)$ can be calculated as follows:

$$\begin{aligned} \frac{\alpha}{m^2} (H(\bar{d}) - \log t)^2 &= \frac{\alpha}{m^2} \left(\sum_{i=1}^m H(\bar{w}_i) - \log t \right)^2 \\ &= \frac{\alpha}{m^2} \left(\sum_{i=1}^m (H(\bar{w}^*_i) + \sigma) - \log t \right)^2 \\ &= \frac{\alpha}{m^2} \left(\sum_{i=1}^m H(\bar{w}^*_i) - \log t + m\sigma \right)^2 \\ &= \frac{\alpha}{m^2} \bullet m^2 \sigma^2 = \alpha \sigma^2 \end{aligned}$$

Similarly, the second component of $\mathcal{C}(\bar{d}, t)$ can be calculated as follows:

$$\begin{aligned} \frac{1-\alpha}{m} \sum_{i=1}^m \left(H(\bar{w}^*_i) - \frac{\log t}{m} + \sigma \right)^2 &= \frac{1-\alpha}{m} \bullet m \sigma^2 \\ &= (1-\alpha) \sigma^2 \end{aligned}$$

Therefore, $\mathcal{C}(\bar{d}, t) - \mathcal{C}^* = \alpha \sigma^2 + (1-\alpha) \sigma^2 = \sigma^2$. Since we assume $\mathcal{C}^* = 0$ and 0 is the best attainable minimal cost, for any minimal cost \mathcal{C}^* , the equality $\mathcal{C}(\bar{d}, t) - \mathcal{C}^* \leq \sigma^2$ holds. \square

4.4 Complexity Analysis of LUB_Search

Since LUB_Search consists of two phases, we analyze the complexity of each phase independently. Assume the height of every hypernym tree is bounded by h . The main cost of the first phase (steps 1-3 of Algorithm 3) is to find the upper bound. The upper bound for each word can be found in $\log h$ steps, so the complexity of the first phase is bounded by $O(m \log h)$.

The complexity of the second phase is determined by either MStep_Greedy_Search or One_Step_Alternative_Search. As analyzed in Section 4.3, MStep_Greedy_Search's complexity is bounded by $O(m^\delta)$. Therefore, complexity of LUB_Search is $O(m \log h + m^\delta)$. In practice, the height of hypernym trees is constant, and the complexity of LUB_Search can be rewritten as $O(m + m^\delta)$. Due to the fact that the complexity of One_Step_Alternative_Search is $O(m^2)$, the complexity of LUB_Search is bounded by $O(m^2)$ if we use One_Step_Alternative_Search as the greedy strategy.

4.5 The Cost Function \mathcal{C} Revisited

In Equation 1, α determines the importance each term plays in evaluating the cost. Thus, the two terms should have similar scales or ranges; otherwise, it is hard to choose a suitable value for α . Having this in mind, we normalize the first term with m^2 . This makes sense, especially when words are uniformly distributed in each $W(\bar{w}_i)$. Under uniformity assumption, $H(\bar{d}) = \sum_{i=1}^m H(\bar{w}_i)$ (Lemma 2). Let $H(\bar{w}) =$

$\frac{1}{m} \sum_{i=1}^m H(\bar{w}_i)$, the first term can be written as:

$$\begin{aligned} \frac{\alpha}{m^2} (H(\bar{d}) - \log t)^2 &= \frac{\alpha}{m^2} m^2 \left(H(\bar{w}) - \frac{\log t}{m} \right)^2 \\ &= \alpha \left(H(\bar{w}) - \frac{\log t}{m} \right)^2 \end{aligned}$$

Suppose $\sum_{i=1}^m (H(\bar{w}_i) - \frac{\log t}{m})^2 \approx m \cdot (H(\bar{w}^*) - \frac{\log t}{m})^2$ for some value $H(\bar{w}^*)$, then the second term of Equation 1 can be written as:

$$\frac{1-\alpha}{m} \sum_{i=1}^m \left(H(\bar{w}_i) - \frac{\log t}{m} \right)^2 = (1-\alpha) \left(H(\bar{w}^*) - \frac{\log t}{m} \right)^2$$

As shown in the above analyses, by using m^2 and m as the normalizing factors, the first and the second terms of Equation 1 have comparable scales or value ranges. As a result, it is reasonable to use α as a single parameter to adjust the significance of the two terms in the cost function.

5. PRIVACY PROTECTION AND OTHER PRACTICAL ISSUES

From a privacy protection point of view, how is t -plausibility different from other text sanitization and data anonymization techniques against possible attacks?

5.1 Text Sanitization

Since the t -PAT is designed to sanitize text in general, it can be adopted in any text or document sanitization process. As emphasized before, instead of removing sensitive words or replacing sensitive words with synonyms, t -PAT generalizes the sensitive words with a well-defined information theoretic measure. If no domain specific word ontologies or hypernym trees are available, WordNet can be adopted. More specifically, we show how to use t -PAT to solve the problem presented in [3], where the privacy protection constraint is that sanitized documents cannot be linked to fewer than k records in an external database.

The ERASE protocol proposed in [3] assumes that the external database stores demographic information and each record (in relational format) is associated with an individual. A record r is associated with a given document d (a medical report) if they have common words. ERASE sanitizes d by removing some common words according to certain rules to achieve the aforementioned privacy constraint. Without loss of generality, assume medical report d is associated with one patient. t -PAT can achieve the same privacy constraint as follows: first build hypernym trees for each attribute of the external database. Secondly, the sensitive words in d can be identified as the set of common words between d and some record r in the external database. By setting $t = k$ in t -PAT, we can generalize or sanitize d using the hypernym trees. According to the definition of t -plausibility, the resulting document clearly satisfies the privacy constraint.

5.2 k -Anonymity

k -anonymity was proposed in [12, 15] to sanitize structured data. As stated in Section 2, existing k -anonymization techniques do not fit well with our problem domain, as the diversity in text (even short segments of text) is such that the likelihood that k text fragments belonging to different individuals would be semantically similar and grammatically

equivalent is low. Either semantics would be lost, or the solution would require a full parsing and understanding of semantics so as to generate an common anonymized text that corresponds semantically to the k (possibly grammatically diverse) texts. However, t -PAT can be adopted to anonymize structured data and the resulting data still preserves the spirit of the privacy protection guaranteed by k -anonymity. Next we provide detailed analysis from an information theoretic perspective, and assume k -anonymization is achieved via generalization.

Let X denote a dataset, Y denote the k -anonymous dataset computed from X , and $x \triangleleft y$ denote that $x \in X$ is directly generalized to $y \in Y$. In general, k -anonymity can be generalized to the following theorem [8]:

Theorem 3. Y achieved through generalization satisfies k -anonymity if and only if $\forall y \in Y, P(x \triangleleft y) \leq \frac{1}{k}$, i.e., the probability that y is generalized from x is no bigger than $\frac{1}{k}$.

For generalization based k -anonymization, a set of hypernym trees are generally given. Using these hypernym trees, t -PAT can anonymize X to Y . If Y is the result of traditional k -anonymization techniques, then given x , there is only one y such that $x \triangleleft y$. On the other hand, if Y is the result of t -PAT, then it is possible that x can be generalized to multiple y 's. According to Theorem 3, this does not violate the privacy achieved through k -anonymity. Certain extensions to k -anonymity exist, such as l -diversity [9] among others. Analysis of how these ideas can be incorporated into text anonymization and t -PAT is left for future work.

5.3 From Text to Document

In general, short texts have limited number of sensitive words. Can we apply the proposed approaches to sanitize a document? The answer is positive. However, some cautions are needed. First, since document length varies, choosing a value for t is difficult if we treat the document as a very large piece of text. Also, to achieve uniform plausibility with the same t value for all sensitive words in the document may not be desirable because the degree of sensitivity may vary from word to word. A more natural way to sanitize the document is to break it into text segments. E.g., we can use sentence, paragraph or section as a unit. Then sensitive words can be identified and sanitize using various t values.

Identifying sensitive words is a challenging but a separate problem. There are frameworks that have been proposed to solve the problem [1, 4, 14, 16]. These techniques are domain specific, and additional documents may be required to train the learning algorithms. Also, we can use taggers (e.g., Brill Tagger [2]) to tag the text to identify the nouns, since nouns play significant roles in interpreting the meaning of a natural language. We can treat most nouns as sensitive words if no other options are available. Moreover, word sense disambiguation techniques [7] may be needed to effectively use WordNet because a word can have multiple hypernyms.

6. EMPIRICAL ANALYSES

In the experimental analysis, we mainly validate the theoretic results presented in the previous sections. As stated in Section 5.3, we do not intend to directly use our proposed techniques on sanitizing large documents, and our experimental setting fits this purpose. There are two aspects of performance of the proposed schemes that we are most interested in: accuracy of the heuristics and the running time.

For the accuracy evaluation, we measure the difference in generalization that is found by the exhaustive search and the one found by the heuristic searchers. For the running time evaluation, we measure the gain by pruning-based search and heuristic search schemes over the exhaustive search.

6.1 Data Description

For the purpose of experiments, a collection of 50 words were selected randomly from the Wordnet tree hierarchy. The chosen words are the leaf nodes falling under the “entity” tree node where there are 30,000 possible words. These words were selected such that the height of individual word hypernym tree is close (± 1) to 8 (the average height of hypernym tree under “entity”). The words were subjected to only this height constraint. Wordnet tree structures are kept for all the selected words. If a word is generalizable to more than one sense, the first sense is selected as default. In real scenarios, sense disambiguation tools and domain knowledge may be used to pick the sense pertaining to each word.

6.2 t -PAT

The ESearch_Prune (**EP**) algorithm, introduced in Section 3.4, performs exhaustive search in the worst case. We measure how the pruning strategy improves the performance comparing as compared to the pure exhaustive search algorithm. Figure 3(a) shows the time complexity of EP, where the x -axis shows different sizes of d varying from 10 to 50, and the y -axis shows the running time in seconds. The curves in the figures correspond to different t values from 1024 to 16384. We observe that the running time increases as the size of d increases because when $|d|$ is large, the search space is also large. This observation is consistent for all t values. When the size of d is fixed, the running time increases as t increases since many generalized texts are checked before the pruning condition becomes effective.

In these figures, we do not show the running time of the pure exhaustive search algorithm because the pure exhaustive search is very inefficient. For $|d| = 10$, it took about 56 seconds to complete, and for $|d| = 20$, it took hours. For any larger $|d|$ values, we were not able to report the running time within a reasonable amount of time. From Figure 3(a), we can confirm that pruning strategy is effective.

6.3 Uniform t -PAT

The time complexity of Uniform_ESearch_Prune (**UEP**) (Figure 3(b)) follows the same trend as EP ($\alpha = 0.5$). This validates the proposed pruning strategy. Figure 3(c) reports the time complexity of LUB_Search (**LUBS**) with the MStep_Greedy_Search (**MSGS**) strategy when $t = 4096$ (other values of t provide identical observation). As expected that when δ increases, more time is required to execute LUBS. When $\delta = 3$, LUBS is still more efficient than UEP. However, since MSGS does not utilize any pruning strategy, we do not advocate the use of large δ values. When LUBS uses One_Step_Alternative_Search (**OSAS**), the algorithm is as efficient as the case when δ is equal to 0 or 1. We also measured the time complexity of LUBS with varying t values. Since the height of each hypernym tree is constant, t does not affect the running time of LUBS.

Experiments are conducted to analyze the quality of the proposed heuristics. We first generated optimal solutions using UEP. Then we executed LUBS with the two greedy strategies. For MSGS, we run with three δ values from 1 to

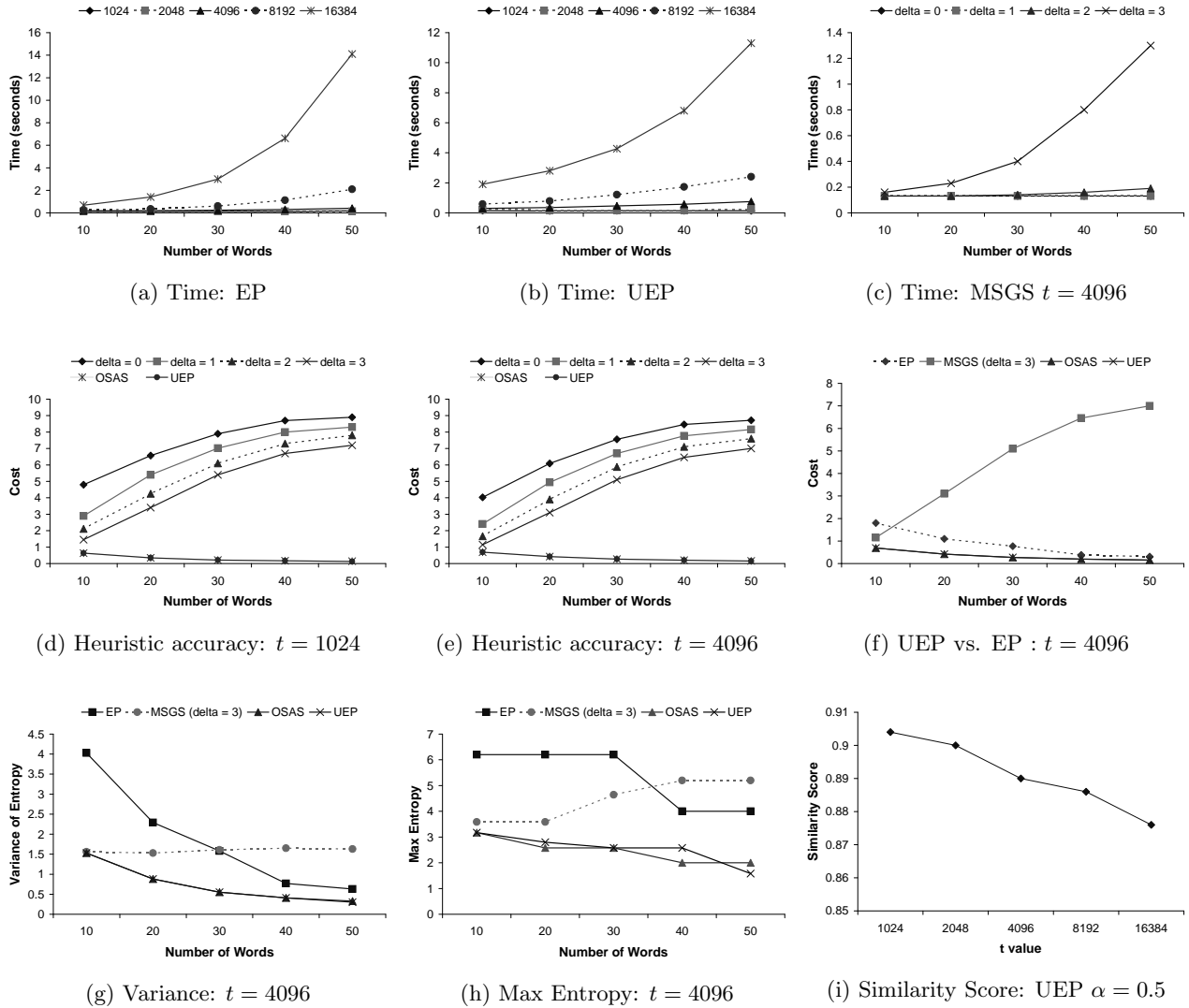


Figure 3: Empirical Results

3. The same experiments are run with three t values. Figures 3(d) and 3(e) show the result. All three figures present similar results: The bottom curves indicate the optimal cost (the lowest cost). For the MSGS heuristic, the bigger the δ , the better (or lower) the cost is. This is consistent with our theoretic reasoning. When δ is large, more possible deviations of \bar{d} will be searched, so it is more likely to find a better result. The OSAS heuristic performs really well, and its result is almost as good as the optimal solution. This matches our intuition that optimal solution is spatially close around the upper bound \bar{d} generated at steps 1-3 of Algorithm 3. As mentioned in Section 4.3, even if MSGS checks more possible deviations of \bar{d} than OSAS (when $\delta = 3$), it does not examine spatially close deviations of \bar{d} as many as OSAS does. Since we assume words in d are independent, the OSAS strategy performs better than MSGS. When words in d are correlated, we expect that MSGS will give better results. We will verify this in the future.

6.4 Uniform t -PAT vs. t -PAT

We have shown that the solution to the t -PAT problem does not protect individual sensitive word equally. Here we validate our claims through empirical results. With the same dataset, first we generate (using the EP algorithm) the optimal solution of t -PAT (Definition 3). We then compare this optimal solution with the solution produced from UEP and other heuristics. Figure 3(f) shows the result regarding the cost function $\mathcal{C}(\bar{d}, t)$ (Equation 1). It can be observed that MSGS ($\delta = 3$) performs worse than EP. The main reason is EP always minimize the first term of the cost function, and the solution generated from MSGS is not very close to the optimal. The OSAS greedy strategy outperforms EP because OSAS produces almost optimal solutions.

Figure 3(g) shows the variance of individual word entropy. The smaller the variance is, the better the uniform plausibility is achieved. For a smaller text, MSGS achieves better uniform plausibility. When $|d|$ is large, more spatially close deviations of d need to be searched for a better solution, but

MSGs fails to do so since δ is fixed to 3. The OSAS strategy achieves almost optimal uniform plausibility. The same conclusion can be drawn from Figure 3(h) which shows the maximum entropy of individual words.

We conducted our experiments with different t and α values, and only show some results with two different t values because the observations do not change with other t values. Regarding other α values (0.25 and 0.75), the variance analysis remains the same. The MSGS strategy is most affected by δ , and OSAS is extremely close to the optimal, the α value can only affect its behavior very little, which depends on the structure of the hypernym trees as well. Overall, the OSAS strategy works very well. Even though the MSGS strategy provides moderate improvement over the upper bound than EP, we expect MSGS will perform better on correlated dataset, and we will verify this in the future.

6.5 Utility of Sanitized Texts

One way to measure the utility of the sanitized texts is to adopt measures from information retrieval literature. Here, we use Cosine Similarity as a measure of utility since it is commonly used. Utility is defined as the similarity between the original document d and the sanitized document. Let \bar{d} be a sanitized document produced from the proposed techniques and d^* be a sanitized document produced from the existing suppression-based techniques (i.e., sensitive words are removed). For illustration purposes, let $d = \{\text{"Diagnose"}, \text{"TB"}\}$ where "TB" is the sensitive word. Assume both "TB" and "Bird_flu" can be generalized to "Infectious_disease". We have $\bar{d} = \{\text{"Diagnose"}, \text{"Infectious_disease"}\}$ (alternatively, $\bar{d} = \{\text{"Diagnose"}, \{\text{"TB"}, \text{"Bird_Flu"}\}\}$) and $d^* = \{\text{"Diagnose"}, *\}$. The normalized frequency vectors are: $fd = \langle \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \rangle$, $fd^* = \langle \frac{1}{\sqrt{2}}, 0 \rangle$ and $f\bar{d} = \langle \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}} \rangle$. Note that the frequency of non-sensitive word does not change, the frequency of the second word in d^* is 0 since it was suppressed, and the frequency of "Infectious_disease" in \bar{d} is divided between "TB" and "Bird_Flu". The cosine similarity is the dot product of these normalized frequency vectors. It is clear that the similarity between d and \bar{d} is greater than that between d and d^* . This result can be generalized to any document, and the proposed approach produces better results than suppression-based techniques.

Experiments are also conducted to measure the change in cosine similarity score between the original text and the sanitized text as t changes. When the value of t increases, the similarity score decreases. Figure 3(i) shows the cosine similarity between the original text and the sanitized text for varying t values. The UEP algorithm was used with an α value of 0.50. Utility measure is not limited to cosine similarity. In the future, we will analyze other forms of utility based on text mining techniques, such as similar document detection and text clustering.

7. CONCLUSION: ABOUT SEMANTICS?

While we have given an information-theoretic measure of privacy and cost of anonymization, what does this do to the text in practical terms? Fully evaluating this would require analyzing this with real readers; such a human subjects study is well beyond the scope of this paper. However, we here present an example, "Uses marijuana for phantom limb pain", to demonstrate both the privacy- and semantics- preserving qualities of our approach. This example was chosen

before we had developed the measures an algorithms, as an example of text that is clearly sensitive (use of an illegal drug), highly individually identifiable (phantom limb pain only occurs in amputees), and contains none of the quasi-identifiers listed in the HIPAA safe harbor rules. Defining **marijuana** and **phantom limb pain** as sensitive, and with $\alpha = 0.5$ and $t = 10$, the sentence sanitizes (using all approaches) to "uses soft drug for pain." This eliminates both sensitivity and identifiability, while preserving readability and much of the semantics.

While further evaluation and development is necessary, we believe that t -PAT provides a valuable supplement to more traditional text sanitization methods, reducing both sensitivity and identifiability of items that remain even after traditional (quasi-)identifiers have been removed.

8. REFERENCES

- [1] M. J. Atallah, C. J. McDonough, V. Raskin, and S. Nirenburg. Natural language processing for information assurance and security: an overview and implementations. In *NSPW '00: Proceedings of the 2000 workshop on New security paradigms*, pages 51–65, New York, NY, USA, 2000. ACM.
- [2] E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.
- [3] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. K. Mohania. Efficient techniques for document sanitization. In *Proceeding of the 17th ACM Conference on Information and Knowledge Mining*, pages 843–852, New York, NY, USA, 2008. ACM.
- [4] M. Douglass, G. Clifford, A. Reisner, W. Long, G. Moody, and R. Mark. De-identification algorithm for free-text nursing notes, 2005.
- [5] M. R. Garey and W. H. F. D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness. 1979.
- [6] The Health Insurance Portability and Accountability Act of 1996. Technical Report Federal Register 65 FR 82462, Department of Health and Human Services, Office of the Secretary, Dec. 2000.
- [7] N. Ide and J. Veronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24:1:1–40, 1998.
- [8] W. Jiang and C. Clifton. Privacy-preserving distributed k-anonymity. In *The 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pages 166–177, Storrs, Connecticut, Aug. 7-10 2005.
- [9] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l -diversity: Privacy beyond k -anonymity. In *Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE 2006)*, Atlanta Georgia, Apr. 2006.
- [10] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
- [11] P. Ruch, R. Baud, A. Rassinoux, P. Bouillon, and G. Robert. Medical document anonymization with a semantic lexicon, 2000.
- [12] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge*

- and Data Engineering*, 13(6):1010–1027, 2001.
- [13] Y. Saygin, D. Hakkani-Tur, and G. Tur. *Sanitization and Anonymization of Document Repositories*, chapter Web and Information Security, pages 133–148. Idea Group Inc., Hershey, PA, USA, 2005.
- [14] L. Sweeney. Replacing personally-identifying information in medical records, the scrub system. pages 333–337, 1996.
- [15] L. Sweeney. k -Anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
- [16] A. Tveit, O. Edsberg, T. B. Rst, A. Faxvaag, O. Nytro, M. T. Nordgrd, Torbjornand Ranang, and A. Grimsmo. Anonymization of general practitioner medical records. In *Proceedings of the second HelsIT Conference*, Trondheim, Norway, 2004.