01 Aug 2009

# T-plausibility: Semantic Preserving Text Sanitization

Wei Jiang
*Missouri University of Science and Technology*, wjiang@mst.edu

Mummoorthy Murugesan

Chris Clifton

Luo Si

## Recommended Citation

W. Jiang et al., "T-plausibility: Semantic Preserving Text Sanitization," *Proceedings of the International Conference on Computational Science and Engineering, 2009*, Institute of Electrical and Electronics Engineers (IEEE), Aug 2009.
The definitive version is available at https://doi.org/10.1109/CSE.2009.353

2009 International Conference on Computational Science and Engineering

# t-Plausibility: Semantic Preserving Text Sanitization

Wei Jiang[1], Mummoorthy Murugesan[2], Chris Clifton[3], Luo Si[4]

[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 and very sensitive information (e.g., diseases and treatment). 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, and develop efficient heuristics to sanitize text documents.**

## I. 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) [5], 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. 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.

Instead of simply removing sensitive terms, the terms can be hidden by more general but semantically related terms to protect sensitive information without unnecessarily degrading document utility. 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. 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

978-0-7695-3823-5/09 $26.00 © 2009 IEEE
DOI 10.1109/CSE.2009.353

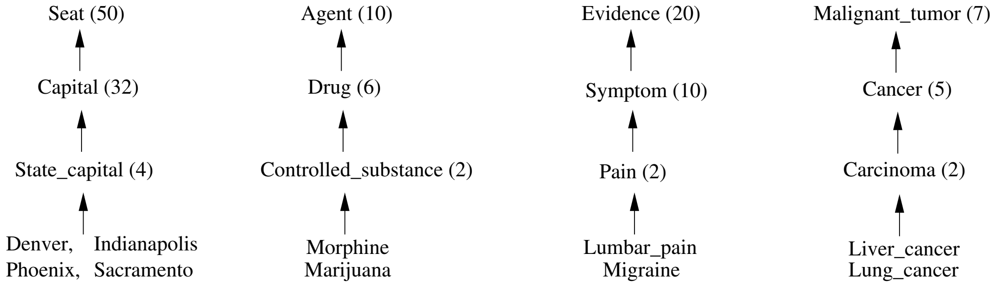


Fig. 2. Word ontologies

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

Fig. 1. A sample text and its sanitized versions

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], [8], [11] 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 II summarizes relevant current work and points out the differences; Section III proposes an information theoretic measure according to the concept of $t$-plausibility and analyzes the hardness of $t$-plausibility based text sanitization problem; Section IV proposes uniform $t$-plausibility measure and an effective and efficient text sanitization heuristic; Section VI validates our analyses via experimental results and Section VII concludes the paper with future research directions.

## II. 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 [9], [12]. 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. A document could be transformed into a tuple in a relational format. Because there are no well defined methods to transform a single text document into a relational database of at least $k$ tuples, applying $k$-anonymization techniques here is not reasonable.

### A. Text Anonymization

Much text anonymization work has mainly concentrated on de-identification of medical documents. The Scrub system [11] finds and replaces patterns of identifying information such as name, location and medical terms with other terms of similar type (e.g., an identified name is replaced with a fake name). Similarly, [13] provides a six-step anonymization scheme to find and replace identifying words with pseudonyms.

In [3], the authors present schemes for removing protected health information (PHI) from free-text nursing notes. They use pattern matching and heuristics to find PHI from nursing notes. Moving away from purely syntactic based recognition of identifying information, a semantic based approach [8] uses MEDTAG, a specialized medical semantic lexicon. Using the semantic tags and manually written disambiguation rules, this system differentiates 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 [10]. The sanitization step uses automatic named entity extraction methods to tag the terms, and then replaces them with dummy values. 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. Pre-defined contextual restrictions guide the sanitization procedure. Assuming the existence of an external database containing demographic information, suppression-based methods were introduced in [2] 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 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 [7] in our examples/experiments to retrieve word ontologies and generate hypernym trees.

## III. $t$-PLAUSIBILITY TEXT SANITIZATION

Before presenting the concept of $t$-plausibility sanitization on text ($t$-PAT), we first introduce key notations and terminologies in Section III-A. The formal definition of $t$-PAT and its hardness is presented in Section III-B.

### A. 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, $d[i]$ $\left(\text{or } \bar{d}[i]\right)$ denote the $i^{th}$ term in $d$ $\left(\text{or } \bar{d}\right)$ (a term is a word, or phrase recognized by the ontology; where we use "word" it could also be such a short phrase) and $|d|$ be the number of terms in $d$. 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 $d =$ [Sacramento, marijuana, lumbar_pain, liver_cancer]. Let $\triangleleft$ denote the generalization operation between documents or terms.

**Definition 1 (Generalizable $\triangleleft$).** We say that $d$ is generalizable to $\bar{d}$ (denoted as $d \triangleleft \bar{d}$) if $|d| = \left|\bar{d}\right|$ 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 $|d| = \left|\bar{d}\right|$. Next, we list additional notations adopted throughout the paper.

- $o = \{o_1, \ldots, o_m\}$: Word ontology set represented as a set of word ontologies related to each word in $d$.
- $d = \{w_1, \ldots, w_m\}$: 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 $d[i]$.
- $\bar{d} = \{\bar{w}_1, \ldots, \bar{w}_m\}$: A generalized text represented as a set of terms, where $\left|\bar{d}\right| = m$, $\bar{w}_i \in o_i$ 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 form WordNet, but for practicality of illustration, we do not use complete ontologies. Based on the previously introduced notations, we define the following functions:

1) $W\left(\bar{w}_i, d, \bar{d}, o\right) \rightarrow \left\{w_i^1, \ldots, w_i^{k_i}\right\}$, the word domain function ($W\left(\bar{w}_i\right)$ for short):
   - Pre-condition: $\bar{w}_i = \bar{d}[i]$ and $d \triangleleft \bar{d}$ according to $o$.
   - Post-condition: $d[i] \in \left\{w_i^1, \ldots, w_i^{k_i}\right\}$, $w_i^j \triangleleft \bar{w}_i$ for $1 \leq j \leq k_i$ and $w_i^1, \ldots, w_i^{k_i}$ are base values in $o_i$.[1]

   For 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) $P_o\left(w_i', \bar{w}_i, d, \bar{d}, o\right) \rightarrow Prob\left(w_i' \triangleleft \bar{w}_i\right)$, the local probability function ($P_o\left(w_i', \bar{w}_i\right)$ for short):
   - Pre-condition: $w_i' \in W\left(\bar{w}_i, d, \bar{d}, o\right)$
   - Post-condition: The probability that given $\bar{w}_i$ (or $d[i]$), $w_i'$ is the original word.

   For 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) $D\left(d, \bar{d}, o\right) \rightarrow \left\{W\left(\bar{w}_1, d, \bar{d}, o\right) \times \cdots \times W\left(\bar{w}_m, d, \bar{d}, o\right)\right\}$, the text domain function ($D(\bar{d})$ for short):
   - Pre-condition: $d \triangleleft \bar{d}$ according to $o$.
   - Post-condition: $\mathcal{D} = \{d_1, d_2, \ldots, d_k\}$ and $d \in \mathcal{D}$, where $k = \prod_{i=1}^{m} k_i$ and $k_i = \left|W\left(\bar{w}_i, d, \bar{d}, o\right)\right|$.

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

   For 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 $\bar{w}_i =$ controlled_substance, then $W(\bar{w}_i)$ returns {Ecstasy, Marijuana}. Assuming uniform distribution in $W(\bar{w}_i)$ and $w_i' =$ marijuana, $P_o(w_i', \bar{w}_i) = \frac{1}{|W(\bar{w}_i)|} = \frac{1}{2}$. Let $d =$ {marijuana, lumbar_pain} and $\bar{d} =$ {controlled_substance, pain}, then $D(d, \bar{d}, o)$ returns $d_1 =$ {marijuana, lumbar_pain}, $d_2 =$ {marijuana, migraine}, $d_3 =$

[1] Base values are the values from $o_i$ at the same level of $w_i$ in $o_i$.

{ecstasy, lumbar_pain} and $d_4 =$ {ecstasy, migraine}. If we assume uniform distribution in both $W(\bar{d}[1])$ and $W(\bar{d}[2])$, for $1 \leq i \leq 4$, $P(d_i, \bar{d}) = \frac{1}{4}$. □

### B. 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 $t$-**P**lausibility **Sa**nitization on **T**ext ($t$-PAT).

**Definition 3 ($t$-PAT).** 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.

In other words, 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 the hardness of $t$-PAT.

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

*Proof:* The reduction is from the subset product problem (SPP), 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 [4]. 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 $M_1, \ldots, M_m$ and the plausibility parameter $t$. The solution of $t$-PAT is a set of $m$ numbers $\{n_1, \ldots, n_m\}$ such that $n_i \in M_i$, and $\prod_{i=1}^{m} n_i$ is equal to the least upper bound of $t$.

The input to SPP is the set of integers $I = \{a_1, \ldots, a_m\}$ 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 $M_1, \ldots, M_m$ and $p$. $\mathcal{A}$ returns a set of $m$ numbers $\{n_1, \ldots, n_m\}$. SPP has a solution iff $\prod_{i=1}^{m} n_i = p$. Suppose $\prod_{i=1}^{m} n_i = p$, then an answer to SPP can be obtained by looking at $\{n_1, \ldots, n_m\}$ 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 there exists a subset $I' \subseteq I$ such that their product is $p$. Based on the input transformation, $\mathcal{A}$ returns $I'$ as well. Since the input and output transformations can be performed in polynomial time, we can conclude that $t$-PAT is NP-hard. ■

### C. Exhaustive Search with Pruning Strategy

To solve $t$-PAT, we can simply enumerate all possible solutions and pick 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 $d = \{w_1, \ldots, w_m\}$ with the given ontology $o = \{o_1, \ldots, o_m\}$. 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 ESearch_Prune($\emptyset, 1, \infty, \emptyset$) and the returned values are the solutions to $t$-PAT.

## IV. $t$-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:

**Algorithm 1** 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
if $i < m$ then
    for $j = 0$ to $h_i$ do
        if $|D(\bar{d} \cup \{\bar{w}_i^{+j}\})| > t_c$ then
            return $(t_c, \bar{d}_c)$
        end if
        $(t_c, \bar{d}_c) \leftarrow$ ESearch_Prune($\bar{d} \cup \{\bar{w}_i^{+j}\}, i+1, t_c, \bar{d}_c$)
        if $t_c = t$ then
            return $(t_c, \bar{d}_c)$
        end if
    end for
else
    if $t \leq |D(\bar{d})| < t_c$ then
        return $(|D(\bar{d})|, \bar{d})$
    end if
end if
return $(t_c, \bar{d}_c)$
```

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 better 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. □

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

### *A. Uniform t-Plausibility and Information Theoretic Measure*

Uniform plausibility implies that each sensitive word needs 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.

Let $m$ be the number of words that need to be generalized in $d$, $H$ be an entropy function and $\alpha$ 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:

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

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

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

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 III-A. Similarly, $H(\bar{d})$ can be calculated as follows:

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

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

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

**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., $\bar{d} = \{$state_capital, drug, pain, carcinoma$\}$). Let $d$ be the original text in Figure 1(b) (i.e., $d = \{$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 $\bar{d} = \{$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 $\bar{d}' = \{$state_capital, drug, pain, carcinoma$\}$,

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

**Algorithm 2** LUB_Search($d, t, o, \delta$) - The **L**east **U**pper **B**ound Search for uniform $t$-PAT

```
Require: A base document $d$, a threshold $t$ and a set of hypernym trees $o$
1: for all $w_i \in d$ do
2:     Find a $\bar{w}_i$ and $H(\bar{w}_i) = \left\lceil \frac{\log t}{m} \right\rceil$
3: end for
4: $c \leftarrow \mathcal{C}(\bar{d}, t)$
5: if $c = 0$ then
6:     return $\bar{d}$
7: end if
8: $(c, \bar{d}) \leftarrow$ One_Step_Alternative_Search($\bar{d}, d, t$)
9: return $\bar{d}$
```

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 $\bar{d}$. □

As mentioned before, the optimal solutions presented in Section III-C do not take into account the concept of uniform plausibility. That is, if $\bar{d}$ is optimal according to Definition 3, not all words in $d$ 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).** Give a text $d$, a set of hypernym trees $o$ (related to $d$), $\alpha$ 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.

### B. Proposed Heuristic

While a cleverly pruned exhaustive search can be used to find the optimal solution, it is not practical for real-world use. (We do present such optimal results, denote UEP, as a baseline in the experiments.) Thus, in this section, we propose a heuristic to generate sanitized texts that possess the property of uniform plausibility. Since we need to know $P_o(w_i', \bar{w}_i)$ value for each $w_i'$ in $W(\bar{w}_i)$ to use the cost function $\mathcal{C}$, for the rest of this section, we assume that words are uniformly distributed in each $W(\bar{w}_i)$. 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 2) 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 2 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 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. If the condition does not hold, the algorithm will continue to the greedy search phase. The procedure One_Step_Alternative_Search at step 8 of Algorithm 2 returns a generalized text that is either the same as $\bar{d}$ or a better generalized document according to Equation 1. Main steps of One_Step_Alternative_Search are given in Algorithm 3. 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.

**Algorithm 3** One_Step_Alternative_Search($\bar{d}, d, t$)

```
Require: $\bar{d}$ a generalized text related to $d$ and $t$ is the privacy threshold
$m \leftarrow |\bar{d}|$
$t_c \leftarrow \mathcal{C}(\bar{d}, t)$
for $i = 1$ to $m$ do
    for $j = 1$ to $m$ do
        $\bar{d}' \leftarrow \bar{d}_{-j} \cup \{\bar{w}_i^+\}$
        if $\mathcal{C}(\bar{d}', t) < t_c$ and $H(\bar{d}') > \log t$ then
            $t_c \leftarrow \mathcal{C}(\bar{d}', t)$
            $\bar{d}_c \leftarrow \bar{d}'$
        end if
        $\bar{d}' \leftarrow \bar{d}_{-j} \cup \{\bar{w}_i^-\}$
        if $\mathcal{C}(\bar{d}', t) < t_c$ and $H(\bar{d}') > \log t$ then
            $t_c \leftarrow \mathcal{C}(\bar{d}', t)$
            $\bar{d}_c \leftarrow \bar{d}'$
        end if
    end for
    $\bar{d} = \bar{d}_c$
end for
return $(t_c, \bar{d}_c)$
```

LUB_Search consists of two phases, and 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 2) 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 second phase, One_Step_Alternative_Search has the complexity of O($m^2$) Thus the complexity of LUB_Search is bounded by O($m^2$).

## V. Privacy Protection and Other Practical Issues

The proposed text sanitization model also performs well against other models, such as $k$-Anonymity [9], [12] and ERASE [2]. Due to space limitations, we omit the details here. In short, the $t$-PAT model is more general in protecting personal privacy. More detailed comparative analysis among existing models can be found in our technical report [6].

We may ask whether the proposed approaches can be applied 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], [3], [11], [13]. These techniques are domain specific, and additional documents may be required to train the learning algorithms.

## VI. Empirical Analyses

According to the proposed schemes, there are two aspects of performance that we are most interested in: accuracy of the heuristic 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 search. For the running time evaluation, we measure the gain by pruning-based search and heuristic search scheme over the exhaustive search.

### A. 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 ($\pm 1$) 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.

### B. t-PAT

The ESearch_Prune (**EP**) algorithm, introduced in Section III-C, 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. These experiments returned solutions with the exact $t$ values.

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.

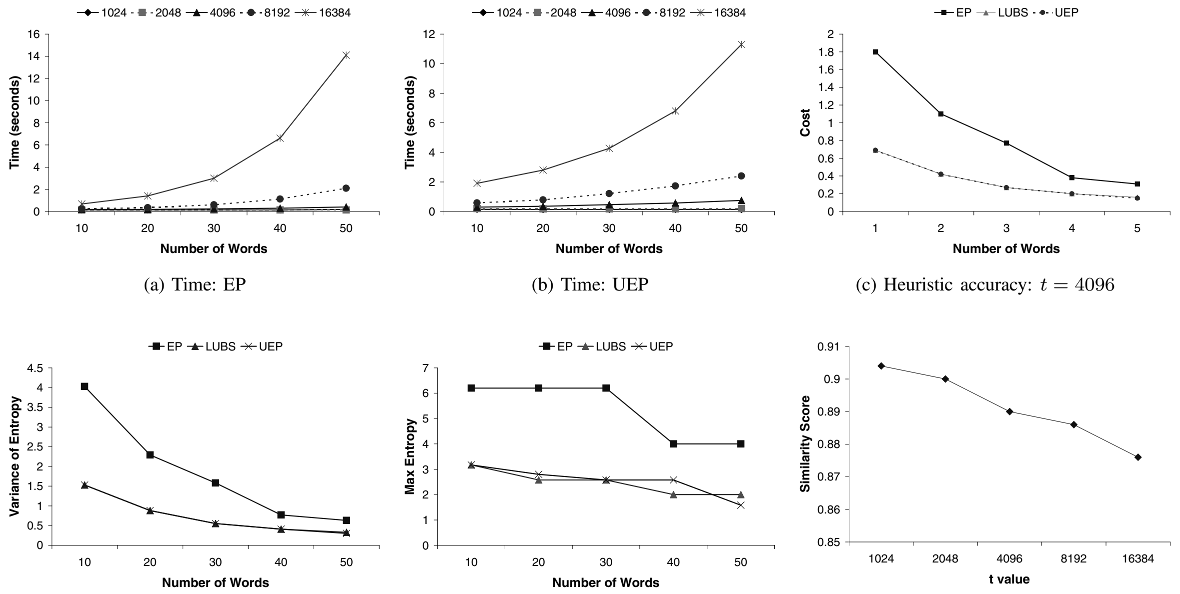
### C. Uniform $t$-PAT

The time complexity of Uniform_ESearch_Prune (**UEP**) (Figure 3(b)) with $\alpha = 0.5$, follows the same trend as EP. This validates the proposed pruning strategy. The running time for LUBS with One_Step_Alternative_Search is much less than the UEP. Also, the solution from LUBS with One_Step_Alternative_Search is same as that of UEP. We first generated optimal solutions using UEP for $t = 4096$. Then we executed LUBS with the same $t$ value. Figure 3(c) shows the result. The LUBS 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 2.

### D. 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. We then compare this optimal solution with the solution produced from UEP and LUBS. Figure 3(c) shows the result regarding the cost function $\mathcal{C}(\bar{d}, t)$ (Equation 1). It can be observed that EP performs worse than LUBS. The main reason is that EP always minimizes only the first term of the cost function in $\mathcal{C}(\bar{d}, t)$. The LUBS greedy strategy outperforms EP because it produces almost optimal solutions with respect to $\mathcal{C}(\bar{d}, t)$. Figure 3(d) shows the variance of individual word entropy. The smaller the variance is, the better the uniform plausibility is achieved. LUBS achieves almost optimal uniform plausibility. The same conclusion can be drawn from Figure 3(e) which shows the maximum entropy of individual words. We only show results with $t = 4096$ because the observations do not vary much with other values. LUBS is extremely close to the optimal, the $\alpha$ value can only affect its behavior very little, which depends on the structure of the hypernym trees as well.

We also measure the utility of the sanitized texts based on Cosine Similarity since it is commonly used in information retrieval literature. Utility is defined as the similarity between the original document $d$ and the sanitized document $\bar{d}$. Let $d^*$ be a sanitized document produced from the existing suppression-based techniques (i.e., sensitive words are removed). For illustration purposes, let $d = \{$"Diagnose", "TB"$\}$ where "TB" is the sensitive word. Assume both "TB" and "Bird_flu" can be generalized to "Infectious_disease". We have $\bar{d} = \{$"Diagnose", "Infectious_disease"$\}$ and $d^* = \{$"Diagnose", *$\}$. The normalized frequency vectors are: $fd = \langle \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \rangle$, $f\bar{d} = \langle \frac{1}{\sqrt{2}}, \frac{1}{2\sqrt{2}} \rangle$ and $fd^* = \langle \frac{1}{\sqrt{2}}, 0 \rangle$. 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. Figure 3(f) shows the similarity score between $d$ and $d^*$, where the UEP algorithm was used for anonymization with an $\alpha$ value of 0.50.



(a) Time: EP

(b) Time: UEP

(c) Heuristic accuracy: $t = 4096$

(d) Variance: $t = 4096$

(e) Max Entropy: $t = 4096$

(f) Similarity Score: UEP $\alpha = 0.5$

Fig. 3. Empirical Results

## VII. 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 and 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.

## Acknowledgment

The authors wish to thank anonymous reviewers for their valuable suggestions. Support for this work was provided by MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research and by the National Science Foundation under Grant No. 0428168.

## 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] 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 (CIKM)*, pages 843–852, New York, NY, USA, 2008. ACM.

[3] M. Douglass, G. Cliffford, A. Reisner, W. Long, G. Moody, and R. Mark. De-identification algorithm for free-text nursing notes, 2005.

[4] M. R. Garey and W. H. F. D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness. 1979.

[5] 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.

[6] W. Jiang, M. Murugesan, C. Clifton, and L. Si. Semantic preserving text sanitization. Technical Report CSD TR# 06-015, Department of Computer Science, Purdue University, West Lafayette, IN, Aug. 2008.

[7] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[8] P. Ruch, R. Baud, A. Rassinoux, P. Bouillon, and G. Robert. Medical document anonymization with a semantic lexicon, 2000.

[9] P. Samarati. Protecting respondents’ identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6):1010–1027, 2001.

[10] 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.

[11] L. Sweeney. Replacing personally-identifying information in medical records, the scrub system. pages 333–337, 1996.

[12] L. Sweeney. $k$-Anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[13] 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.