Processing text for privacy: An information flow perspective

Natasha Fernandes, Mark Dras, and Annabelle McIver *

Dept. Computing, Macquarie University, Australia

Abstract. The problem of text document obfuscation is to provide an automated mechanism which is able to make accessible the content of a text document without revealing the identity of its writer. This is more challenging than it seems, because an adversary equipped with powerful machine learning mechanisms is able to identify authorship (with good accuracy) where, for example, the name of the author has been redacted. Current obfuscation methods are ad hoc and have been shown to provide weak protection against such adversaries. Differential privacy, which is able to provide strong guarantees of privacy in some domains, has been thought not to be applicable to text processing.
In this paper we will review obfuscation as a quantitative information flow problem and explain how *generalised differential privacy* can be applied to this problem to provide strong anonymisation guarantees in a standard model for text processing.

Keywords: Refinement; information flow; privacy; probabilistic semantics; text processing; author anonymity; author obfuscation.

1 Introduction

Up until the middle of the nineteenth century it was common for British authors to publish their work anonymously. There were many reasons for the practice, and surprisingly many well-known authors practised it including (as we now know) Alexander Pope, Jonathan Swift, Jane Austen and Daniel Defoe. But can a work ever be entirely anonymous, in the sense that it is not possible to identify the author with full certainty? Authors typically develop their own personal style, and the famous example of the "Federalist papers" showed that the analysis of word frequencies can be used to build compelling evidence to support the identification of authors of anonymous works [22].

Koppel et al. [14] trace the development of techniques from that of Mosteller and Wallace [22] (and earlier) to more recent machine learning methods, which have taken advantage of the observation that many aspects of style — not only word counts — in writing can be captured by statistical methods. For the last decade, stylometric machine learners have been able to identify authors with

* We acknowledge the support of the Australian Research Council Grant DP140101119.

accuracy better than 90% from a set of 50 candidates, and have been successfully applied to identification tasks on sets of (anonymous) documents written by tens of thousands of authors.

Methods related to these were employed by researchers working on the 2006 Netflix release of a "deidentified" database of movie reviews in order to allow researchers to work on improving its recommendation systems. Unfortunately deidentifying data (i.e. removing names) is very different from properly anonymising it and, in this case, privacy researchers were able to demonstrate publicly that Netflix's data contained more information than intended leading to a lawsuit.

There remain many legitimate reasons why an author might want to disguise his or her identity. Indeed could Netflix have done a better job to protect its contributors whilst still preserving the information contained in the reviews well enough to be useful to researchers working on improving Netflix's recommendation systems? In response to the Netflix lawsuit, and other such breaches of privacy, "PAN" a series of scientific events and shared tasks on digital text forensics[1] proposed a task to encourage research into creating systems which are able to truly anonymise. The statement of the task is:

Given a document, paraphrase it so that its writing style does not match that of its original author, anymore.

As an example, consider this extract from George Orwell's *Nineteen Eighty-Four*:

> "The object of persecution is persecution. The object of torture is torture. The object of power is power."

It's clear that Orwell's intent was to evoke a sense of shock by the overwhelming use of strong repetition. Another way of saying the same thing might be:

> "The aim of persecution, abuse and power is respectively to mistreat, to torture and to control."

which, stripped of of its powerful stylistic ruse, has been rendered into a rather dull opinion.

The range of approaches to "obfuscating" text documents automatically that have been attempted up to and including the PAN task have had limited success. Many of those approaches were inspired by *k-confusability* which articulates the idea of "confusing" some secret with k other things, but turns out to be susceptible to the well-known "linkage" and "intersection" attacks.

Methods based on differential privacy (DP) [10] — which provide some protection from these attacks — have not been attempted to date for this problem. There has been interest for some time in combining DP with machine learning in general (for example, [7]), including recent "deep learning" approaches [1], although applications to text are challenging because of its discrete, complex

[1] http://pan.webis.de/index.html

and unstructured nature. Moreover, a key difference with our application of interest is that we want to conceal the authorship of an individual released document; the goal for DP with machine learning is typically to preserve the privacy of members of the training dataset.

In this paper we link the original goals of the PAN obfuscation task to two theoretical areas in computer science, with the aim of providing a solid foundation for the enterprise and to enable new techniques in theoretical privacy to be applied to this problem.

We explain how this task can be viewed as a problem of Quantitative Information Flow where we describe the result of an obfuscation process as a "channel". In this way we can show upper bounds on the ability of any adversary to determine the real author (whether or not the adversary is using machine learning).

Second, we describe how the novel metrics used in machine learning algorithms for author identification can in fact be used after all to define obfuscators based on differential privacy. The trick is to used *Generalised Differential Privacy* [6] originally used in location-based privacy and which can be used for unstructured data.

2 Text document processing

Text documents are processed in many different ways depending on the objective. For example a document might need to be classified in terms of its topic which can be helpful for cataloging in document repositories; or documents can be paraphrased so that domain professionals are able to determine which documents are relevant for their research or report compilation. Statistical and machine learning approaches are the standard way now to tackle these tasks [18]; most recently, approaches falling within the "deep learning" paradigm, using neural networks with many layers, have become dominant and produced state-of-the-art results for many tasks [26].

All these approaches use very different algorithms and representations of documents, but the basic idea is the same, even when the representations and implementations differ: thousands of document samples are analysed to identify important "features" depending on the specific goal of the task. This constitutes the "learning phase" and the result is a "best possible" correlation between categories and the discovered features. Learning algorithms (for classification problems) are evaluated by subjecting the learned correlation to the identification to datasets which are not part of the learning set, and typically counts of correct identification or classification are used to rate the success of the method.

For us the aim is to determine how to obfuscate automatically according to the following constraints:

> *The result of an obfuscated document must retain as much of the original content in such a way that the author of the obfuscated document cannot be identified.*

As a simplification, we focus on the identification of author, and (separately) topic classification (rather than full "content") both of which are examples of "classification problems" in machine learning.

2.1 Representing documents for topic classification and author identification

In machine learning documents are transformed into representations that have been found to enable the discovery of features which perform well on a particular classification task. A very simple representation is to choose the word components of a document, so for example,

"The object of persecution is persecution" can be represented by the set:

$$\{\text{“}\mathit{The}\text{”}, \text{“}\mathit{object}\text{”}, \text{“}\mathit{of}\text{”}, \text{“}\mathit{persecution}\text{”}, \text{“}\mathit{is}\text{”}\} \ . \tag{1}$$

This, of course has lost some useful details such as the number of times that words appear; an alternative richer representation is a "bag of words" which, in this case, retains the repetition of "*persecution*":

$$\{\!|\text{“}\mathit{The}\text{”}, \text{“}\mathit{object}\text{”}, \text{“}\mathit{of}\text{”}, \text{“}\mathit{persecution}\text{”}, \text{“}\mathit{persecution}\text{”}, \text{“}\mathit{is}\text{”}|\!\} \ . \tag{2}$$

Even though it still loses a lot information from the original sentence, such as word order, it turns out that the the bag of words representation is still very useful in topic classification, where correlations between topics and the types and frequency of words can be used to assign a topic classification to a document. It can also be used in some stylometric analysis where authors can be correlated with the number of times they use a particular word — in the identification of the authors of the Federalist papers, it was discovered that Hamilton only used "while" whereas Madison preferred "whilst", and used "commonly" much more frequently than did Hamilton.

More recent, widely used author identifiers use "character n-gram" representation for documents. The *n-gram* representation transforms a document into a list of each subsequence of characters of length n, including (sometimes) spaces and punctuation. Such a character 3-gram representation of our example is:

$$\langle \text{“}\mathit{The}\text{”}, \text{“}\mathit{he\ }\text{”}, \text{“}\mathit{e\ o}\text{”}, \text{“}\mathit{\ ob}\text{”}, \text{“}\mathit{obj}\text{”}, \text{“}\mathit{bje}\text{”}, \text{“}\mathit{jec}\text{”}, \text{“}\mathit{ect}\text{”}, \text{“}\mathit{ct\ }\text{”} \ldots \rangle \ . \tag{3}$$

This representation seems to capture things like systematic punctuation and common word stems, all of which can characterise an author. A particular character n-gram-based method of interest is the one developed by Koppel et al. [15]. This method uses character 4-grams (but without spaces) to classify authorship on a document set consisting of blog posts from thousands of authors, and achieve in excess of 90% precision with 42% coverage for a 1000-author dataset. On account of its strong performance and suitability for large author sets, and the fact that it underpins the winning systems of PAN shared tasks on author identification [24, 13], this algorithm is one of the standard inference

attackers used in the PAN obfuscation task. This is therefore the authorship identification algorithm we use in the rest of this paper.

2.2 Privacy versus utility

Obfuscating a document means changing the words somehow, and with the use of machine learning as an adversary (as in author identification) or as a friend (as in topic identification) we can see that the bag of words (2) or n-gram representation (3) will be affected.

What we would like is to be able to show that for any adversary whether or not they are using the n-gram representation that the obfuscation method reduces their ability to identify authors, whereas using a state-of-the-art method based on a bag of words representation the topic identification remains almost as it was before obfuscation.

To deal with the former, we shall follow Alvim et al. [2] to model a privacy mechanism as an information flow channel; for the latter we will use generalised differential privacy to show how to preserve topicality using an appropriate metric for "meaning".

3 Channels, secrets and information flow

A *privacy mechanism* produces observations determined by secret inputs; the elements of the channel model for information flow are inputs of type $\mathcal{X}$, observations of type $\mathcal{Y}$ and a description of how the inputs and observations are correlated. For any set $\mathcal{S}$ we write $\mathbb{D}\mathcal{S}$ for the set of discrete distributions on $\mathcal{S}$.

A *channel* between $\mathcal{X}$ and $\mathcal{Y}$ is a (stochastic) matrix whose $\mathcal{X}$-indexed rows sum to 1. We write the type of such channels/matrices as $\mathcal{X}\!\!\rightarrow\!\!\triangleright\mathcal{Y}$ and for $C\colon\mathcal{X}\!\!\rightarrow\!\!\triangleright\mathcal{Y}$ its constituents are elements $C_{x,y}$ at row x and column y that gives the conditional probability of output y from input x, the x'th row $C_{x,-}$ and the y'th column $C_{-,y}$. Any row $C_{x,-}$ of $C\colon\mathcal{X}\!\!\rightarrow\!\!\triangleright\mathcal{Y}$ can be interpreted as an element of $\mathbb{D}\mathcal{Y}$.

A *secret* is a distribution in $\mathbb{D}\mathcal{X}$; initially we call such secrets priors, by which we mean that the adversary might have some prior knowledge which means that knows some secret values are more likely than others, however the fact that his knowledge is represented as a distribution means that he does not know for sure. The mechanism modelled by a channel C produces a correlation between the inputs and the observables.

Given a channel $C\colon\mathcal{X}\!\!\rightarrow\!\!\triangleright\mathcal{Y}$ and prior $\pi\colon\mathcal{X}$ the joint distribution $J\colon\mathbb{D}(\mathcal{X}\times\mathcal{Y})$ is given by $J_{x,y}:=\pi_x C_{x,y}$. For each y the column $J_{-,y}$, the adversary can update his knowledge a-posteriori using Bayesian reasoning that revises the prior: i.e. normalising $J_{-,y}$ [2] to give the posterior induced on π by that y. We write $\pi\rangle C$ for the joint distribution J, and $\overrightarrow{J}:\mathbb{D}\mathcal{Y}$ for the (right) marginal probability defined

[2] If several distinct y's produce the same posterior, they are amalgamated; if there is y with zero marginal probability, it and its (undefined) posterior are omitted.

$\overrightarrow{J}_y := \sum_{x:\mathcal{X}} J_{x,y}$. For each observation y we denote the corresponding posterior $J^y := \overrightarrow{J} \,/\, \overrightarrow{J}_y$.

There are two operations on channels which we will use to model two attacks on privacy.

Definition 1. *Let $C: \mathcal{X} \to \mathcal{Y}_1$ and $D: \mathcal{X} \to \mathcal{Y}_2$ be channels. We define the* sequential composition *$C;D: \mathcal{X} \to (\mathcal{Y}_1 \times \mathcal{Y}_2)$ as follows:*

$$(C;D)_{x,(y_1,y_2)} \quad := \quad C_{x,y_1} \times D_{x,y_2} \ .$$

Sequential composition allows the adversary to amalgamate his knowledge about the secret which is leaked from bth C and D.

The second operator models the situation where a channel leaks information about a secret from $\mathcal{X}$ which has an interesting correlation wth a second secret $\mathcal{Z}$. The adversary can then use channel $C: \mathcal{X} \to \mathcal{Y}$ to deduce some information about the second secret!

Definition 2. *Given channel $C: \mathcal{X} \to \mathcal{Y}$ and joint distribution $Z: \mathbb{D}(\mathcal{Z} \times \mathcal{X})$ expressing an interesting correlation between two secret types $\mathcal{Z}$ and $\mathcal{X}$, we define the* Dalenius composition *$Z \cdot C: \mathcal{Z} \to \mathcal{Y}$ defined by "matrix multiplication":*

$$(Z \cdot C)_{z,y} \quad := \quad \sum_{x:\mathcal{X}} Z_{z,x} \times C_{x,y} \ .$$

Dalenius composition[3] can be used the model the risk posed by mechanisms that inadvertently release information about a second secret that is known to be correlated with secrets associated with the mechanism.

3.1 Vulnerability induced by gain-functions

When a channel publishes its observables, the most important concern is to determine whether an adversary can do anything damaging with the information released. We can investigate an adversary's ability to use the information effectively using the idea of "vulnerability" [4], a generalisation of entropy, no longer necessarily e.g. Shannon, and whose great variety allows fine-grained control of the significance of the information that might be leaked [4, 5].

Given a secret-space $\mathcal{X}$, vulnerability is induced by a *gain function* over that space, typically g of type $\mathbb{G}_{\mathcal{W}}\mathcal{X} = \mathcal{W} \to \mathcal{X} \to \mathbb{R}$, for some space of *actions* $w: \mathcal{W}$. When $\mathcal{W}$ is obvious from context, or unimportant, we will omit it and write just $g: \mathbb{G}\mathcal{X}$. Given g and w (but not yet x) the function $g.w$ is of type $\mathcal{X} \to \mathbb{R}$ [4] and

[3] Named after Tore Dalenius who pointed out this risk in statistical databases [9]

[4] We write dot for function application, left associative, so that function g applied to argument w is $g.w$ and then $g.w.x$ is $(g.w)$ applied to x, that is using the Currying technique of functional programming. This convention reduces clutter of parentheses, as we see later.

can thus be regarded as a random variable on $\mathcal{X}$. As such, it has an expected value on any distribution π over $\mathcal{X}$, written $\mathcal{E}_\pi\, g.w := \sum_{x:\,\mathcal{X}} g.w.x \times \pi_x$.[5]

Once we have x, the (scalar) value $g.w.x$ is simply of type $\mathbb{R}$ and represents the gain to an adversary if he chooses action w when the secret's actual value is x. A particularly simple example is where the adversary tries to guess the exact value of the secret. His set of actions is therefore equal to $\mathcal{X}$, with each action a guess of a value; we encode this scenario with gain function bv defined

$$bv.w.x = (1 \text{ if } w{=}x \text{ else } 0)\ , \tag{4}$$

so that the adversary gains 1 if he guesses correctly and 0 otherwise. A special case of this is when an attacker tries to guess a property of the secret (rather than the whole secret). For example let $\sim$ be an equivalence class over secrets, and suppose that the attacker tries to guess the equivalence class. The guesses $\mathcal{W}$ now correspond to equivalence classes, and:

$$bv_\sim.w.x = (1 \text{ if } x \in w \text{ else } 0)\ . \tag{5}$$

A gain function $g{:}\,\mathbb{G}\mathcal{X}$ induces a *g-vulnerability* function $V_g{:}\,\mathbb{D}\mathcal{X}{\to}\mathbb{R}$ so that $V_g[\pi]$ for $\pi{:}\,\mathbb{D}\mathcal{X}$ is the maximum over all choices $w{:}\,\mathcal{W}$ of the expected value of $g.w$ on π, that is $\max_w(\mathcal{E}_\pi\, g.w)$. In the simple 1-or-0 case above, the vulnerability V_{bv} is called the *Bayes vulnerability*; it is one-minus the Bayes-Risk of Decision Theory, and it gives the maximum probability of an adversary guessing the secret if his prior knowledge about it is π.

We can now use g-vulnerability to determine whether the information leaked through a channel is helpful to the adversary.

Definition 3. *Given a prior $\pi \in \mathbb{D}\mathcal{X}$, a channel $C{:}\,\mathcal{X} \twoheadrightarrow \mathcal{Y}$ and gain function $g{:}\,\mathbb{G}\mathcal{X}$, we define the* average posterior vulnerability *as*

$$V_g[\pi\rangle C] \quad := \quad \sum_{y:\,\mathcal{Y}} \overrightarrow{J}_y \times V_g[J^y]\ ,$$

where $J := (\pi\rangle C)$.

For each observation, the posterior J^y is the adversary's revised view of the value of the secret; the posterior is actually more vulnerable because the adversary can choose to execute a different action (compared to his choice relative to the prior) to optimise the vulnerability $V_g[J^y]$. The posterior vulnerability $V_g[\pi\rangle C]$ is then his average increase in gain. Comparing $V_g[\pi\rangle C]$ and $V_g[\pi]$ then gives an idea of how much information the adversary can usefully use relative to the scenario determined by g.

In this paper we shall use the *multiplicative g-leakage*, defined by

$$\mathcal{L}_g(C) \quad := \quad V_g[\pi\rangle C]/V_g[\pi]\ , \tag{6}$$

[5] In general we write $\mathcal{E}_\pi\, f$ for the expected value of function $f{:}\,\mathcal{X}{\to}\mathbb{R}$ on distribution $\pi{:}\,\mathbb{D}X$.

which gives the relative increase in gain. Moreover the leakage measure exhibits an important robust approximation which will be relevant for privacy mechanisms in text processing.

Theorem 1. *[4] Let $C: \mathcal{X} \nrightarrow \mathcal{Y}$ be a channel, and let $u: \mathbb{D}\mathcal{X}$ be the uniform prior over $\mathcal{X}$. Then for all priors π and non-negative gain functions g we have that:*

$$V_g[\pi \rangle C]/V_g[\pi] \quad \leq \quad V_{g_{bv}}[u \rangle C]/V_{g_{bv}}[u] \ .$$

A final theoretical idea which will be useful for our application to privacy is that of *security refinement*. If $C \sqsubseteq D$ (defined below) then D is more secure than C in any scenario, because D's posterior vulnerability relative to any gain function is always less than C's and therefore the information D releases is less useable than the information released by C.

Definition 4. *Let $C: \mathcal{X} \nrightarrow \mathcal{Y}^1$, and $D: \mathcal{X} \nrightarrow \mathcal{Y}^2$ be channels. We say that $C \sqsubseteq D$ if*

$$V_g[\pi \rangle C] \quad \geq \quad V_g[\pi \rangle D] \ ,$$

for all gain functions g and priors π.

We can use security refinement to express compositionality properties.

Theorem 2. *[19, 3] Let C, D, E be channels and $Z: \mathbb{D}(\mathcal{Z} \times \mathcal{Y})$ be a correlation between secret types $\mathcal{Z}$ and $\mathcal{X}$. The following inequalities hold.*

1. $C \sqsubseteq D \quad \Rightarrow \quad C; E \sqsubseteq D; E$
2. $C \sqsubseteq D \quad \Rightarrow \quad Z \cdot C \sqsubseteq Z \cdot D$

3.2 Privacy mechanisms as channels

A privacy mechanism is normally modelled as a function $\mathcal{K}$ which, given a value x from a secret set $\mathcal{X}$, outputs some observable value $y: \mathcal{Y}$. The exact output could be determined by a probability distribution which, in an extreme instance such as redaction, could be a point distribution without any randomness applied.

Traditional approaches to privacy are founded on a principle we call "confusablity". Roughly speaking a mechanism imbues privacy by ensuring that the real value of the secret could be confused amongst several other values. In this section we examine confusability in terms of information flow to show how simple confusability mechanisms provide weak privacy.

3.3 Attacks on simple confusability

Traditional approaches to privacy in text programming use the idea of *k-anonymity* [25], which is related to confusability.

Definition 5. *A channel $C \in \mathcal{X} \nrightarrow \mathcal{Y}$ is k-confusable if for each column y (observable), the entries $C_{x,y}$ are non-zero for at least k distinct values of x.*

Although k-confusable seems like a nice, straightforward property, it has some problems when combined with prior knowledge, and k-confusable mechanisms are susceptible to *intersection* and *linkage attacks*.

Intersection attacks A mechanism that is k-confusable separates the values of the secret into two subsets (for each observation): one for secret values that are still possible, and one for values which are not possible.

An *intersection* attack refers to the scenario where two different mechanisms are used, one after another. An adversary is able to combine the information flow from both mechanisms to deduce more about the value of the secret than he can from either mechanism separately. For example define two channels as follows. Let $\mathcal{X}{:=}\{x_0, x_1, x_2, x_3\}$ and $\mathcal{Y} = \{y_0, y_1\}$.

$$C_{x_i,y_j} := (i = j \bmod 2) \tag{7}$$

$$D_{x_i,y_j} := 1 \textit{ iff } (j = 0 \wedge i < 2) \vee (j = 1 \wedge i \geq 2)\ . \tag{8}$$

Both C and D are 2-confusable since C divides the secret into two equivalence classes: $\{x_0, x_2\}$ and $\{x_1, x_3\}$, whereas D divides it into $\{x_0, x_1\}$ and $\{x_2, x_3\}$. Thus if only C or D is used then indeed the secret is somewhat private, but if both are used one after the other then the secret is revealed entirely, since the adversary can identify the secret by locating it simultaneously in an equivalence class of C and of D.

We can, model such a scenario by the sequential composition of the two mechanisms separately, i.e. the mechanism of an intersection attack is modelled by $C; D$. The susceptibility of k-confusable mechanisms to intersection attacks is summed up by a failure of compositionality for k-confusability.

Lemma 1. *k-confusability is not preserved by to sequential composition.*

Proof. We use the counterexample described above: C and D defined respectively at (7) and (8) are 2-confusable but $C; D$ is not 2-confusable.

Lem. 1 implies that mechanisms based on k-confusability are vulnerable to intersection attacks, a flaw that has been pointed out elsewhere [11].

Linkage attacks A *linkage attack* can be applied when the adversary has some prior knowledge about how some secret $\mathcal{Z}$ is correlated to another secret $\mathcal{X}$. When information leaks about $\mathcal{X}$ through a channel $C{:}\,\mathcal{X} \nrightarrow \mathcal{Y}$ the adversary is able to deduce something about $\mathcal{Z}$. A simple example of this occurs when for example secret z has value z_0 exactly when x has value x_1 or x_2, and z has value z_1 otherwise. In this example z and x are *linked* through the correlation defined

$$Z_{z_i,x_j} \quad := \quad (i = j \bmod 2)\ . \tag{9}$$

In this case, since the mechanism C defined above at (7) leaks whether x is in $\{x_0, x_2\}$ or $\{x_1, x_3\}$, this information put together with correlation Z leaks the value of z exactly. Even though C is 2-confusable.

Dalenius composition $Z \cdot C$ now models such linkage attacks, combining correlations with information flows to yield a mechanism describing the leaks about a correlated secrets. As for intersection attacks, we see that k-confusability fails compositionality with respect to Dalenius composition.

Lemma 2. *k-confusability is not preserved by Dalenius composition.*

Proof. We observe that C defined above at (7) is 2-confusable but that $Z \cdot C$ is not 2-confusable (for z), where Z is defined at (9).

Lem. 2 implies that privacy that relies on k-confusability is vulnerable to attacks that can use prior knowledge.

3.4 Universal confusability

We can avoid intersection attacks and linkage attacks by strengthening k-confusability to "universal confusability".

Definition 6. *We say that a channel C is* universally confusable *if it is k-confusable for all $k \geq 1$.*

A channel is universally confusable if all its entries $C_{x,y}$ are non-zero. This means that for any posterior reasoning, the channel will maintain any extent of confusability that was already present in the prior. In fact universal confusability is (somewhat) robust against intersection and linkage attacks, because the strong confusability property is compositional with respect to sequential and Delanius composition. Universal confusability is particularly important for text processing because all kinds of unforeseen and unexpected correlations can be learned and used, even if they are too strange to understand.

3.5 Differential privacy

We turn to the question of how to implement mechanisms that are universally confusable; the answer is given by *differential privacy*, which not surprisingly was defined to defend against linkage and intersection attacks.

The definition of an *ϵ-differentially private mechanism* is normally described as a function of type $\mathcal{X} \to \mathbb{D}\mathcal{Y}$, satisfying the following constraint. Let $dist\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a distance function, then for all $x, x' \in \mathcal{X}$ with $dist(x, x') \leq 1$, and properties α, we must have:

$$\mathcal{K}.x(\alpha) / \mathcal{K}.x'(\alpha) \quad \leq \quad e^{\epsilon} \ . \tag{10}$$

In fact, as has been pointed out by Alvim et al. [2] the mechanism $\mathcal{K}$ corresponds to a channel in $C^{\mathcal{K}}\colon \mathcal{X} \nrightarrow \mathcal{Y}$ where the rows are defined by $C^{\mathcal{K}}_{x,y} := \mathcal{K}.x(Y = y)$. From (10) it is clear that $C^{\mathcal{K}}$ is strongly confusable because if any non-zero entry was present, the multiplicative constraint would fail to hold.

Moreover we can also obtain an upper bound for the scenario of an attacker trying to use the information leaked to guess the secret, in the sense that the following leakage bound holds [4]. For any prior π,

$$\begin{array}{ll} & \textit{The probability of correctly guessing the secret after applying } \mathcal{K} \\ \leq & V_{bv}[\pi \rangle C^{\mathcal{K}}] \\ \leq & \textit{Sum of the column maxima of } C^{\mathcal{K}} \times V_{bv}[\pi] \ . \end{array}$$

What this means is that even if the attacker uses machine learning to try to deduce properties about the original data, its ability to do so is constrained by this upper bound.

As an example, suppose there are three possible values a secret can take, drawn from x_a, x_b, x_c, each a distance 1 apart from eachother. [6] A differentially private mechanism $\mathcal{K}$ could release three possible results, say *a, b, c*, with corresponding channel:

$$C^{\mathcal{K}}_{x_i j} \quad = \quad 1/2 \quad \textit{if } i = j \ , \ \textit{else } 1/4 \ .$$

Here $\mathcal{K}$ is $\log 2$-differentially private, since the maximum of $\mathcal{K}.x(\alpha)$ is at most $\max_{j,i' \in a,b,c} C^{\mathcal{K}}_{x_i j} / C^{\mathcal{K}}_{x_{i'} j} \leq \frac{1}{2} / \frac{1}{4} = 2$.

Unfortunately we cannot apply the original definition of differential privacy (10) to text documents because, unlike databases, texts are highly unstructured. Indeed the applicability of differential privacy to text documents has been dismissed [8, 23]. We propose instead to use a generalisation of differential privacy that can apply to unstructured domains, suggesting that we can after all find an obfuscation mechanism based on generalised differential privacy. The trick to generalising differential privacy is to use a general distance function as follows.

Definition 7. *[6] Let $\mathcal{K}: \mathcal{X} \to \mathbb{D}\mathcal{Y}$, and let dist: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a distance function on $\mathcal{X}$. We say that $\mathcal{K}$ is* ϵ-differentially private *with respect to dist if, for all properties α, we must have:*

$$\mathcal{K}.x(\alpha) / \mathcal{K}.x'(\alpha) \quad \leq \quad e^{\epsilon \times dist(x,x')} \ .$$

Def. 7 says that a mechanism imbues privacy by confusing the exact value of a secret x with other values x' with a level proportional to $dist(x, x')$. Thus if x, x' are "close" (as measured by *dist*) then it's quite likely that they could be confused, but if they are far apart, then they would be less likely, although still possibly, be confused.

Putting this together with the channel theorem above, means if we choose ϵ so that $e^{\epsilon \times d(x,x')}$ is as close to 1 as we can make it, then the chance of distinguishing x from x' becomes extremely small.

Even if we do no know the channel matrix exactly, we are still able to obtain a bound on the information leakage.

Theorem 3. *Let $\mathcal{K}$ be an ϵ- generalised differentially private mechanism wrt. metric d. Then for any gain function g,*

$$\mathcal{L}_g(C^{\mathcal{K}}) \quad \leq \quad e^{\epsilon \times d^\star} \ ,$$

where $d^\star := \max_{x,x' \in \mathcal{X}} d(x, x')$.

[6] These could, for example, correspond to different possible data values in a database.

3.6 Privacy versus utility

Information leakage on its own, in the case that it is large, implies that the probability of determining some property of the system will be high; if the upper bound is small, then it implies the mechanism does not leak very much information about anything. When we bring utility into the mix what we want is that the mechanism leaks a lot of information about a property which is not deemed sensitive, but keeps secret some other property that is deemed private. Not surprisingly there are constraints as to how much both requirements can be served simultaneously, however differential privacy can be used as a way to randomise whilst preserving some modicum of utility. We first use some notions from Quantitative Information flow to understand the trade-off between privacy and utility.

Let $\sim_A$ and $\sim_T$ represent two equivalence classes on a set of (secret) data $\mathcal{S}$. We want to release the equivalence class $\sim_T$ but keep $\sim_A$ private using some mechanism M. We can determine how successful we are by measuring the leakage with respect to the two equivalence classes, where we use a specialised version of vulnerability based on the scenario where an adversary tries to guess which equivalence class.

Definition 8. *M is ϵ-hiding wrt. $\sim_A$ if*

$$\mathcal{L}_{bv_{\sim_A}}(M) \leq 1 + \epsilon \ ,$$

where $bv_{\sim_A}$ is defined at (5) and leakage is defined at (6).

The maximum chance of an adversary guessing which equivalence class of $\sim_A$ the secret is for an ϵ-hiding mechanism is bounded above by $(1+\epsilon) \times V_{bv_{\sim_A}}[\pi]$, giving a robust privacy guarantee on $\sim_A$.

Definition 9. *M is Δ-revealing wrt. $\sim_T$ if*

$$1 + \Delta \leq \mathcal{L}_{bv_{\sim_T}}(M) \ ,$$

where $bv_{\sim_T}$ is defined at (5) and leakage is defined at (6).

The best chance of an adversary guessing which equivalence class of $\sim_T$ the secret is for a Δ-revealing mechanism could therefore be *as much as* $(1+\Delta) \times V_{bv_{\sim_T}}[\pi]$.

Theorem 4. *If $M_1 \sqsubseteq M_2$ then the following applies:*

- *If M_1 is ϵ-hiding of $\sim_A$ then so is M_2*
- *If M_2 is Δ-revealing of $\sim_T$ then so is M_1*

Note that when data is provided to the user in a different representation, such as character n-grams, this is called "post-processing"; as noted elsewhere [4] post-processing is an instance of refinement, thus, as Thm. 4 indicates the action of transforming documents into either character n-grams or some other representation provides more privacy and less accuracy for utility.

Next we can look at some constraints between privacy and utility.

Theorem 5. *If $\sim_A \subseteq \sim_T$ and M is both ϵ hiding for $\sim_A$ and Δ revealing for $\sim_T$ (both under a uniform prior) then $\Delta \leq \epsilon$.*

Proof. Note that $\mathcal{L}_{bv_{\sim_T}}(M)$ is equal to $V_{\sim_T}[u\rangle M]/V_{\sim_T}[u]$. But this is bounded above by $N \times V_{\sim_A}[u\rangle M]/V_{\sim_T}[u]$, where N is the size of the maximum equivalence class of $\sim_T$. But now $V_{\sim_T}[u]$ is equal to $N/|\mathcal{S}|$, thus leakage of $bv_{\sim_T}(M)$ is bounded above by $N \times V_{\sim_A}[u\rangle M] \times |\mathcal{S}|$ which is equal to $\mathcal{L}_{bv_{\sim_A}}(M)$. The result now follows.

In particular if $\sim_A = \sim_T$ then revealing any of $\sim_T$ will reveal the same about $\sim_A$. In general if $\sim_A$ is finer than $\sim_T$ (as equivalence relations) revealing the equivalence class for $\sim_T$ almost exactly, already reveals quite a lot about the equivalence classes of $\sim_A$

Consider however the following example where there are four secret values: $\{a, b, c, d\}$. Suppose we have equivalence classes of $\sim_T$ are $\{\{a, b\}, \{c, d\}\}$ and for $\sim_T$ are $\{\{a, c\}, \{b, d\}\}$. The mechanism given by

$$M_{x,y} \quad := \quad 1 \textit{ if } \begin{pmatrix} x \in \{a, b\} \wedge y = 0 \\ \vee\ x \in \{c, d\} \wedge y = 1 \end{pmatrix} \quad \textit{else } 0 \ .$$

has maximum leakage 2, and is 1-revealing wrt. $\sim_T$ and 0-revealing wrt. $\sim_A$; this means that the adversary has maximum chance of 1 of guessing $\sim_T$, but minimal chance of $1/2$ of guessing $\sim_A$.

This suggests that where $\sim_A$ represents equivalence classes over authors, and $\sim_T$ represents equivalence classes over topics, if enough different authors write on the same topic, there is a good chance of being able to disguise the writing style whilst remaining in the same topic.

4 Generalised differential privacy and obfuscation

We can start to bring to bear the above observations to our simplified PAN obfuscation task. In particular we explore whether there are mechanisms whose properties can be understood from the perspective of generalised differential privacy. In our simplified version we imagine that we are already working with a bag-of-words (BoW) representation and our mechanism $\mathcal{K}$ will produce another (randomised) bag-of-words representation, i.e.

$$\mathcal{K} : BoW \to \mathbb{D}BoW \ .$$

Unlike our example above, we can no longer work with clear, a priori equivalence relations for authorship ($\sim_A$) and topic ($\sim_T$). Instead we use, as is done in machine learners, similarity relationships for categorising topics and identifying authors. For topicality we use a metric based on a learned distance between "Word2Vec word embeddings" and its lifting to documents via the "Earth Movers distance" [16], and for authorship we use the "Ruzicka metric". Both have been found experimentally to provide good results in author identification and topic classification.

Word2Vec [21] is a representation of words as a vector of values which, roughly speaking, captures relationships between words in terms of their meanings. Since this is a learned representation its accuracy depends very much on the quality of the documents. Remarkably the representation supports a metric [7] which captures similarity in meaning between words. For example Word2Vec embeddings put "queen" and "monarch" close together, but "monarch" and "engineer" far apart. Using the distance between words defined on Word2Vec representations as a base, the Earth Mover's distance can then be defined to compare documents for topicality. An example is given at Fig. 1.

Definition 10. *Let d, d' be documents represented as bags of words. Define $|d - d'|_T$ to be the word mover's distance between the movement based on the distance between Word2Vec word embeddings.*

Informally, given two documents d, d' represented as bags of words, we let R be a "move relation" so that $R_{w,w'} \in [0, 1]$ represents the proportion of $w \in d$ that corresponds to $w' \in d'$. R is set up so that for each $w' \in d'$, we have $\sum_{w \in d} R_{w,w'} = 1$, and for each $w \in d$, we have $\sum_{w \in d} R_{w',w} = 1$. The cost of the move is given by $\sum_{w,w'} R_{w,w'} \times dist(w, w')$, and the word mover's distance is then the minimum over all such move relations.

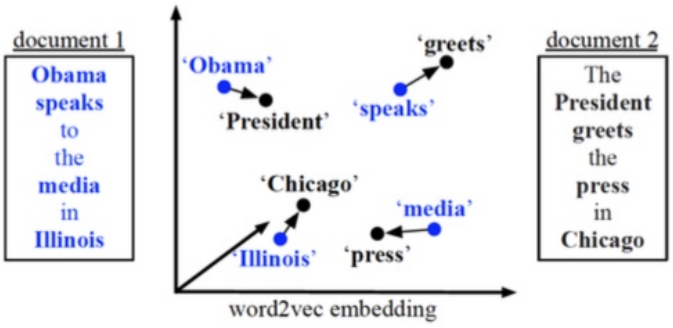

Fig. 1. Depiction of a move relation defining the Word Mover distance [16]

Definition 11. *Let d, d' be documents and d_T be its representation as a a character n-gram vector. In this representation, the vector is composed of discovered "features" which are experimentally found to be good for grouping similar writing styles together. With this in place, we define* $|d - d'|_A := (1 - \frac{\sum_i \min |d_i - d'_i|}{\sum_j \max |d_j - d'_j|})$.

[7] There are several ways to define distance between word embeddings, but "cosine similarity" seems to be a popular one; this isn't a metric, but can be used to define one.

Documents close in the $|\cdot|_A$ metric are likely to be authored by the same author. To obtain a mechanism $\mathcal{K}$ which has a privacy guarantee on obfuscation, we would have the following:

$$\mathcal{K}(d)(\alpha)/\mathcal{K}(d')(\alpha) \quad \leq \quad e^{\epsilon \times dist(d,d')} \ ,$$

for *dist* an appropriate metric. Since this has the form of a differentially private mechanism it would be somewhat resistant to linkage and intersection attacks. Similar to Thm. 3, among distances no more than some fixed K, and $\epsilon \approx 1/10K$ then the right-hand side shows that the entries in each column of the channel for those documents are approximately 1.1, thus suggesting that all such documents would be confused with eachother.

It can also be shown [20] that using the Laplace distribution combined with a given metric *dist* it is possible to define a mechanism M so that the output remains close to the input x with high probability (proportional to ϵ) when measured using *dist*.

4.1 Experiments

Dataset	**Accuracy**	**Obfuscation Accuracy**		
Reuters	Baseline	Scale=0.1	Scale=0.2	Scale=0.5
Raw	71.1	-	-	-
Content-Words	68.5	67.9	67.9	41.7
BOW-1000	65.9	62.1	63.5	41.9
BOW-500	64.1	61.7	62.1	40.9
BOW-200	47.9	46.9	48.5	27.1
BOW-50	23.9	20.0	19.0	6.2
Fan fiction	Baseline	Scale=0.1	Scale=0.2	Scale=0.5
Raw	70.6	-	-	-
Content-Words	67.7	67.7	67.6	4.9
BOW-1000	48.0	35.3	40.2	2.0
BOW-500	46.1	34.3	34.3	5.9
BOW-200	36.3	19.6	18.6	8.8
BOW-50	13.7	4.9	4.9	1.0

Fig. 2. Results for authorship attribution over the various unobfuscated and obfuscated test sets. Uniformly randomly assigning authorship would have an accuracy of 1% over 100 possible authors for the Fan fiction dataset, and 5% over 20 authors for the Reuters dataset.

Using the above observations as a guide, we designed a simple mechanism using BoW representations based on Def. 10 designed therefore to preserve topicality. The idea is to use an underlying Laplace mechanism combined with the Word2Vec distance independently applied to each word in the input bag of words.

Dataset	**Accuracy**	**Obfuscation Accuracy**		
Reuters	Baseline	Scale=0.1	Scale=0.2	Scale=0.5
Raw	81.4	-	-	-
Content-Words	81.4	81.6	81.0	71.9
BOW-1000	80.4	80.8	80.8	75.2
BOW-500	79.2	79.4	79.4	70.7
BOW-200	76.0	76.0	76.0	66.7
BOW-50	66.3	67.9	68.1	61.7
Fan fiction	Baseline	Scale=0.1	Scale=0.2	Scale=0.5
Raw	82.4	-	-	-
Content-Words	83.3	79.4	79.4	54.9
BOW-1000	83.3	77.5	76.5	57.8
BOW-500	81.4	80.4	81.4	63.7
BOW-200	79.4	71.6	71.6	53.9
BOW-50	60.8	49.0	49.0	46.1

Fig. 3. Results for topic classification over the various unobfuscated and obfuscated test sets. Classification accuracy is significantly lower for scale=0.5, which corresponds to more obfuscation. However, accuracy is still well above the 'random' baseline of 20%.

Next we tested the results, both for privacy and for topicality; our hypothesis was that randomising directly on words would mean that the character n-gram representation would be changed sufficiently to hide stylistic traits. Moreover, our theoretical approach shows only that where documents close in topicality can be confused, so therefore can their authors. Authors that are only known for their work on a single topic cannot be confused with authors who write on entirely different subjects.

To test the results we needed large collections of documents written by different authors, and representing a number of different topics. We were able to use one standard dataset from the Natural Language Processing (NLP) literature; a second data set was constructed by us.

1. The Reuters RCV1 dataset is a standard dataset used in language processing tasks, and consists of over 800,000 Reuters news articles separated into various topics [17]. Although not originally constructed for author attribution work, it has been used previously in this domain by making use of the <byline> tags inside articles which designate article authors [22]. The dataset was chosen because it contains documents of reasonable length, which is required for successful author identification. In addition, this dataset is similar to the dataset on which the Word2Vec vectors used in this experiment were trained on, and thus we would expect high quality outputs when using Word2Vec with this data.
2. Our second data set consisted of "Fan fiction" samples[8]. This data set therefore consists of stories collected over the 5 most popular book-based topics. Fan fiction has been used previously in PAN author attribution tasks, and is

[8] https://www.fanfiction.net

suitable for this task because of the content length of the texts and the diversity of authorship styles present in these texts, as stylistic writing qualities are important in this domain.

For each of the documents in the data sets we used our obfuscation mechanism described above to a bag of words representation. We then used appropriate machine learners to try to categorise the results by author and (separately) by topic. In each case we applied the same machine learning techniques to the original (bag of words representations) of the documents to provide a baseline with which to compare.

In Fig. 2 we can see the result of obfuscation: with increasing randomness (as measured by Scale) the ability to identify the author becomes harder, as compared to the Baseline (i.e. unobfuscated documents). This is compared to Fig. 3 which we can see preserves the topicality very well — which is to be expected because of the use of the Laplace mechanism based on Word2Vec.

5 Conclusions and future work

This paper has brought two conceptual ideas together to provide some foundations for privacy mechanisms in text document processing. We used generalised differential privacy based on metrics used in machine learning as a way to create a mechanism, and noted how to understand the privacy that it provides in terms of generalised differential privacy cast in terms of channels for quantitative information flow.

We also observed experimentally that the mechanism seems to preserve topicality well, whilst achieving good privacy. We note here that although we have not provided a mechanism that produces human-readable documents, the mechanism still maintains a variety of words, which fits with the spirit of the PAN obfuscation task.

There is, of course, a long way to go before we have a true summarisation mechanism that is private; with this foundation we have the tools to understand the extent of privacy in future obfuscation mechanisms as they become available.

While the approach outlined in this paper used a simple Word2Vec embedding substitution mechanism over a bag of words representation, there is very promising recent work that uses deep learning to generate paraphrased text, taking text as input. For instance, [12] gives a method for producing syntactically controlled adversarial paraphrases for text: paraphrases that have the goal of confounding a machine learner, which in our context would be an inference attacker; an alternative approach based on generative adversarial networks is described by [27]. Incorporating a DP mechanism, along the lines of the one presented in this paper, is one possible avenue to solving the original obfuscation problem presented in Sec 2.

References

1. M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communication Security (CCS '16)*, pages 303–318, Vienna, Austria, October 24–28 2016.
2. Mário S. Alvim, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy versus quantitative information flow. *CoRR*, abs/1012.4250, 2010.
3. Mário S. Alvim, Konstantinos Chatzikokolakis, Annabelle McIver, Carroll Morgan, Catuscia Palamidessi, and Geoffrey Smith. Additive and multiplicative notions of leakage, and their capacities. In *IEEE 27th Computer Security Foundations Symposium, CSF 2014, Vienna, Austria, 19-22 July, 2014*, pages 308–322. IEEE, 2014.
4. Mário S. Alvim, Kostas Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *Proc. 25th IEEE Computer Security Foundations Symposium (CSF 2012)*, pages 265–279, June 2012.
5. Mário S. Alvim, Andre Scedrov, and Fred B. Schneider. When not all bits are equal: Worth-based information flow. In *Proc. 3rd Conference on Principles of Security and Trust (POST 2014)*, pages 120–139, 2014.
6. K. Chatzikokolakis, M.E. Andrés, N.E. Bordenabe, and C. Palamidessi. Broadening the scope of differential privacy using metrics. In *International Symposium on Privacy Enhancing Technologies Symposium*, volume 7981 of *LNCS*. Springer, 2013.
7. K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, March 2011.
8. Chad Cumby and Rayid Ghani. A machine learning based system for semi-automatically redacting documents. In *Proceedings of the Twenty-Third Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2011.
9. T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15:429–44, 1977.
10. Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.
11. Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273. ACM, 2008.
12. Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *North American Association for Computational Linguistics*, 2018 (to appear).
13. Mahmoud Khonji and Youssef Iraqi. A Slightly-modified GI-based Author-verifier with Lots of Features (ASGALF). In *Working Notes for CLEF 2014 Conference*, 2014.
14. Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *JASIST*, 60(1):9–26, 2009.
15. Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Authorship attribution in the wild. *Language Resources and Evaluation*, 45(1):83–94, 2011.
16. Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 957–966, 2015.

17. David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
18. Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA, 1999.
19. Annabelle McIver, Larissa Meinicke, and Carroll Morgan. Compositional closure for Bayes Risk in probabilistic noninterference. In *Proceedings of the 37th international colloquium conference on Automata, languages and programming: Part II*, volume 6199 of *ICALP'10*, pages 223–235, Berlin, Heidelberg, 2010. Springer.
20. Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103. IEEE, 2007.
21. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.
22. Frederick Mosteller and David L Wallace. Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed federalist papers. *Journal of the American Statistical Association*, 58(302):275–309, 1963.
23. David Sánchez and Montserrat Batet. C-sanitized: A privacy model for document redaction and sanitization. *Journal of the Association for Information Science and Technology*, 67(1):148–163, 2016.
24. Shachar Seidman. Authorship Verification Using the Imposters Method. In *Working Notes for CLEF 2013 Conference*, 2013.
25. Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.
26. Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *CoRR*, abs/1708.02709, 2017.
27. Zhengli Zhao, Dheeru Dua, and Sameer Singh. Generating natural adversarial examples. In *International Conference on Learning Representations*, 2018.