A Machine Learning Based System for Semi-Automatically Redacting Documents

Chad Cumby
Accenture Technology Labs
161 N. Clark St
Chicago, Illinois 60601
chad.m.cumby@accenture.com

Rayid Ghani
Accenture Technology Labs
161 N. Clark St
Chicago, Illinois 60601
rayid.ghani@accenture.com

Abstract

Redacting text documents has traditionally been a mostly manual activity, making it expensive and prone to disclosure risks. This paper describes a semi-automated system to ensure a specified level of privacy in text data sets. Recent work has attempted to quantify the likelihood of privacy breaches for text data. We build on these notions to provide a means of obstructing such breaches by framing it as a multi-class classification problem. Our system gives users fine-grained control over the level of privacy needed to obstruct sensitive concepts present in that data. Additionally, our system is designed to respect a user-defined utility metric on the data (such as disclosure of a particular concept), which our methods try to maximize while anonymizing. We describe our redaction framework, algorithms, as well as a prototype tool built in to Microsoft Word that allows enterprise users to redact documents before sharing them internally and obscure client specific information. In addition we show experimental evaluation using publicly available data sets that show the effectiveness of our approach against both automated attackers and human subjects.The results show that we are able to preserve the utility of a text corpus while reducing disclosure risk of the sensitive concept.

Introduction

There has been a lot of recent interest in methods for protecting the privacy of individuals contained in publicly released structured databases. Approaches such as *K*-anonymity (Sweeney 2002), *L*-diversity (Kifer and Gehrke 2006), and noise based methods have been shown to be effective both in identifying records where possible privacy breaches might occur, and in perturbing, suppressing or generalizing attributes until these records are protected.

In addition to protecting structured databases, organizations such as intelligence agencies, government agencies, and large companies also need to redact sensitive information from unstructured and semi-structured documents before publicly releasing them. In many large enterprises (such as Accenture), redaction needs to take place before documents can be shared even with internal colleagues. These documents might contain sensitive client or project information that cannot be disclosed publicly or internally. Despite this need, there has been very little work focused on techniques to protect sensitive information in text and semi-structured data. The motivation for this work comes from Accenture, large consulting company, where an internal knowledge management system might contains documents describing projects done for various client companies. Often, contractual agreements stipulate that to re-use a document externally or even internally, the identity of the client company as well as specific client confidential information must be removed from the document. Thus a user must remove any uniquely identifying information that an attacker could use to infer the identity of the client. The same kinds of requirements occur in many other domains such as intelligence and healthcare. For example in healthcare specific sensitive topics about patient health such as HIV/AIDS are protected under law, and even when subpoenaed, health records must be manually scrubbed to remove reference to these topics.

In such processes there is necessarily a tradeoff between redacting enough information to protect the sensitive concept, while not over-redacting to the point where the utility of the document for various tasks has been eliminated. The goal of any system designed to help users redact document corpora is to optimally manage this tradeoff and allow users to make informed choices when performing their task.

A major challenge in building such a system is developing precise notions of privacy for text data. Unlike database records, text data does not necessarily contain an explicitly identified sensitive attribute. Instead, the important thing to protect in a document might be that identity of the software that is being used in a project, or that helicopter parts are being produced in the southeast US, or the identity of the document's author, etc. This ambiguity presents difficulty both in identifying the sensitive concepts present, and in modifying the text to protect them.

This paper describes a system for protecting sensitive information in text data, and how it has been implemented in our company to aid in scrubbing project materials before submission to the corporate repository. There are two types of information that the system aids in redacting:

- **Client Identifying Information (CII)** - This information includes any words and phrases that reveal what client company the document pertains to.
- **Personally Identifying Information (PII)** - This in-

cludes any person names, location names, social security numbers, phone numbers, credit card numbers, etc.

To deal with the problem of the CII present in a document, our system casts redaction as new type of problem we call Text Inference Blocking, in which we use machine learning to train a classifier to recognize the sensitive client identity given the text of a document, and then perturb the text to defeat this classifier in a specific way. This perturbation is formulated to optimally maintain a utility metric for the document and is described in detail in this paper.

To identify and redact PII our system uses currently available statistical NLP tools such as the Stanford Named Entity recognizer, as well as simple template based recognizers for sensitive numbers.

In our deployed document redaction system, the redaction algorithms for CII and PII are used in an interactive fashion in which a document is analyzed to suggest sets of words to suppress or generalize for particular instances, and a human supervisor reviews the suggestions. Our system also allows for a batch redaction mode, which we simulate in experiments shown on two standard text categorization data sets. We present several algorithms that fit into our framework and empirically compare their performance on several data sets. We also present a user study that shows that our algorithms are not only effective in protecting against large-scale automated attacks but also work against human attacks.

General Problem Formulation

We treat the detection of a sensitive concept (such as CII) as a multi-class classification problem and present several algorithms that allow varying levels of redaction to take place. We assume the set of sensitive and utility concepts are known in advance, but this work can be extended to cover cases where new concepts can be built dynamically. An attacker who then uses either automatic or manual methods to deduce the sensitive concept is classifying the document into the concept set. Our work attempts to optimally perturb a document to maximize the classification error for the sensitive class within this set. In the terminology of (Chow, Golle, and Staddon 2008) we are performing *Inference Detection* as well as attempting to *block* these inferences. In addition we seek to perturb the document while preserving its *utility* (using a number of different metrics).

Thus for example, the set of features/words necessary to suppress in order to obscure the fact that a document is about *Ford* and not *GM* – while retaining information that conveys that this document is about automobiles – is very different than the set of words which indicate *Ford* with high support.

Depending on the sensitivity of the materials in question, redacting a document until the true class is obscured within a set of two classes may be insufficient. By analogy with the k-anonymity framework, some usage scenarios require a higher k value. For example, if a software company wished to scrub a set of job postings to obscure its activities in the mobile phone domain from a competitor, it might want to make a posting confusable with only the closest category, say medical device programming. However, an attacker inferring that the company is hiring for either of these two skills may still be unacceptable. In this situation redacting within a larger confusion set is necessary.

We present several algorithms based on the model of maximizing classification error for a sensitive class, under the umbrella term *Text Inference Blocking*. Some of our techniques allow a user to specify the size k of the confusion set which the true class must be indistinguishable with, which we denote as *k-confusability*. Our basic algorithms also treat the case where a known utility metric is specified for a document, and we find an optimal solution in which $k - confusability$ is achieved while maximizing this metric.

Related Work

Manual document sanitization has a long history in government and industry, with many guideline documents published (*eg* (NSA-SNAC 2005)) describing the correct procedures for redacting physical documents, as well as many types of electronic documents, to ensure that the deletion or masking of key sections is irreversible. In addition the cryptography community has been very active to provide security mechanisms (such as (Haber et al. 2008)) that guarantee through digital signatures that a document has been redacted without any malicious tampering.

From the perspective of automatic techniques for aiding the document sanitization process via data mining, machine learning, and related techniques, the research community has just begun to address the problem within the past few years.

A number of authors (Terrovitis, Mamoulis, and Kalnis 2008) have adapted methods such as k-anonymity to ”unstructured” data by treating text data as a variable length database record, or set of untyped values, with the assumption that the sensitive value to protect is deterministically identified by a set of quasi-identifier words. In (Chakaravarthy et al. 2008) Chakaravarthy et al describe an approximate set-covering procedure that attempts to delete terms from a document that deterministically identify an entity of interest. Many of these approaches are geared towards the problem of anonymizing search engine query logs.

A variation, Differential Privacy (Dwork 2006) is a framework for providing provably secure, private, results from a statistical database. Queries on aggregate statistics about arbitrary datatypes are performed against the original data, and the results perturbed through noise addition, to fulfill the guarantee that query results will be the same within some given ϵ margin after the deletion of any record. Unfortunately this framework places the heavy restriction that no form of an original element of the dataset can be made public directly. Thus all utility for the types of tasks which require direct access to sanitized text is lost in this setup.

Recently, Chow et al (Chow, Golle, and Staddon 2008; Chow, Oberst, and Staddon 2009) addressed the problem of inference detection for sensitive topics by building association rules mined from web query results. These rules provide a ranking of words in a test document in terms of how much support a topic rule containing the word has. Yet it does not deal with many of the difficult issues in stopping these inferences, such as the tradeoff in utility of the data after it is modified. In (Dalvi et al. 2004) Dalvi et al described a utility sensitive adversarial optimization in the context of designing a learning algorithm to defeat the adversary.

Recent work by Jones et al (Jones et al. 2007; 2008) has shown that when dealing with text, treating the data as sets of tokens and applying adaptations of techniques from the database community (k-anonymity, etc.) is often insufficient. In (Jones et al. 2007) they showed that by training simple classifiers and regressors, they could reconstruct quasi-identifiers such as age, zip code, and gender, which then could be used to re-identify the authors of specific queries despite token set based anonymization. Detection and obstruction of these types of inferences using learned classifiers is at the heart of our technique.

Redaction System

As mentioned earlier, in many large companies it is necessary to remove Client Identifiable Information (CII) and Personally Identifiable Information (PII) before reusing business materials from other projects. For CII, in order to comply with client contracts it is insufficient to simply remove the canonical name of the client. Abbreviations of the company name as well as any uniquely identifying information (locations or products uniquely associated with the company for example) that could be used to deduce the identity of the company also need to be removed, along with several other kinds of information.

To aid in this currently very labor intensive activity, we have developed a redaction system that assists in the process. The system is composed of a back-end service layer that can apply redaction methods to input text, either in batch or interactive mode, and a front-end tool that is currently implemented as a MS-Office add-in. A diagram of the system is shown in Fig. 1. The back-end service layer includes the classifier models used for redaction, plus the different algorithmic components used to identify and score PII and CII terms.

In the first implementation of our system, we have created CII models for 450 of our company's clients from documents in a 350000 document corpus. In addition to the novel redaction methods for CII which form the bulk of our work, the back-end services layer also processes the text to recognize several classes of PII as follows:

- Named Entity Recognizer - Utilizes the state-of-the-art Stanford Named Entity Recognizer (Finkel, Grenager, and Manning 2005), which is a Conditional Random Field model trained to recognize *people*, *organizations*, and *locations*.
- US Social Security Numbers - Template-based regular expression model.
- Credit Card Numbers - Template-based regular expression model.

The front-end client application (shown in Fig. 2) is designed to help a user quickly redact a document interactively. When a document is loaded, the Redaction add-in builds the word frequency vector of the text in the document and returns suggested CII and PII terms to redact. The CII terms are scored by the SIMPLEREDACTOR algorithm (described in next section) in order to identify the terms that give away the identity of the client. Our application allows the user to adjust the level of redaction. The user can move the slider to change the yellow-highlighted redaction list. For example if the true client is Carrefour and the next closest client model is for Danone, the list contains the word "Carrefour" and client specific terms that distinguish the document from those regarding Danone. If the user increases the slider to indicate that they want to lower the redaction threshold it will suggest more general words such as "retail" or "Paris".

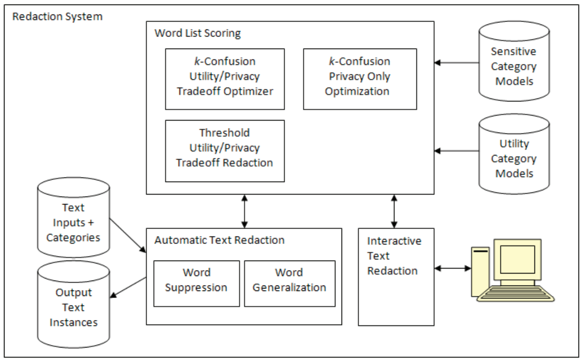

Redaction System
Word List Scoring
k-Confusion Utility/Privacy Tradeoff Optimizer
k-Confusion Privacy Only Optimization
Threshold Utility/Privacy Tradeoff Redaction
Sensitive Category Models
Utility Category Models
Text Inputs + Categories
Output Text Instances
Automatic Text Redaction
Word Suppression
Word Generalization
Interactive Text Redaction

Figure 1: Redaction system functional architecture.

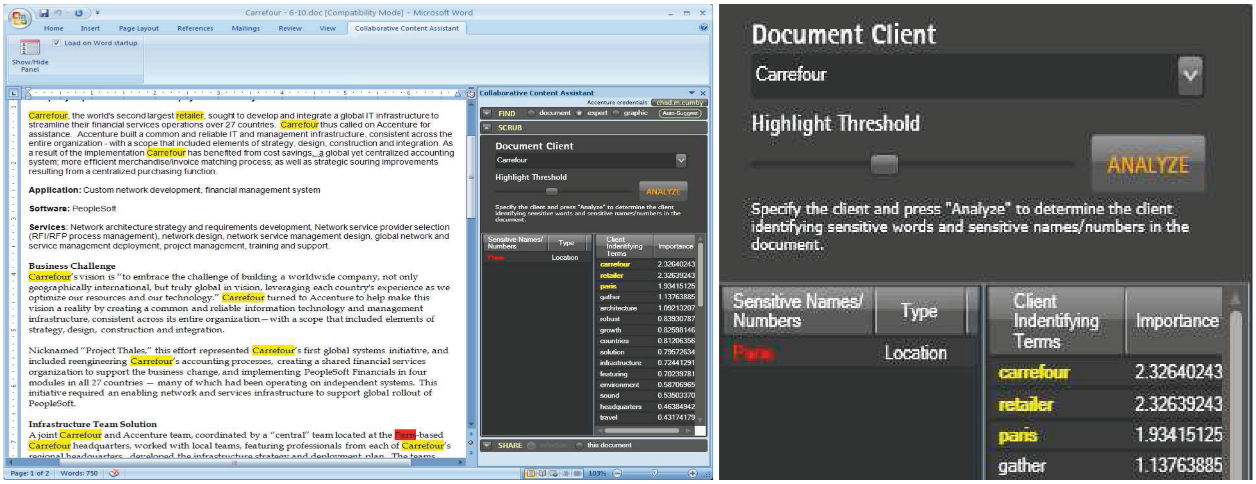
In the screenshots pictured below in Fig. 2, we show the prototype implementation that has been deployed within Accenture, where a user is attempting to redact the document to remove words indicating that Carrefour is the client (as opposed to Danone). A number of the top most sensitive CII and PII terms returned by our system have been highlighted. A user can also access a number of options by right-clicking on an individual sensitive term in the list, such as navigating to each occurrence of the term or automatically redacting each with a placeholder.

Redacting CII with Inference Blocking

We now describe the problem of Text Inference Blocking for redacting a set of documents. We have a set D of documents, where each $d \in D$ can be associated with a sensitive category $s \in S$. In addition each document can be associated with a finite subset of non-sensitive utility categories $U_d \subset U$. We assume that an external adversary has access to a disjoint set of documents D', each of which is associated with some $s \in S$ and some subset of the utility categories U.

For a document d, we define the problem of obscuring the sensitive category s while preserving the identity of the utility categories U_d in a standard multi-class classification framework. (d, s) pairs are generated i.i.d. according to some distribution $P_S(d, s)$, and (d, U_d) according to $P_U(d, U_d)$. Generally s and U_d are not independent given d. Our goal is to define an *inference blocking* function $Redact : D \to D$ that minimizes $P_S(Redact(d), s)$ and maximizes $P_U(U_d|Redact(d))$.

In our example, an employee wants to share a set of documents that are about projects in specific industries for spe-



Carrefour, the world's second largest retailer, sought to develop and integrate a global IT infrastructure to streamline their financial services operations over 27 countries. Carrefour thus called on Accenture for assistance. Accenture built a common and reliable IT and management infrastructure, consistent across the entire organization - with a scope that included elements of strategy, design, construction and integration. As a result of the implementation Carrefour has benefited from cost savings, a global yet centralized accounting system; more efficient merchandise/invoice matching process; as well as strategic souring improvements resulting from a centralized purchasing function.

Application: Custom network development, financial management system

Software: PeopleSoft

Services: Network architecture strategy and requirements development, Network service provider selection (RFI/RFP process management), network design, network service management design, global network and service management deployment, project management, training and support.

Business Challenge
Carrefour's vision is "to embrace the challenge of building a worldwide company, not only geographically international, but truly global in vision, leveraging each country's experience as we optimize our resources and our technology." Carrefour turned to Accenture to help make this vision a reality by creating a common and reliable information technology and management infrastructure, consistent across its entire organization—with a scope that included elements of strategy, design, construction and integration.

Nicknamed "Project Thales," this effort represented Carrefour's first global systems initiative, and included reengineering Carrefour's accounting processes, creating a shared financial services organization to support the business change, and implementing PeopleSoft Financials in four modules in all 27 countries — many of which had been operating on independent systems. This initiative required an enabling network and services infrastructure to support global rollout of PeopleSoft.

Infrastructure Team Solution
A joint Carrefour and Accenture team, coordinated by a "central" team located at the Paris-based Carrefour headquarters, worked with local teams, featuring professionals from each of Carrefour's

Accenture credentials: chad.m.cumby

Document Client: Carrefour

Highlight Threshold

Specify the client and press "Analyze" to determine the client identifying sensitive words and sensitive names/numbers in the document.

Sensitive Names/ Numbers	Type
Paris	Location

Client Indentifying Terms	Importance
carrefour	2.32640243
retailer	2.32639243
paris	1.93415125
gather	1.13763885
architecture	1.09213207
robust	0.83930787
growth	0.82598146
countries	0.81206356
solution	0.79572634
infrastructure	0.72441291
featuring	0.70239781
environment	0.58706965
sound	0.53503370
headquarters	0.46384942
travel	0.43174179

Figure 2: Screenshot of redaction system with detail on right.

cific clients. They are not allowed to disclose the name of the client but they would like to make sure the industry of the client is obvious. In our system, the *client* becomes the sensitive category and needs to be obscured while the *industry* of the client becomes the utility category. The goal for $Redact$ is to redact each document so that the privacy is maximized while minimizing the utility loss. $Redact$ needs to maximize the reduction in the conditional probability of the true sensitive category (client in this case) given the document and minimize the reduction in the conditional probability of the true utility category (industry in this case) given the document. In order to provide constraints for this problem there are a number of alternative formulations that we explore below.

Features and Sanitization Operators To develop a practical inference blocking algorithm for documents based on minimizing the likelihood of a document and category pair, we must first define a representation for each document and operators with which to perturb it. We model a document d as a feature vector $x =< x_1..x_n >$ with a finite space of n features. For most of our discussion we will concentrate on a bag-of-words model for representing d, with each word appearing in the document set represented as a binary feature x_i. We envision natural variations on our methods which could treat the document as a set of factors in a topic model (LDA, etc), n-grams of words, or other linguistic features, depending on the type of sensitive information that is being scrubbed.

Any feature representation chosen must have corresponding operators used to perturb the document based on the inference blocking algorithm. Two natural ones also seen in anonymization work on structured data with categorical values are suppression and generalization. In suppression, certain 1-valued features are set to 0, corresponding to these words being removed from the document in our setting. Generalization operators for text must rely on domain specific taxonomies of linguistic features – eg for words, WordNet may be used to find a generalization by following the "IsA" hierarchy to find a semantic abstraction. A concern with generalization operators for text is that to model some types of utility categories, it could be possible to use a different feature representation. In an inference blocking algorithm with a generalization operator, a mapping function from one feature space to the other would be necessary.

Inference Blocking Algorithms To model the joint distribution $P_S(D,S)$ for different types of sensitive categories that are associated with text documents in a way that allows us to minimize $P_S(Redact(d),s)$ and maximize $P_U(U_d|Redact(d))$, different assumptions about the form of the distribution given a feature-space can be made. For longer documents where the sensitive category is a known topic, such as documents about "Ford", the Naive Bayes model with the bag-of-words features is an effective classifier. Thus we model $P_S(D,S)$ for a given doc/category pair as:

$$\begin{aligned} P_S(s,d) &\propto \qquad\qquad\qquad\qquad (1) \\ P_{nb}(s|x) &= \frac{P_{nb}(s)P_{nb}(x|s)}{P_{nb}(x)} = \frac{P_{nb}(s)}{P_{nb}(x)}\prod_i^n P_{nb}(x_i|s) \end{aligned}$$

We model the likelihood of each category of a given (d, U_d) similarly in an independent fashion according to Eq. 1. Based on the above model we develop our inference blocking algorithms.

The following formulation deals explicitly with suppression as the method of sanitizing, although it could be extended to use generalization operators. Here our intuition is that for each document, we can use generative models (Naive Bayes in this case) to identify the features present in a given document instance that imply the sensitive category less than the utility categories, and sanitize enough of them to obscure the sensitive one.

K-Confusability One important constraint we place on the redaction process, in order to avoid over-redaction, is to require that the Naive Bayes likelihood from Eq. 1 of the true sensitive category for a sanitized document example $Redact(x)$ be less than the likelihood of k other categories.

Definition 1 *For a learned multiclass classifier H outputting a total ordering $\pi = y_1 \succ \ldots \succ y_n$ over n classes for a given example x with true class y, we say a new example $\hat{x}$ is k-*confusable *with x if $H(\hat{x})$ outputs an ordering $\hat{\pi}$ with at least k classes preceding y.*

Similar to the case of k-anonymity (Sweeney 2002), our definition does not place constraints on the inference blocking algorithm to apply to an example x. Rather, for a given classifier our criteria is that at least k other classes must be predicted as the class of $\hat{x}$ before y. In some sense this criteria is not as strong as guarantees such as ϵ-differential privacy. However, with the assumption that an attacker has no information about the classifier used to model $P(S|D)$ or the inference blocking procedure used, our criteria is empirically sufficient to defeat many attack scenarios (see the Human Subject experiment later) Additionally, in the future we hope to strengthen our criteria to provide bounded guarantees on the likelihood of any classifier inferring the true class of a k-confusable example in certain concept classes.

Our basic procedure for redacting text documents to ensure k-confusability for sensitive categories is to develop a linear program we call K-REDACTOR, creating a k-confusable example $\hat{x} = Redact(x)$ that is still recognizable as belonging to the utility class u. In this section we will consider only a single utility class u per example x. Here, let $\bar{s} = \bar{s}_1, \ldots, \bar{s}_{k-1} \in S$ be the sequence of $k-1$ classes obtained by ranking all $\bar{s} \in S \setminus s$ by $P(\bar{s})P(x|\bar{s})$.

K-REDACTOR:

$$\begin{aligned} &min\ f(\hat{x}) = -\sum_{x_i} Utility(x_i, u)\hat{x}_i \qquad (2)\\ &\quad s.t.\ 0 \leq \hat{x}_i \leq freq(x_i),\\ &\quad \sum_{x_i} (log(P(x_i|s))\hat{x}_i) \leq \sum_{x_i} log(P(x_i|\bar{s}_1))\hat{x}_i\\ &\quad \vdots\\ &\quad \sum_{x_i} (log(P(x_i|s))\hat{x}_i) \leq \sum_{x_i} log(P(x_i|\bar{s}_{k-1}))\hat{x}_i \end{aligned}$$

where

$$\begin{aligned} Utility(x_i, u) &= OVALogOdds(x_i, u)\\ &= (1 - P(u))log(P(x_i|u))\\ &- \sum_{\bar{u} \in U \setminus u} P(\bar{u})log(P(x_i|\bar{u})) \end{aligned}$$

In this procedure, the objective is to maximize a one-vs-all version of the Naive Bayes decision criterion (OVALogOdds) for the true utility class u with respect to the rest of the utility classes $\bar{u} = U \setminus u$ (see (Rennie et al. 2003)). We re-weight the feature class-conditional likelihood of the true class to be equal to the sum of the prior weights from the "complement" classes. The constraints on the linear program ensure that if a feasible solution exists, k-confusability for our model classifier is guaranteed.

Next we consider the sub-case where k-confusability is desired for some set of examples, without a corresponding set of utility categories. Here we attempt to minimize the amount of redaction, while maintaining the constraints. Thus we substitute the objective as $Utility(x_i) = 1$. This procedure can be approximated by the simple greedy algorithm shown here:

SIMPLEREDACTOR:
For a document example x of class s create an ordered list of features to suppress using the metric:

$$(1 - P(s))log(P(x_i|s)) - \sum_{\bar{s}_i} P(\bar{s}_j)log(P(x_i|\bar{s}_j))$$

From this list, suppress words from x until the conditional log-likelihood (LL) of $s|\hat{x}$ is less than the LL of $k-1$ other classes.

Experiments & Results

In this section we describe our experiments to test the effectiveness of our text inference blocking methods for obscuring sensitive categories. We first test the effects of different parameter settings on learned automated classifiers as a mode of attack. If our system can foil these classifiers, then an attacker scanning for sensitive information in a corpus of masked documents using them would be deterred. Although we are primarily concerned with large-scale automated privacy attacks by adversaries who would use learned classifiers, we also show below that the performance of learned classifiers seems to correlate with human performance in defeating our redaction. This allows our techniques to also defeat human attacks.

Our work was motivated by the need for redaction in large enterprises and government agencies. Our initial experiments were done on our internal document repository that contained over 100,000 documents where the goal was to redact the client identity. Due to restrictions on sharing that data set and hence the inability of other researchers to replicate our results, we also experiment with the well known Industry-Sector data set introduced in (McCallum et al. 1998), and the 20 Newsgroups data set (Lang 1995). Industry-Sector contains 6440 documents corresponding to company websites in a two level hierarchy of 103 industry classes organized into 12 sector categories. 20 Newsgroups contains 19997 posts across 20 categories corresponding to the originating newsgroup. For both data sets we first removed stopwords from a standard list, and extracted unigram word features after removing words that only appeared once. As many studies have shown the benefits of feature selection for text categorization with Naive Bayes (Yang and Pedersen 1997), in all cases we limit the size of the feature space to the top 10000 features ranked by mutual information with the document class.

Industry-Sector Experiments

For the Industry-Sector dataset we treat the leaf level of the hierarchy (industries) as the set of sensitive categories that we would like to obscure, and the sector level as the set of utility categories that should be preserved. Since these two sets of categories are very related, this is a suitable data set to test our methods' ability to balance the privacy/utility tradeoff. From a practical perspective, this task is very similar to the task of obscuring what company a document pertains to, while preserving information about the industrial sector it belongs to.

For these experiments, we trained a Naive Bayes classifier for the 103 industry categories and another for the 12 sector categories using all 6440 documents. These classifiers are used to model the likelihoods for the inference blocking procedure shown earlier.

Our methodology to simulate the scenario of an attacker trying to defeat our inference blocking method and infer the sensitive category for a document relies on two assumptions. First, the attacker has access to the same dataset as the sanitizer along with the industry and sector labels, *minus* the document which is being redacted. Second, we assume an attacker has no information about the inference blocking used or its parameters.

We apply our inference blocking method K-Redactor to the Indusry-Sector examples, treating the industry category as the sensitive class, and the sector category as the utility class. To evaluate the effects of each method, we use the leave-one-out procedure by training Naive Bayes on the entire corpus except each redacted document, and testing on the redacted documents. These classifiers simulate an outside party applying a learned classifier to recognize the sensitive and utility categories of each document.

We varied k between 1 and 5 and applied the utility-maximizing K-Redactor optimization of Eq. 2 to each document of the Industry-Sector corpus. We show the sensitive class (industry) error vs. utility class (sector) accuracy for three settings of k using the Naive Bayes test classifier in Fig. 3 below. For each number of guesses i along the x-axis, we examine the i most likely classes returned by the industry and sector Naive Bayes classifiers. If the true class is within that set we count it as a true positive for each measure. We see high error on the industry class up to k guesses and then a sharp drop-off, with high accuracy on the sector classification.

We also would like to compare the results for K-Redactor obtained by substituting our objective with the OddsRatio, FreqLogP, and InfoGain feature scores relative to the utility class set. To do this we define an evaluation metric *k-eval* that reflects our goal of k-confusability: the average of sector accuracy and industry error-rate up to $k-1$ guesses, and industry accuracy at k guesses. For a set of redacted document examples $\hat{X}$ let $Acc_{sec}(k, \hat{X})$ be the number $\hat{X}$ for which the top k classes returned by the sector classifier contained the true class, and $Acc_{ind}(k, \hat{X})$ be the number where the top k returned by the industry classifier did. Then

$$\text{k-eval}(k, \hat{X}) = \frac{Acc_{sec}(k, \hat{X}) + (1 - Acc_{ind}(k, \hat{X})) + Acc_{ind}(k+1, \hat{X})}{|\hat{X}|}$$

We average this metric over $k = 2 \ldots 5$ and report the results for each objective function in Table. 1.

Objective Function	k-Eval	s Cat Error	u Cat Accuracy	Suppressed %
OVALogOdds	**.834**	.683	.861	.524
OddsRatio	.599	.706	.388	.303
FreqLogP	.553	.925	.572	.987
InfoGain	.599	.706	388	.302

Table 1: Industry-Sector results for K-Redactor.

Experiment with human subjects

Here we show the results of our method evaluated with a human user study. Again this is not how the inference blocking would be applied, rather we want to see how similar the results from simulated attackers (classifiers) are to human attackers. For this experiment the 20 Newsgroups dataset was used. The dataset was redacted using the SimpleRedactor procedure with the k parameter set to 1(unredacted), 2, and 5 - into separate test sets. 50 human subjects recruited from our organization were randomly shown redacted posts from one of the 5 test sets in the context of an online game we called "Redactron", which instructed a subject to pick the most likely set of newsgroups from which the post originated. Subjects could pick multiple answers for each post, up to all 20 groups. In the game they would receive 1 point for a correct answer, and -.05 points for each incorrect group option picked. Thus it was in the players' interest to pick as many answers needed to guess the correct category for each post, but no more. In all 1154 examples were labeled in this fashion. In Fig. 4 we show average error rates on this task broken down by how many guesses were made by the subjects. We see that at higher k levels, error rates for respondents selecting 1 answer option was much higher than those selecting 2+ answer options. Players marking 4+ options exhibited mixed results, and generally very few posts had this many answers.

In Table 1 we show an example of the top 10 words suppressed from a document from the alt.atheism newsgroup by SimpleRedactor with $k = 2$ and $k = 5$. The list of words to confuse the document with the next most likely category, soc.religion.christianity, is very different than the one to confuse it with the whole top 5. In particular the entire $k = 2$ list does not contain the words *faith* or *christianity*, since these words do not distinguish alt.atheism documents from soc.religion.christianity documents. In Fig. 5 we show an example redacted post from the sci.med newsgroup.

Discussion

These experiments' aim was to show the range of performance for our text inference blocking framework on some simple privacy vs utility tasks. With the Industry-Sector experiments we demonstrated that our algorithms have the ability to block the inference of the sensitive class (industry), while maintaining the identity of the utility class (sector). Our k-confusability experiment shows that our optimization has the ability to provide fine-grained control over the level

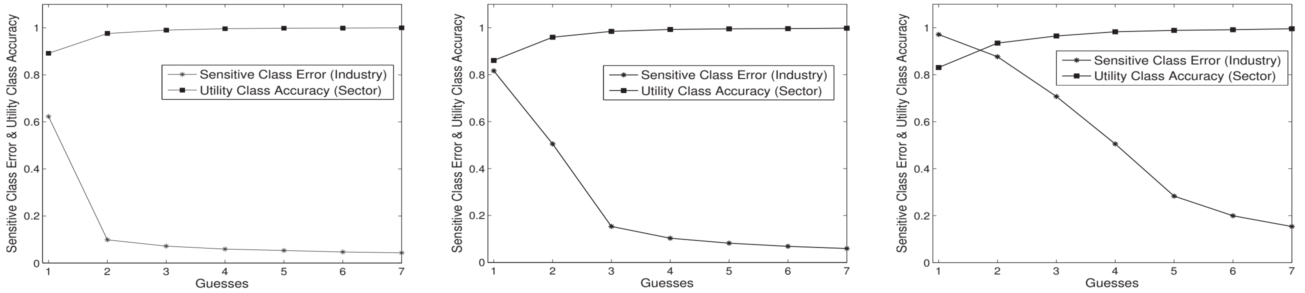


Figure 3: K-REDACTOR Industry-Sector results for $k = 2$ (left), $k = 3$(right), $k = 5$(bottom).

k=2	k=5
religion	faith
dogma	christian
system	dogma
encourages	system
toronto	encourages
humans	beliefs
beliefs	humans
genocide	secular
philosopher	philosopher
prison	christianity

Table 2: Top 10 words to obscure alt.atheism document when $k = 2$ and $k = 5$.

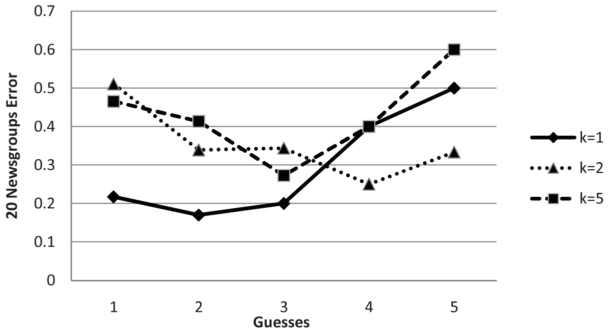
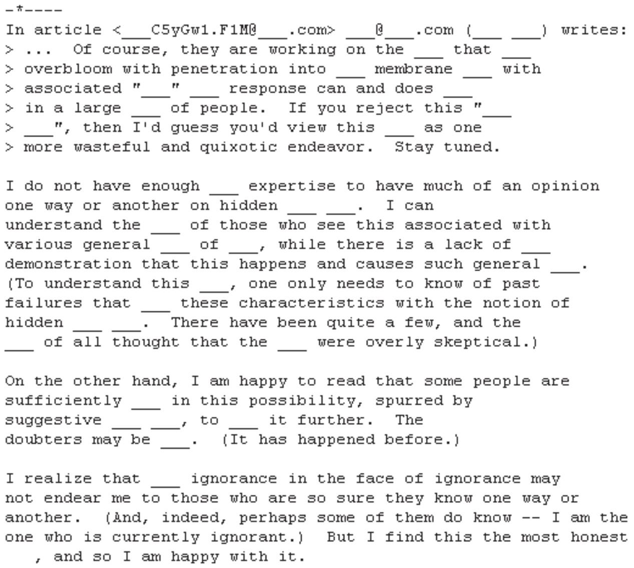


Figure 4: Error rates human subject 20 Newsgroup experiment.

of obscurity for a sensitive category with the intuitive notion of a confusion set, while maintaining high levels of utility. In the human subject experiment, we showed that human attackers may be affected by our inference blocking methods in a similar manner to that of the automatic test classifiers.

Conclusion & Future Work

In this work we have introduced a privacy framework for protecting sensitive information in text data, and presented an implementation designed to aid in redacting client information from enterprise business documents. We believe that protecting sensitive information in text is an area of growing importance as text data sources become larger and business needs for data sharing and integration become more acute. Additional contributions in this paper include the text privacy framework, algorithms for achieving privacy while maximizing utility, and experimental results using automated and human attack models.



```
-*----
In article <___C5yGw1.F1M@___.com> ___@___.com (___ ___) writes:
> ...  Of course, they are working on the ___ that ___
> overbloom with penetration into ___ membrane ___ with
> associated "___" ___ response can and does ___
> in a large ___ of people.  If you reject this "___
> ___", then I'd guess you'd view this ___ as one
> more wasteful and quixotic endeavor.  Stay tuned.

I do not have enough ___ expertise to have much of an opinion
one way or another on hidden ___ ___.  I can
understand the ___ of those who see this associated with
various general ___ of ___, while there is a lack of ___
demonstration that this happens and causes such general ___.
(To understand this ___, one only needs to know of past
failures that ___ these characteristics with the notion of
hidden ___ ___.  There have been quite a few, and the
___ of all thought that the ___ were overly skeptical.)

On the other hand, I am happy to read that some people are
sufficiently ___ in this possibility, spurred by
suggestive ___ ___, to ___ it further.  The
doubters may be ___.  (It has happened before.)

I realize that ___ ignorance in the face of ignorance may
not endear me to those who are so sure they know one way or
another.  (And, indeed, perhaps some of them do know -- I am the
one who is currently ignorant.)  But I find this the most honest
   , and so I am happy with it.
```

Figure 5: Example sci.med newsgroup post redacted in the “Redactron” experiment.

In the future we’d like to extend aspects of this framework in several ways: experimenting with different feature representations, both in the internal model of our sensitive information and in simulating attackers; working with generalization operators in our models explicitly; modeling more complicated inferences and sensitive concepts such as entities and relations, which might require modeling structured inference and learning; and also to work out a more thorough theory for our inference blocking framework, with guarantees on the hardness of reverse engineering our redaction.

Acknowledgments Thanks to Yaron Rachlin for his contributions to an earlier version of this work.

References

Chakaravarthy, V. T.; Gupta, H.; Mohania, M. K.; and Roy, P. 2008. Efficient techniques for document sanitization. In *Proceedings of CIKM-2008*.

Chow, R.; Golle, P.; and Staddon, J. 2008. Detecting privacy leaks using corpus-based association rules. In *Proceedings of KDD-2008*.

Chow, R.; Oberst, I.; and Staddon, J. 2009. Sanitization's slippery slope: The design and study of a text revision assistant. In *Proceedings of SOUPS-2009*.

Dalvi, N.; Domingos, P.; Mausam; Sanghai, S.; and Verma, D. 2004. Adversarial classification. In *Proceedings of KDD-2004*.

Dwork, C. 2006. Differential privacy. In *Proceedings of ICALP-2006*.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL-2005)*, 363–370.

Haber, S.; Hatano, Y.; Honda, Y.; Horne, W. G.; Miyazaki, K.; Sander, T.; Tezoku, S.; and Yao, D. 2008. Efficient signature schemes supporting redaction, pseudonymization, and data deidentification. In *Proceedings of ASIACCS-2008*, 353–362.

Jones, R.; Kumar, R.; Pang, B.; and Tomkins, A. 2007. I know what you did last summer: Query logs and user privacy. In *Proceedings of CIKM-2007*.

Jones, R.; Kumar, R.; Pang, B.; and Tomkins, A. 2008. Vanity fair: Privacy in querylog bundles. In *Proceedings of CIKM-2008*.

Kifer, D., and Gehrke, J. 2006. l-diversity: Privacy beyond k-anonymity. In *Proceedings of ICDE-2006*.

Lang, K. 1995. Newsweeder: Learning to filter netnews. In *Proceedings ICML-95*.

McCallum, A. K.; Rosenfeld, R.; Mitchell, T. M.; and Ng, A. Y. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of ICML-1998*, 359–367.

NSA-SNAC. 2005. Redacting with confidence: How to safely publish sanitized reports converted from word to pdf. Technical Report I333-015R-2005, Information Assurance Directorate, National Security Agency.

Rennie, J. D. M.; Shih, L.; Teevan, J.; and Karger, D. R. 2003. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of ICML-2003*.

Sweeney, L. 2002. Achieving k-anonymity privacy protection using generalization and suppression. *Intl. Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10.

Terrovitis, M.; Mamoulis, N.; and Kalnis, P. 2008. Privacy-preserving anonymization of set-valued data. *Proceedings of VLDB Endow.* 1(1).

Yang, Y., and Pedersen, J. O. 1997. A comparative study on feature selection in text categorization. In *Proceedings of ICML-97*.