Web-Based Inference Detection

Jessica Staddon[1]*, Philippe Golle*[1]*,*
Bryce Zimny[2]*

[1] *Palo Alto Research Center*
{*staddon,pgolle*}*@parc.com*

[2] *University of Waterloo*
bzzimny@student.cs.uwaterloo.ca

Abstract

Newly published data, when combined with existing public knowledge, allows for complex and sometimes unintended inferences. We propose semi-automated tools for detecting these inferences prior to releasing data. Our tools give data owners a fuller understanding of the implications of releasing data and help them adjust the amount of data they release to avoid unwanted inferences.

Our tools first extract salient keywords from the private data intended for release. Then, they issue search queries for documents that match subsets of these keywords, within a reference corpus (such as the public Web) that encapsulates as much of relevant public knowledge as possible. Finally, our tools parse the documents returned by the search queries for keywords not present in the original private data. These additional keywords allow us to automatically estimate the likelihood of certain inferences. Potentially dangerous inferences are flagged for manual review.

We call this new technology Web-based inference control. The paper reports on two experiments which demonstrate early successes of this technology. The first experiment shows the use of our tools to automatically estimate the risk that an anonymous document allows for re-identification of its author. The second experiment shows the use of our tools to detect the risk that a document is linked to a sensitive topic. These experiments, while simple, capture the full complexity of inference detection and illustrate the power of our approach.

1 Introduction

Information has never been easier to find. Search engines allow easy access to the vast amounts of information available on the Web. Online data repositories, newspapers, public records, personal webpages, blogs, etc., make it easy and convenient to look up facts, keep up with events and catch up with people.

On the flip side, information has never been harder to hide. With the help of a search engine or web information integration tool [45], one can easily infer facts, reconstruct events and piece together identities from fragments of information collected from disparate sources. Protecting information requires hiding not only the information itself, but also the myriad of clues that might indirectly lead to it. Doing so is notoriously difficult, as seemingly innocuous information may give away one's secret.

To illustrate the problem, consider a redacted biography [8] (shown in the left-hand side of figure 6) that was released by the FBI. Prior to publication, the biography was redacted to protect the identity of the person whom it describes. All directly identifying information, such as first and last names, was expunged from the biography. The redacted biography contains only keywords that apply to many individuals, such as "half-brother", "Saudi", "magnate" and "Yemen". None of these keywords is particularly identifying on its own, but in aggregate they allow for near-certain identification of Osama Bin Laden. Indeed, a Google search for the query "Saudi magnate half-brother" returns in the top 10 results, pages that are all related to the Bin Laden family. This inference, as well as potentially many others, should be anticipated and countered in a thorough redaction process.

The need to protect secret information from unwanted inferences extends far beyond the FBI. In addition to intelligence agencies and the military, numerous government agencies, businesses and individuals face the problem of insulating their secrets from the information they disclose publicly. In the litigation industry for example, information protected by client-attorney privilege must be redacted from documents prior to disclosure. In the healthcare industry, it is common practice and mandated by some US state laws, to redact sensitive information

*This work was done while a coop student at the Palo Alto Research Center.

(such as HIV status, drug or alcohol abuse and mental health conditions) from medical records prior to releasing them. Among individuals, anonymous bloggers are a good example of people who seek to ensure that their posts do not disclose their secret (their identity). This is made challenging by the fact that in some cases very little personal information may suffice to infer the blogger's identity. For example, if the second author of this paper were to reveal his first name (Philippe) and mention the first name of his wife (Sanae), then his last name (or at least, a strong candidate for his last name) can be inferred from the first hit returned by the Google query, "Philippe Sanae wedding".

In all these instances, the problem is not access control, but *inference control*. Assuming the existence of mechanisms to control access to a subset of information, the problem is to determine what information can be released publicly without compromising certain secrets, and what subset of the information cannot be released. What makes this problem difficult is the quantity and complexity of inferences that arise when published data is combined with, and interpreted against, the backdrop of public knowledge and outside data.

This paper breaks new ground in considering the problem of inference detection not in a restricted setting (such as, e.g., database tables), but in all its generality. We propose the first all-purpose approach to detecting unwanted inferences. Our approach is based on the observation that the combination of search engines and the Web, which is so well suited to detect inferences, works equally well defensively as offensively. The Web is an excellent proxy for public knowledge, since it encapsulates a large fraction of that knowledge (though certainly not all). Furthermore, the dynamic nature of the Web reflects the dynamic nature of human knowledge and means that the inferences detected today may be different from those drawn yesterday. The likelihood of certain inferences can thus be estimated automatically, at any point in time, by issuing search queries to the Web. Returning to the example of the biography redacted by the FBI, a simple search query could have flagged the risk of re-identification coming from the keywords "Saudi", "magnate" and "half-brother".

The Web is an ideal resource for identifying inferences because keyword search allows for efficient detection of the information that is associated with an individual. Such associations can be just as important in identifying someone as their personal attributes. As an example, consider the fact that the top 2 hits returned by the Google query, "pop singer vogueing"[1] have nothing to do with the singer Madonna, whereas the top 3 hits returned by the Google query, "gay pop singer vogueing"[2] all pertain to Madonna. The attribute "gay" helps to focus the results *not* because it is an attribute of Madonna (at least not as it is used in the top 3 hits) but rather it is an attribute associated with a large subset of her fanbase. Similarly, the entire first page of hits returned by the query "naltrexone acamprosate" all pertain to alcoholism, not because they are alcoholism symptoms or in some other way part of the definition of alcoholism, but rather they are associated with alcoholism because they are drugs commonly used in its treatment.

We propose generic tools for detecting unwanted inferences automatically using the Web. These tools first extract salient keywords from the private data intended for release. Then, they issue search queries for documents that match subsets of these keywords, within a reference corpus (such as the public Web) that encapsulates as much of relevant public knowledge as possible. Finally, our tools parse the documents returned by the search queries for keywords not present in the original private data. These additional keywords allow us to automatically estimate the likelihood of certain inferences. Potentially dangerous inferences are flagged for manual review. We call this new technology Web-based inference control.

We demonstrate the success of our inference detection tools with two experiments. The first experiment shows the use of our tools to automatically estimate the risk that an anonymous document allows for re-identification of its author. The second experiment shows the use of our tools to detect the risk that a document is linked to a sensitive topic. These experiments, while simple, capture the full complexity of inference detection and illustrate the power of our approach.[3]

OVERVIEW. We discuss related work in section 2. We define our models and tools, as well as our basic algorithm for Web-assisted inference detection in section 3. We list a number of potential applications of Web-assisted inference control in section 4. Section 5 describes two experiments that demonstrate the success of our inference control tools. Section 6 provides an example using Web-based inference detection to improve the redaction process. We conclude in section 7.

2 Related Work

Our work can be viewed both as a new technique for inference detection and as a new way of leveraging Web search to understand content. There is substantial existing work in both areas, but ours is the first Web-based approach to inference detection. We discuss the most closely related work in these areas below.

INFERENCE DETECTION. Most of the previous work on inference detection has focused on database content (see, for example, [33, 21, 43, 19]). Work in this area takes as input the database schema, the data themselves and,

sometimes, relations amongst the attributes of the database that are meant to model the outside knowledge a human may wield in order to infer sensitive information. To the best of our understanding, no systematic method has been demonstrated for integrating this outside knowledge into an inference detection system. Our work seeks to remedy this by demonstrating the use of the Web for this purpose. When coupled with simple keyword extraction, this general technique allows us to detect inference in a variety of unstructured documents.

A particular type of inference allows the identification of an individual. Sweeney looks for such inferences using the Web in [35] where inferences are enabled by numerical values and other attributes characterizable by regular expressions such as SSNs, account numbers and addresses. Sweeney does not consider inferences based on English language words. We use the indexing power of search engines to detect when words, taken together, are closely associated with an individual.

The closely related problem of author identification has also been extensively studied by the machine learning community (see, for example, [25, 11, 24, 34, 20]). The techniques developed generally rely on a training corpus of documents and use specific attributes like selfcitations [20] or writing style [25] to identify authors. Our work can be viewed as exploiting a previously unstudied method of author identification, using information authors reveal about themselves to identify them.

Atallah, et al. [2], describe how natural language processing can potentially be used to sanitize sensitive information when the sanitization rules are already known. Our work is focused on using the Web to identify the sanitization rules.

WEB-ASSISTED QUERY INTERPRETATION. There is a large body of work on using the Web to improve query results (see, for example, [16, 32, 10]). One of the fundamental ideas that has come out of this area is to use overlap in query results to establish a connection between distinct queries. In contrast, we analyze the content of the query results in order to detect connections between the query terms and an individual or topic.

WEB-BASED SOCIAL NETWORK ANALYSIS. Recently, the Web has been used to detect social networks (e.g., [1, 23]). A key idea in this work is using the Web to look for co-occurences of names and using this to infer a link in a social network. Our techniques can support this type of analysis, when, for example, names in a network when entered as a Web query, yield a name that is not already in the network. However, our techniques are aimed at a broader goal, that is, understanding *all* inferences that can be drawn from a document.

WEB-ASSISTED CONTENT ANALYSIS AND ANNOTATION. There is a large body of work on using the Web to understand and analyze content. Nakov and Hearst [30] have shown the power of using the Web as training data for natural language analysis. Web-assistance for extracting keywords for the purposes of content indexing and annotation is studied in [12, 37, 26]. This work is focused on automated, Web-based tools for understanding the meaning of the text as written, as opposed to the inferences that can be drawn based on the text. That said, in our work we use very simple content analysis tools, and improvements to our approach could involve more sophisticated content analysis tools including Web-based tools such as those developed in these works.

WEB-BASED DATA AGGREGATION. Finally, we note that the commercial world is beginning to offer Web-based data aggregation tools (see, for example [14, 13, 31]) for the purposes of tracking competitor behavior, doing market analysis and intelligence gathering. We are not aware of support for pre-production inference control in these offerings, as is the focus of this paper.

3 Model and Generic Algorithm

Let $\mathcal{C}$ denote a private collection of documents that is being considered for public release, and let $\mathcal{R}$ denote a collection of reference documents. For example, the collection $\mathcal{C}$ may consist of the blog entries of a writer, and the collection $\mathcal{R}$ may consist of all documents publicly available on the Web.

Let $K(\mathcal{C})$ denote all the knowledge that can be computed from the private collection $\mathcal{C}$. The set $K(\mathcal{C})$ informally represents all the statements and facts that can be logically derived from the information contained in the collection $\mathcal{C}$. The set $K(\mathcal{C})$ could in theory be computed with a complete and sound theorem prover given all the axioms in $\mathcal{C}$. In practice, such a computation is impossible and we will instead rely on approximate representations of the set $K(\mathcal{C})$. Similarly let $K(\mathcal{R})$ denote all the knowledge that can be computed from the reference collection $\mathcal{R}$.

Informally stated, the problem of inference control comes from the fact that the knowledge that can be extracted from the union of the private and reference collections $K(\mathcal{C} \cup \mathcal{R})$ is typically greater than the union $K(\mathcal{C}) \cup K(\mathcal{R})$ of what can be extracted separately from $\mathcal{C}$ and $\mathcal{R}$. The inference control problem is to understand and control the difference:

$$\text{Diff}(\mathcal{C}, \mathcal{R}) = K(\mathcal{C} \cup \mathcal{R}) - \Big(K(\mathcal{C}) \cup K(\mathcal{R})\Big).$$

Returning to the Osama Bin Laden example discussed in the introduction, consider the case where the collection $\mathcal{C}$ consists of the single declassified FBI document [8], and where $\mathcal{R}$ consists of all information publicly available on the Web. Let S denote the statement:

"The declassified FBI document is a biography of Osama Bin Laden". Since the identity of the person to whom the document pertains has been redacted, it is impossible to learn the statement S from $\mathcal{C}$ alone, and so $S \notin K(\mathcal{C})$. The statement S is clearly not in $K(\mathcal{R})$ either since it is impossible to compute from $\mathcal{R}$ alone a statement about a document that is in $\mathcal{C}$ but not in $\mathcal{R}$. It follows that S does not belong to $K(\mathcal{C}) \cup K(\mathcal{R})$. But, as shown earlier, the statement S belongs to $K(\mathcal{C} \cup \mathcal{R})$. Indeed, we learn from $\mathcal{C}$ that the document pertains to an individual characterized by the keywords "Saudi", "magnate", "half-brothers", "Yemen", etc. We learn from $\mathcal{R}$ that these keywords are closely associated with "Osama Bin Laden". If we combine these two sources of information, we learn that the statement S is true with high probability.

It is critical to understand Diff$(\mathcal{C}, \mathcal{R})$ prior to publishing the collection $\mathcal{C}$ of private documents, to ensure that the publication of $\mathcal{C}$ does not allow for unwanted inferences. The owner of $\mathcal{C}$ may choose to withhold from publication parts or all of the documents in the collection based on an assessment of the difference Diff$(\mathcal{C}, \mathcal{R})$. Sometimes, the set of sensitive knowledge K^* that should not be leaked is explicitly specified. In this case, the inference control problem consists more precisely of ensuring that the intersection Diff$(\mathcal{C}, \mathcal{R}) \cap K^*$ is empty.

3.1 Basic Approach

In this work, we consider the case in which $\mathcal{C}$ can be any arbitrary collection of documents. In particular, contrary to prior work on inference control in databases, we do not restrict ourselves to private documents formatted according to a well-defined structure. We assume that the collection $\mathcal{R}$ of public documents consists of all publicly available documents, and that the public Web serves as a good proxy for this collection. Our generic approach to inference detection is based on the following two steps:

1. UNDERSTANDING THE CONTENT OF THE DOCUMENTS IN THE PRIVATE COLLECTION $\mathcal{C}$. We employ automated content analysis in order to efficiently extract keywords that capture the content of the document in the collection $\mathcal{C}$. A wide array of NLP tools are possible for this process, ranging from simple text extraction to deep linguistic analysis. For the proof-of-concept demonstrations described in section 5, we employ keyword selection via a "term frequency - inverse document frequency" (TF.IDF) calculation, but we note that a deeper linguistic analysis may produce better results.

2. EFFICIENTLY DETERMINING THE INFERENCES THAT CAN BE DRAWN FROM THE COMBINATION OF $\mathcal{C}$ AND $\mathcal{R}$. We issue search queries for documents that match subsets of the keywords extracted in step 1, within a reference corpus (such as the public Web) that encapsulates as much of relevant public knowledge as possible. Our tools then parse the documents returned by the search queries for keywords not present in the original private data. These additional keywords allow us to automatically estimate the likelihood of certain inferences. Potentially dangerous inferences are flagged for manual review.

3.2 Inference Detection Algorithm

In this section, we give a generic description of our inference detection algorithm. This description emphasizes conceptual understanding. Specific instantiations of the inference detection algorithms, tailored to two particular applications, are given in section 5. These instantiations do not realize the full complexity of this general algorithm partly for efficiency reasons and partly because of the attributes of the application. We start with a description of the inputs, outputs and parameters of our generic algorithm.

INPUTS: A private collection of documents $\mathcal{C} = \{C_1, \ldots, C_n\}$, a collection of reference documents $\mathcal{R}$ and a list of sensitive keywords K^* that represent sensitive knowledge.

OUTPUT: A list $\mathcal{L}$ of inferences that can be drawn from the union of $\mathcal{C}$ and $\mathcal{R}$. Each inference is of the form:

$$(W_1, \ldots, W_k) \Rightarrow K_0^*,$$

where $W_1, \ldots, W_k$ are keywords extracted from documents in $\mathcal{C}$, and $K_0^* \subseteq K^*$ is a subset of sensitive keywords. The inference $(W_1, \ldots, W_k) \Rightarrow K_0^*$, indicates that the keywords $(W_1, \ldots, W_k)$, found in the collection $\mathcal{C}$, together with the knowledge present in $\mathcal{R}$ allow for inference of the sensitive keywords K_0^*. The algorithm returns an empty list if it fails to detect any sensitive inference.

PARAMETERS: The algorithm is parameterized by a value α that controls the depth of the NLP analysis of the documents in $\mathcal{C}$, by two values β and γ that control the search depth for documents in $\mathcal{R}$ that are related to $\mathcal{C}$, and finally by a value δ that controls the depth of the NLP analysis of the documents retrieved by the search algorithm. The values α, β, γ and δ are all positive integers. They can be tuned to achieve different trade-offs between the running time of the algorithm and the completeness and quality of inference detection.

UNDERSTANDING THE DOCUMENTS IN $\mathcal{C}$. Our basic algorithm uses TF.IDF (term frequency - inverse document frequency, see [28] and section 5.1) to extract from each document C_i in the collection $\mathcal{C}$ the top α keywords

that are most representative of C_i. Let S_i denote the set of the top α keywords extracted from document C_i, and let $S = \cup_{i=1}^{n} S_i$.

INFERENCE DETECTION. The list $\mathcal{L}$ of inferences is initially empty. We consider in turn every subset $\mathcal{S}' \subseteq \mathcal{S}$ of size $|\mathcal{S}'| \leq \beta$. For every such subset $\mathcal{S}' = (W_1, \ldots, W_k)$, with $k \leq \beta$, we do the following:

1. We use a search engine to retrieve from the collection $\mathcal{R}$ of reference documents the top γ documents that contain all the keywords $W_1, \ldots, W_k$.

2. With TF.IDF, we extract the top δ keywords from this collection of γ documents. Note that these keywords are extracted from the aggregate collection of γ documents (as if all these documents were concatenated into a single large document), not from each individual document.

3. Let K_0^* denote the intersection of the δ keywords from step 2 with the set K^* of sensitive keywords. If K_0^* is non-empty, we add to $\mathcal{L}$ the inference $\mathcal{C}' \Rightarrow K_0^*$.

The algorithm outputs the list $\mathcal{L}$ and terminates.

3.3 Variants of the Algorithm

The algorithm of section 3.2 can be tailored to a variety of applications. Two such applications are discussed in exhaustive detail in section 5. Here, we discuss briefly other possible variants of the basic algorithm.

DETECTING ALL INFERENCES. In some applications, the set of sensitive knowledge K^* may not be known or may not be specified. Instead, the goal is to identify all possible inferences that arise from knowledge of the collection of documents $\mathcal{C}$ and the reference collection $\mathcal{R}$. A simple variation of the algorithm given in 3.2 handles this case. In step 3 of the inference detection phase, we record all inferences instead of only inferences that involve keywords in K^*. Note that this is equivalent to assuming that the set K^* of sensitive knowledge consists of all knowledge. The algorithm may also track the number of occurrences of each inference, so that the list $\mathcal{L}$ can be sorted from most to least frequent inference.

ALTERNATIVE REPRESENTATION OF SENSITIVE KNOWLEDGE. The algorithm of section 3.2 assumes that the sensitive knowledge K^* is given as a set of keywords. Other representations of sensitive knowledge are possible. In some applications for example, sensitive knowledge may consist of a topic (e.g. alcoholism, or sexually transmitted diseases) instead of a list of keywords. To handle this case, we need a pre-processing step which converts a sensitive topic into a list of sensitive keywords. One way of doing so is to issue a search query for documents in the reference collection $\mathcal{R}$ that contain the sensitive topic, then use TF.IDF to extract from these documents an expanded set of sensitive keywords.

4 Example Applications

This section describes a wide array of potential applications for Web-based inference detection. All these applications are based on the fundamental algorithm of section 3. The first two applications are the subjects of the experiments described in detail in section 5. Experimenting with other applications will be the subject of future work.

REDACTION OF MEDICAL RECORDS. Medical records are often released to third parties such as insurance companies, research institutions or legal counsel in the case of malpractice lawsuits. State and federal legislation mandates the redaction of sensitive information from medical records prior to release. For example, all references to drugs and alcohol, mental health and HIV status must typically be redacted. This redaction task is far more complex than it may initially appear. Extensive and up-to-date knowledge of diseases and drugs is required to detect all clues and combinations of clues that may allow for inference of sensitive information. Since this medical information is readily available on public websites, the process of redacting sensitive information from medical records can be partially automated with Web-based inference control. Section 5.3 reports on our experiments with Web-based inference detection for medical redaction.

PRESERVING INDIVIDUAL ANONYMITY. Intelligence and other governmental agencies are often forced by law (such as the Freedom of Information Act) to release publicly documents that pertain to a particular individual or group of individuals. To protect the privacy of those concerned, the documents must be released in a form that does not allow for unique identification. This problem is notoriously difficult, because seemingly innocuous information may allow for unique identification, as illustrated by the poorly redacted Osama Bin Laden biography [8] discussed in the introduction. Web-based inference control is perfectly suited to the detection of indirect inferences based on publicly available data. Our tools can be used to determine how much information can be released about a person, entity or event while preserving k-anonymity, i.e. ensuring that it remains hidden in a group of like-entities of size at least k, and cannot be identified any more precisely within the group. Section 5.2 reports on our experiments with Web-based inference detection for preserving individual anonymity.

FORMULATION OF REDACTION RULES. Our Web-based inference detection tools can also be used to pre-compute a set of redaction rules that is later applied to a collection of private documents. For a large collection of private documents, pre-computing redaction rules may be more efficient than using Web-based inference detection to analyze each and every document. In 1995 for example, executive order 12958 mandated the declassification of large amounts of government data [9] (hundreds of millions of pages). Sensitive portions of documents were to be redacted prior to declassification. The redaction rules were exceedingly complex and formulating them was reportedly nearly as time-consuming as applying them. Web-based inference detection is an appealing approach to automatically expand a small set of seed redaction rules. For example, assuming that the keyword "missile" is sensitive, web-based inference detection could automatically retrieve other keywords related to missiles (e.g. "guidance system", "ballistics", "solid fuel") and add them to the redaction rule.

PUBLIC IMAGE CONTROL. This application considers the problem of verifying that a document conforms to the intentions of its author, and does not accidentally reveal private information or information that could easily be misinterpreted or understood in the wrong context. This application, unlike others, does not assume that the set of unwanted inferences is known or explicitly defined. Instead, the goal of this application is to design a broad, general-purpose tool that helps contextualize information and may draw an author's attention to a broad array of potentially unwanted inferences. For example, Web-based inference detection could alert the author of a blog to the fact that a particular posting contains a combination of keywords that will make the blog appear prominently in the results of some search query. This problem is related to other approaches to public image management, such as [13, 31]. Few technical details have been published about these other approaches, but they do not appear focused on inference detection and control.

LEAK DETECTION. This application helps a data owner avoid accidental releases of information that was not previously public. In this application of Web-based inference control, the set of sensitive knowledge K^* consists of all information that was not previously public. In other words, the release of private data should not add anything to public knowledge. This application may have helped prevent, for example, a recent incident in which Google accidentally released confidential financial information in the notes of a PowerPoint presentation distributed to financial analysts [22].

5 Experiments

Our experiments focus on exploring the first two privacy monitor applications of section 4: redaction of medical records and preserving individual anonymity. In testing these ideas, we faced two main challenges that constrained our experimental design. First, and most challenging, was designing relevant experiments that we could execute given available data. The second, more pragmatic, challenge was getting the right tools in place and executing the experiments in a time-efficient manner. We describe each of these challenges, and our approach to meeting them, in more detail below.

5.1 Experimental Design Challenges and Tools

Ideally, our idea of Web-based inference detection would be tested on authentic documents for which privacy is a chief concern. For example, a corpus of medical records being prepared for release in response to a subpoena would be ideal for evaluating the ability of our techniques to identify sensitive topics. However, such a corpus is hard to come by for obvious reasons. Similarly, a collection of anonymous blogs would be ideal for testing the ability of our techniques to identify individuals, but such blogs are hard to locate efficiently. Indeed, the excitement over the recently released AOL search data, as illustrated by the quick appearance of tools for mining the data (see, for example, [44, 4]), demonstrates the widespread difficulty in finding data appropriate for vetting data mining technologies, of which our inference detection technology is an example.[4]

Given the difficulties of finding unequivocally sensitive data on which to test our algorithms, we used instead publicly available information about an individual, which we anonymized by removing the individual's first and last names. In most cases, the public information about the individual, thus anonymized, appeared to be a decent substitute for text that the individual might have authored on their blog or Web page.

All of our experiments rely on Java code we wrote for extracting text from html, on calculation of an extended form of TF.IDF (see definition below) for identifying keywords in documents and on the Google SOAP search API [18] for making Web queries based on those keywords.

Our code for extracting text from html uses standard techniques for removing html tags. Because our experiments involved repeated extractions from similarly formatted html pages (e.g Wikipedia biographies) it was most expedient to write our own code, customized for those pages, rather than retrofitting existing text extraction code such as is available in [3].

As mentioned above, in order to determine if a word is a keyword we use the well known TF.IDF metric (see, for example, [28]). The TF.IDF "rank" of a word in a document is defined with respect to a corpus, C. We state the definition next.

Definition 1 *Let D be a document that contains the word W and is part of a corpus of documents, C. The* **term frequency** *(TF) of W with respect to D is the number of times W occurs in D. The* **document frequency** *(DF) of W with respect to the corpus, C, is the total number of documents in C that contain the keyword W. The TF.IDF value associated with W is the ratio: TF/DF.*

Our code implements a variant of TF.IDF in which we first use the British National Corpus (BNC) [27] to stem lexical tokens (e.g. the tokens "accuse", "accused", "accuses" and "accusing" would all be mapped to the stem "accuse"). We then use the BNC again to associate with each token the DF of the corresponding stem (i.e. "accuse" in the earlier example).

As with text extraction from html, there are open source (and commercial) offerings for calculating TF.IDF based on a reference corpus. We did not, however, have a reference corpus on which to base our calculations, and thus opted to write our own code to compute TF.IDF based on the DF values reported in the BNC (which is an excellent model for the English language as a whole, and thus presumably also for text found on the Web).

Our final challenge was experimental run-time. Although we did not invest time optimizing our text extraction code for speed it nevertheless proved remarkably efficient in comparison with the time needed to execute Google queries and download Web pages. In addition, Google states that they place a constraint of $1,000$ queries per day for each registered developer on the Google SOAP Search API service [18]. This constraint required us to amass enough Google registrations in order to ensure our experiments could run uninterrupted; in our case, given the varying running times of our experiments, 17 registrations proved enough. The delay caused by query execution and Web page download caused us to modify our algorithms to do a less thorough search for inferences than we had originally intended. These modifications almost certainly cause our algorithms to generate an incomplete set of inferences. However, it is also important to note that despite our efforts, our results contain some links that should have been discarded because they either don't represent new information (e.g. scrapes of the site from which we extracted keywords) or don't connect the keywords in our query to the sensitive words in a meaningful way (e.g. an online dictionary covering a broad swath of the English language). Hence, it is possible to improve upon our results by changing the parameters of our basic experiments to either do more filtering of the query results or analyze more of the query results and require a majority contain the sensitive word(s).

We describe each experiment in detail below.

5.2 Web-based De-anonymization

As discussed in section 4 one of our goals is to demonstrate how keyword extraction can be used to warn the end-user of impending identification. Our inference detection technology accomplishes this by constantly amassing keywords from online content proposed for posting by the user (e.g. blog entries) and issuing Web queries based on those keywords. The user is alerted when the hits returned by those queries return their name, and thus is warned about the risk of posting the content.

We simulated this setting with Wikipedia biographies standing in for user-authored content. We removed the biography subject's name from the biography and viewed the personal content in the biography as being a condensed version of the information an individual might reveal over many posts to their blog, for example. From these "anonymized" biographies we extracted keywords. Subsets of keywords formed queries to Google. A portion of the returned hits were then searched for the biography subject's name and a flag was raised when a hit that was not a Wikipedia page contained a mention of the biography's subject. For efficiency reasons, we limited the portion and number of Web pages that were examined. In more detail, our experiment consists of the following steps:

Input: a Wikipedia biography, B:

1. Extract the subject, N, of the biography, B, and parse N into a first name, N_1, optional middle name or middle initial, N_1', and a last name, N_2 (where N_j is empty if a name in that position is not given in the biography).[5]
2. Extract the top 20 keywords from the Wikipedia biography, B, forming the set, S_B, through the following steps:
   (a) Extract the text from the html.
   (b) Calculate the enhanced TF.IDF ranking of each word in the extracted text (section 5.1). If present, remove N_1, N_1' and N_2 from this list, and select the top 20 words from the remaining text as the ordered set, S_B.
3. For $x = 20, 19, \ldots, 1$, issue a Google query on the top x keywords in S_B. Denote this query by Q_x. For example, if W_1, W_2, W_3 are the top 3 keywords, the Google query Q_3 is: $W_1 \quad W_2 \quad W_3$, with no additional punctuation. Let $\mathcal{H}_x$ be the set of hits

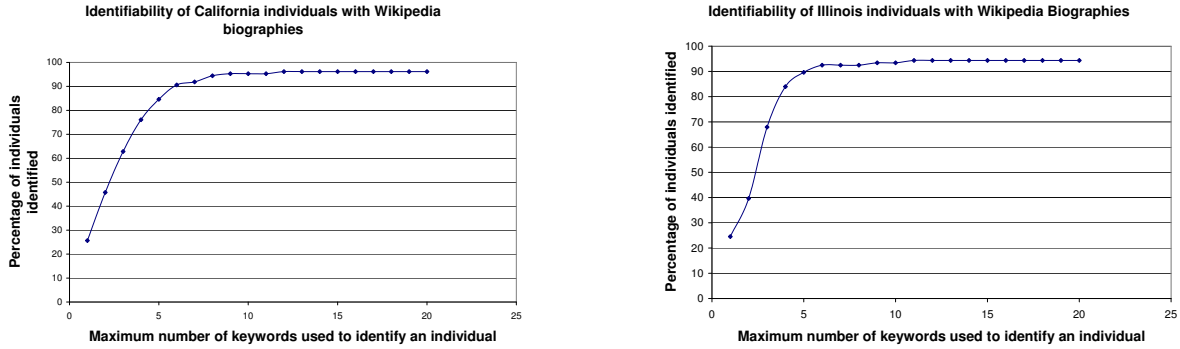

Figure 1: Using 20 keywords per person, extracted from each resident's Wikipedia biography, the percentage of individuals who were identifiable based on x keywords or less for $x = 1, \ldots, 20$. The graph on the left shows results for the 234 biographies of California residents in Wikipedia and the graph on the right shows the results for the 106 biographies of Illinois residents in Wikipedia.

returned by issuing query Q_x to Google with the restrictions that the hits consist solely of html or text[6] and that no hits from the en.wikipedia.org Web site be returned.

4. Let $H_{x,1}, H_{x,2}, H_{x,3} \in \mathcal{H}_x$ be the first, second and third hits (respectively) resulting from query Q_x.[7] For $x = 20, 19, \ldots, 1$, determine if $H_{x,1}$, $H_{x,2}$ and $H_{x,3}$ contain references to subject, N, by searching for contiguous occurrences of N_1, N_1' and N_2 (meaning, no words appear in between the words in a name) within the first 5000 lines of html in each of $H_{x,1}$, $H_{x,2}$ and $H_{x,3}$. Record any such occurrences.

Output: S_B, each query Q_x that contains N_1, N_1' and N_2 contiguously in at least one of the three examined hits, and the url of the particular hit(s).

We ran this test on the 234 biographies of California residents, and the 106 biographies of Illinois residents contained in Wikipedia. The results for both states are shown in Figure 1 and are very similar. In each case, 10 or fewer keywords (extracted from the Wikipedia biography) suffice to identify almost all the individuals. Note that statistics in Figure 1 are based solely on the output of the code, with no human review.

We also include example results (keywords, url, biography subject) in Figure 2. These results illustrate that the associations a person has may be as useful for identifying them as their personal attributes. To highlight one example from the figure, 50% of the first page of hits returned from the Google query "nfl nicole goldman francisco pro" are about O. J. Simpson (including the top 3 hits), but there is no reference to O. J. Simpson in any of the first page of hits returned by the query "nfl francisco pro". Hence, the association of O. J. Simpson with his wife (Nicole) and his wife's boyfriend (Goldman) is very useful to identifying him in the pool of professional football players who once were members of the San Francisco 49ers.

PERFORMANCE. In our initial studies, there was wide variation, from a few minutes to over an hour, in the total time it took to process a single biography, B, depending on the length of the Web pages returned and the number of hits. Hence, in order to efficiently process a sufficiently large number of biographies we restricted the code to only examining the first 5000 lines of html in the returned hits from a given query, and to only search the first 3 hits returned from any given query. With these restrictions, each biography took around 20 minutes to process, with some variation due to differences in biography length. In total, our California experiments took around 78 hours and our Illinois experiments took about 35 hours. Our experimental code does not keep track of the number of queries issued per registration and doing so may yield better performance because switching between registrations occurred only upon receiving a Google SOAP error and so caused some delay.

Our code was not optimized for performance and improvements are certainly possible. In particular, our main slow down came from the text extraction step. One improvement would be to cache Web sites to avoid repeat extractions.

Keywords	URL of Top Hit	Name of Person
campaigned soviets	http://www.utexas.edu/features/archive/2004/election_policy.html	Ronald Reagan
defense contra reagan	http://www.pbs.org/wgbh/amex/reagan/peopleevents/pande08.html	Caspar Weinberger
reagan attorney edit pornography	http://www.sourcewatch.org/index.php?title=Edwin_Meese_III	Edwin Meese
nfl nicole goldman francisco pro	http://www.brainyhistory.com/years/1997.html	O. J. Simpson
kung fu actors	http://www.amazon.com/Kung-Fu-Complete-Second-Season/dp/B0006BAWYM	David Carradine
medals medal raid honor aviation	http://www.voicenet.com/~lpadilla/pearl.html	Jimmy Doolittle
fables chicago indiana	http://www.indianahistory.org/pop_hist/people/ade.html	George Ade
wisconsin illinois chicago architect designed	http://www.greatbuildings.com/architects/Frank_Lloyd_Wright.html	Frank Lloyd Wright

Figure 2: Excerpts from our de-anonymization experiments. Each row lists keywords extracted from the Wikipedia biography of an individual (categorized under "California" or "Illinois"), a hit returned by a Google query on those keywords that is one of the top three hits returned and contains the individual's name, and the name of the individual.

5.3 Web-based Sensitive Topic Detection

Another application of Web-based inference detection is the redaction of medical records. As discussed earlier, it is common practice to redact all information about diseases such as HIV/Aids, mental illness, and drug and alcohol abuse, prior to releasing medical records to a third party (such as, e.g., a judge in malpractice litigation). Implementing such protections today relies on the thoroughness of the redaction practitioner to keep abreast of all the medications, physician names, diagnoses and symptoms that might be associated with such conditions and practices. Web-based inference detection can be used to improve the thoroughness of this task by automating the process of identifying the keywords allowing such conditions to be inferred.

To demonstrate how our algorithm can be used in this application, our experiments take as input a page that is viewed as authoritative about a certain disease. In our experiments, we used Wikipedia to supply pages for alcoholism and sexually transmitted diseases (STDs). The text is then extracted from the html, and keywords are identified. To identify keywords that might allow the associated disease to be inferred we then issued Google queries on subsets of keywords and examined the top hit for references to the associated disease. In general, we counted as a reference any mention of the associated disease. The one exception to this rule is that we filtered out some medical term sites since such sites list unrelated medical terms together (for indexing purposes) and we didn't want such lists to trigger inference results.

In the event that such a reference was found we recorded those keywords as being potentially inference-enabling. In practice, a redaction practitioner might then use this output to decide what words to redact from medical records before they are released in order to preserve the privacy of the patient.

To gain some confidence in our approach we also used a collection of general medical terms as a "control" and followed the same algorithm. That is, we made Google queries using these medical terms and looked for references to a sensitive disease (STDs and alcoholism) in the returned links. The purpose of this process was to see if the results would differ from those obtained with keywords from the Wikipedia pages about STDs and alcoholism. We expected a distinct difference because the Wikipedia pages should yield keywords more relevant to STDs and alcoholism, and indeed the results indicate that is the case.

The following describes our experiment in more detail.

1. *Input*: An ordered set of sensitive words, $K^* = \{v_1, \ldots, v_b\}$, for some positive integer b, and a page, B. B is either the Wikipedia page for alcoholism [40], the Wikipedia page for sexually transmitted diseases (STDs) [41] or a "control" page of general medical terms.
   (a) If B is a Wikipedia page, extract the top 30 keywords from B, forming the set S_B, through the following steps:
      i. Extract text from html.
      ii. Calculate the enhanced TF.IDF ranking of each word in the extracted text (section 3). Select the top 30 words as the ordered set, $S_B = \{W_1, W_2, \ldots, W_{30}\}$.
   (b) If B is a medical terms page, extract the terms using code customized for that Web site and

let $W_B = \{W_1, W_2, \ldots, W_{30}\}$ be a subset of 30 terms from that list, where the selection method varies with each run of the experiment (see the results discussion below for the specifics).

(c) For each pair of words $\{W_i, W_j\} \in S_B$, let $Q_{i,j}$ be the query consisting of just those two words with no additional punctuation and the restriction that no pages from the domain of source page B be returned, and that all returned pages be text or html (to avoid parsing difficulties). Let $H_{i,j}$ denote the first hit returned after issuing query $Q_{i,j}$ to Google, after known medical terms Web sites were removed from the Google results[8].

(d) For all $i, j \in \{1, \ldots, 30\}$, $i \neq j$, and for $\ell \neq \{1, \ldots, b\}$, search for the string $v_\ell \in K^*$ in the first 5000 lines of $H_{i,j}$. If v_ℓ is found, record v_ℓ, w_i, w_j and $H_{i,j}$ and discontinue the search.

2. *Output*: All triples $(v_\ell, Q_{i,j}, H_{i,j})$ found in step 1, where v_ℓ is in the first 5000 lines of $H_{i,j}$.

RESULTS FOR STD EXPERIMENTS. We ran the above test on the Wikipedia page about STDs [41], B, and a selected set, B', of 30 keywords from the medical term index [29]. The set B' was selected by starting at the 49^{th} entry in the medical term index and selecting every 400^{th} word in order to approximate a random selection of medical terms. As expected, keyword pairs from input B generated far more hits for STDs ($306/435 > 70\%$) than keyword pairs from B' ($108/435 < 25\%$). The results are summarized in figure 3.

RESULTS FOR ALCOHOLISM EXPERIMENTS. We ran the above test on the Wikipedia page about alcoholism [40], B, and a selected set, B', of 30 keywords from the medical term index [29]. For the run analyzed in Figure 4, the set B' was selected by starting at the 52^{nd} entry in the medical term index and selecting every 100^{th} word until 30 were accumulated in order to approximate a random selection of medical terms. As expected, keyword pairs from input B generated far more hits for alcoholism (47.82%) than B' (9.43%). In addition, we manually reviewed the URLs that yielded a hit in $v \in K^*_{Alc}$ for a seemingly innocuous pair of keywords. These results are summarized in figure 4.

APPLYING THE RESULTS. When redacting medical records, a redaction practitioner might use the results in figures 3 and 4 to choose content to redact. For example, figure 4 indicates the medications naltrexone and acamprosate should be removed due to their popularity as alcoholism treatments. The words identified as STD-inference enabling are far more ambiguous (e.g. "transmit", "infected"). However, for some individuals the very fact that even general terms are frequently associated with sensitive diseases may be enough to justify redaction (e.g. a politician may desire the removal of any "red flag" words). In general though, we think a redaction practitioner could defensibly *not* make it a practice to redact such general terms given their association with other, less sensitive, diseases. This emphasizes that our techniques support semi-automation, but not full automation, of the redaction process.

PERFORMANCE. Amortizing the cost of text extraction from the Wikipedia source page over all the queries, determining if each keyword pair yielded a top hit containing a sensitive word took approximately 150 seconds. Hence, each of the experiments in figures 3 and 4 took around 6 hours, since 435 pairs from the Wikipedia page were tested along with 435 pairs from the "control" set of keywords.

As in the de-anonymization experiments, our main time cost was due to the process of text extraction from html. For these experiments caching is likely to significantly improve performance as many of the medical resource sites were visited multiple times.

6 Use Scenario: Iterative Redaction

As mentioned in sections 1 and 4, the process of sanitizing documents by removing obviously identifying information like names and social security numbers can be improved by using Web-based inference detection to identify pieces of seemingly innocuous information that can be used to make sensitive inference. To illustrate this idea, we return to the poorly redacted FBI document in the left-hand side of figure 6. Algorithms like those presented in sections 3.2 and 5 can be used to identify sets of keywords that allow for undesired inferences. Some or all of those keywords can then be redacted to improve the sanitization process.

We emphasize that the strategy for redacting based upon the inferences detected by our algorithms is a research problem that is not addressed by this paper. Indeed many strategies are possible. For example, one might redact the minimum set of words (in which case, the redactor seeks to find a minimum set cover for the collection of sets output by the inference detection algorithm). Alternatively, the redactor might be biased in favor of redacting certain parts of speech (e.g. nouns rather than verbs) to enhance readability of the redacted document.

The type of redaction strategy that is employed may influence the Web-based inference detection algorithm. For example, if the goal is to redact the minimum set of words, then it is necessary to consider all possible sets

Summary of STD Experiments

Input Web Page, B: Wikipedia STD site [41]

Extracted Keywords, S_B: transmit, sexually, transmitting, transmitted, infection, std, sti, hepatitis, infected, infections, transmission, stis, herpes, viruses, virus, chlamydia,_[9], stds, sexual, disease, hiv, membrane, genital, intercourse, diseases, pmid, hpv, mucous, viral, 2006

Input Web Page, B'**, ("control" page)**: Medical Terms Site [29]

Extracted "Control" Keywords, S'_B: Ablation, Ah-Al, Aneurysm, thoracic, Arteria femoralis, Barosinusitis, Bone mineral density, Cancer, larynx, Chain-termination codon, Cockayne syndrome, Cranial nerve IX, Dengue, Disorder, cephalic, ECT, Errors of metabolism, inborn, Fear of nudity, Fracture, comminuted, Gland, thymus, Hecht-Beals syndrome, Hormone, thyroxine, Immunocompetent, Iris melanoma, Laparoscopic, Lung reduction surgery, Medication, clot-dissolving, Mohs surgery, Nasogastric tube, Normoxia, Osteosarcoma, PCR (polymerase chain reaction), Plan B

Sensitive Keywords, K^*_{STD}: STD, Chancroid, Chlamydia, Donovanosis, Gonorrhea, Lymphogranuloma venereum, Non-gonococcal urethritis, Syphilis, Cytomegalovirus, Hepatitis B, Herpes, HSV, Human Immunodeficiency Virus, HIV, Human papillomavirus, HPV, genital warts, Molluscum, Severe acute respiratory syndrome, SARS, Pubic lice, Scabies, crabs, Trichomoniasis, yeast infection, bacterial vaginosis, trichomonas, mites, nongonococcal urethritis, NGU, molluscum contagiosm virus, MCV, Herpes Simplex Virus, Acquired immunodeficiency syndrome, aids, pubic lice, HTLV, trichomonas, amebiasis, Bacterial Vaginosis, Campylobacter Fetus, Candidiasis, Condyloma Acuminata, Enteric Infections, Genital Mycoplasmas, Genital Warts, Giardiasis, Granuloma Inguinale, Pediculosis Pubis, Salmonella, Shingellosis, vaginitis

Percentage of words in S_B **yielding a top hit containing word(s) in** K^*_{STD}: 33.33%

Percentage of word pairs in S_B **yielding a top hit containing word(s) in** K^*_{STD}: 70.34%

Percentage of "control" words in S'_B **yielding a top hit containing word(s) in** K^*_{STD}: 3.33%

Percentage of "control" word pairs in S'_B **yielding a top hit containing word(s) in** K^*_{STD}: 24.83%

Example keyword pairs from S_B **returning a top hit containing a word in** K^*_{STD}:[10]

Keywords	URL of Top Hit	Sensitive Word in Top Hit
transmit, infected	http://www.rci.rutgers.edu/ insects/aids.htm	HIV
transmit, mucous	http://research.uiowa.edu/animal/?get=empheal	Herpes
transmitting, viruses	http://www.cdc.gov/hiv/resources/factsheets/transmission.htm	Hepatitis B
transmitted, viral	http://www.eurosurveillance.org/em/v10n02/1002-226.asp	Hepatitis B
transmitted, infection	http://www.plannedparenthood.org/sti/	STD
transmitted, disease	http://www.epigee.org/guide/stds.html	STD
infection, mucous	http://www.niaid.nih.gov/factsheets/sinusitis.htm	HIV
infected, disease	http://www.ama-assn.org/ama/pub/category/1797.html	HIV
infected, viral	http://www.merck.com/mmhe/sec17/ch198/ch198a.html	Cytomegalovirus
infections, viral	http://www.nlm.nih.gov/medlineplus/viralinfections.html	Cytomegalovirus
virus, disease	http://www.mic.ki.se/Diseases/C02.html	Cytomegalovirus

Figure 3: Summary of experiments to identify keywords enabling STD inferences.

Summary of Alcoholism Experiments

Input Web Page, B: Wikipedia Alcoholism site [40]

Extracted Keywords, S_B: alcoholism, alcohol, drunk, alcoholic, alcoholics, naltrexone, drink, addiction, dependence, detoxification, diagnosed, screening, drinks, moderation, abstinence, 2006, disorder, drinking, behavior, questionnaire, cage, treatment, citation , acamprosate, because, pharmacological, anonymous, extinction, sobriety, dsm

Input Web Page, B'**, ("control" page)**: Medical Terms Site [29]

Extracted "Control" Keywords, S'_B: ABO blood group, Alarm clock headache, Ankle-foot orthosis, Ascending aorta, Benign lymphoreticulosis, Breast bone, Carotid-artery stenosis, Chondromalacia patellae, Congenital, Cystic periventricular leukomalacia, Discharge, DX, Enterococcus, Familial Parkinson disease type 5, Fondation Jean Dausset-CEPH, Giant cell pneumonia, Heart attack, Hormone, parathyroid, Impetigo, Itching, Laughing gas, M. intercellulare, Membranous nephropathy, MRSA, Nerve palsy, laryngeal, Oligodendrocyte, Pap Smear, Phagocytosis, Postoperative, Purpura, Henoch-Schonlein

Sensitive Keywords, K^*_{Alc}: alcoholism, alcoholic(s), alcohol

Percentage of words in S_B **yielding a top hit containing word(s) in** K^*_{Alc}: 23.33%

Percentage of word pairs in S_B **yielding a top hit containing word(s) in** K^*_{Alc}: 47.82%

Percentage of "control" words in S'_B **yielding a top hit containing word(s) in** K^*_{Alc}: 0.00%

Percentage of "control" word pairs in S'_B **yielding a top hit containing word(s) in** K^*_{Alc}: 9.43%

Example word sets from S_B **returning a top hit containing a word in** K^*_{Alc}:[11]

Keywords	URL of Top Hit
naltrexone	http://www.nlm.nih.gov/medlineplus/druginfo /medmaster/a685041.html
acamprosate	http://www.nlm.nih.gov/medlineplus/druginfo /medmaster/a604028.html
dsm, detoxification	http://www.aafp.org/afp/20050201/495.html[12]
dsm, detoxification, dependence	http://www.aafp.org/afp/20050201/495.html

Figure 4: Summary of experiments to identify keywords enabling alcoholism inferences.

Redacted Word(s)	Example Link	Sensitivity of Word(s)
50, 52, 54	http://multimedia.belointeractive.com/attack /binladen/1004blfamily.html	Having 50 or more siblings is very characteristic of Osama Bin Laden.
Boston	http://www.time.com/time/magazine/article /0,9171,1000943,00.html?promoid=googlep	Many of Osama's relatives reside in Boston.[13]
magnate	http://www.outpostoffreedom.com/bin_ladin.htm	Osama's father was a building magnate.
denounced, denunciation	http://www.cairnet.org/html/911statements.html	A number of groups (including Bin Laden's family) have denounced his actions.
condemnation	http://www.usnews.com/usnews/politics/whispers /archive/september2001.htm	A number of groups (including Bin Laden's family) have condemned his actions.

Figure 5: Words redacted as a result of Web-based inference detection. Column 1 is the word or words, column 2 is a link using those words output by the algorithm, and column 3 explains why the word(s) are sensitive.

of keywords when looking for inferences. In contrast, if readability is an important concern, then the considered sets might be those favoring certain word types.

What we discuss here is one example of using Web-based inference detection to improve the redaction process. The approach we take is influenced by readability and performance (i.e. speed of the redaction process) but is by no means an optimal approach with respect to either concern. We began by applying some simple redaction rules to the document [8]. Specifically, we removed all location references since our example in section 1 indicated those were important to identifying the biography subject, any dates near September 11, 2001, which is clearly a memorable date, and finally, all citation titles since when paired with the associated publication, these enable the citation articles to be easily retrieved. The resulting redacted document is depicted in figure 6, where grey rectangles indicate the redaction resulting from the rules just described.

Our subsequent redaction proceeded iteratively. At each stage, we extracted the text from the current document, calculated the keywords ordered by the TF.IDF metric and searched for inferences drawn from subsets of a specified number of the top keywords. We then evaluated the output of the algorithm by checking to ensure the produced links did indeed reflect identifying inferences. If a link did not use all the queried keywords in a discussion about Osama Bin Laden then it was deemed invalid. A common source of invalid links were news article titles printed in the side-bar of the link that did not make use of the keywords found in the main body. For example, the query "condone citing prestigious", yields the top hit [6] (a humor site) because a sidebar links to an article with "Osama" in the title, however, none of the keywords are used in the description of that article.[14]

We incorporated manual review of the links because the current form of our algorithms involves too little content analysis to provide confidence that a returned link reflects a strong connection between the associated keywords and Osama Bin Laden. In addition, given the high security nature of most redaction settings it is unlikely that a purely automated process will ever be accepted.

For those inferences that were found valid, we made redactions to prevent such inferences and repeated this process for the newly redacted document. The following makes the steps we followed precise.

1. Dates near September 11, 2001, titles of all citations and location names were removed from the biography [8].
2. For $i = 2, \ldots, 5$:
   (a) We executed Google queries for each i-tuple in the top n_i keywords in the biography. The n_i values were chosen based on performance constraints as described in section 5.[15] The (i, n_i) values were: $(2, 50), (3, 20), (4, 15)$ and $(5, 13)$. We concluded with 5-tuples because no valid inferences were found for that run of the algorithm, and only 7% of the links returned by the algorithm run for $(i, n_i) = (4, 15)$ were valid. For each (i, n_i) execution of the algorithm we received a list of sets of keywords that were potentially inference-enabling, and the associated top link leading the algorithm to make this conclusion.
   (b) We reviewed the returned links to see if all the corresponding keywords were used in a discussion of Osama Bin Laden. If so, we made a judgement as to which keyword or keywords to remove to remove the inference while preserving readability of the document.
   (c) We incremented i and returned to step (a) with the current form of the redacted document.

Figure 5 lists the words that were redacted as a result of our Web-based inference detection algorithm. The table also gives an example link output by the algorithm that motivates the redaction and a brief explanation of why the word is sensitive (gained from the manual review of the link(s)). Note that while our algorithm found some document features to be identifying that are unlikely to have been covered by a generic redaction rule (e.g. Osama Bin Laden's father's attribute of being a building magnate) it left other, seemingly unusual, attributes (such as Osama Bin Laden potentially being one of 20 children). Since the Web is at best a *proxy* for human knowledge, and our algorithm used the Web in a limited way (i.e. our analysis was limited to a few hits with little NLP use), it seems likely that inferences were missed. Hence, we emphasize that our tool is best used to semi-automate the redaction process.

Finally, we note that the act of redacting information may introduce as well as remove, privacy problems. For example, as noted by Vern Paxson [39], redacting "Boston" without redacting "Globe" may allow the sensitive term "Boston" to be inferred. Our tool suggests "Boston" for redaction, as opposed to "Boston Globe", because a number of Osama Bin Laden's relatives reside there, however, acting on this recommendation is problematic precisely because of the difference between the nature of the inference and the document usage of the term. An improved algorithm would understand the use of the term within the document and use this to guide the redaction process.

Our final redacted document is shown in the right hand side of figure 6.

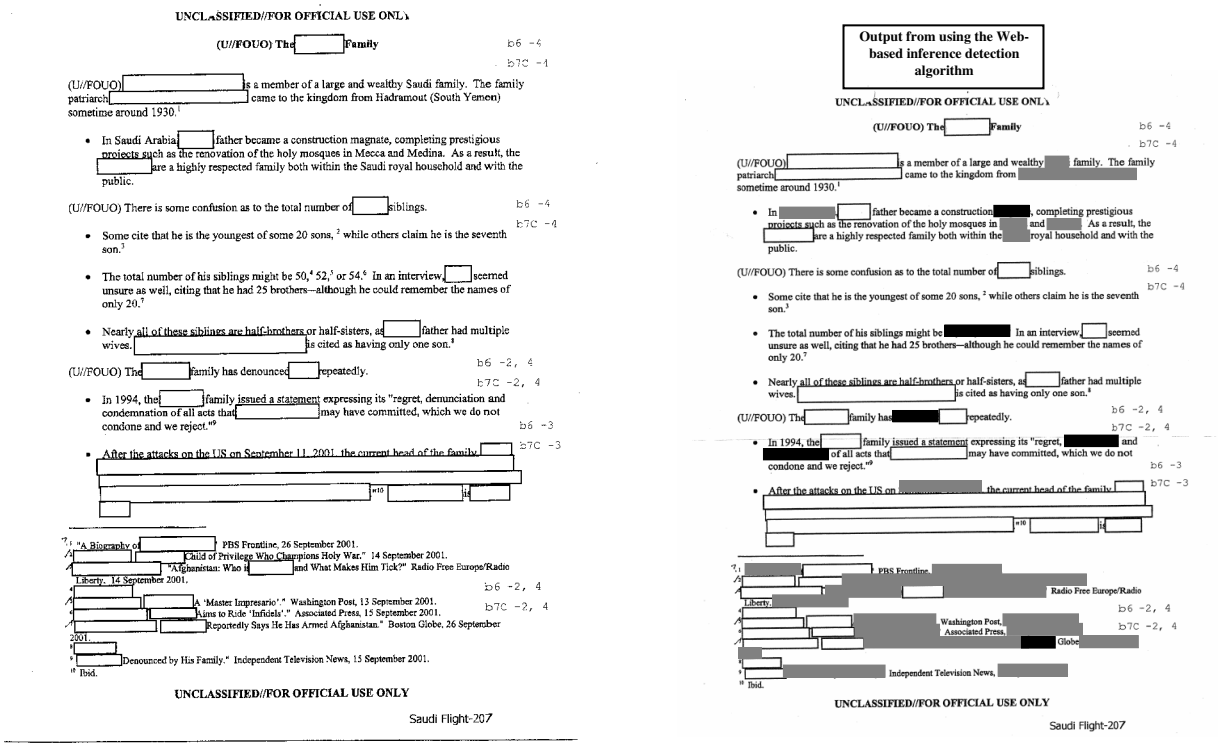

Figure 6: The left picture shows the original FBI-redacted biography. The right hand side shows the document resulting from using the Web-based inference detection algorithm, where black rectangles represent redactions recommended by the algorithm and grey rectangles are redactions coming from removing dates in 2001, locations and the titles of cited articles (i.e. the grey and black rectangles are redactions made by the authors of this paper).

7 Conclusion

We have introduced the notion of using the Web to detect undesired inferences. Our proof-of-concept experiments demonstrate the power of the Web for finding the keywords that are likely to identify a person or topic.

As is to be expected with an initial work, there remains a lot of room for improvement in the algorithms. In particular, to produce an inference detection tool capable of functioning in real-time, as is needed in some applications, improvements already discussed such as Web caching, additional filtering of results to improve precision, and deeper hit analysis to improve recall, are needed. Another avenue for improvement is through deeper content analysis (i.e. beyond keyword extraction). For example, employing a tool capable of deeper semantic analysis such as [15] may allow for both more meaningful extraction of words and phrases for generating queries, and improved analysis of the returned hits for more accurate inference detection. In addition, simple improvements to the content analysis such as better filtering of stop words and html syntax, would create more useful keyword lists.

Acknowledgement

The authors are very grateful to Richard Chow and Vern Paxson for their help in revising earlier versions of this paper.

References

[1] B. Aleman-Meza, M. Nagarajan, C. Ramakrishnan, L. Ding, P. Kolari, A. Sheth, B. Arpinar, A. Joshi and T. Finin. Semantic analytics on social networks: experiences in addressing the problem of conflict of interest detection. *15th International World Wide Web Conference*, 2006.

[2] M. Atallah, C. McDonough, S. Nirenburg, and V. Raskin. Natural Language Processing for Information Assurance. *Proc. 9th ACM/SIGSAC New Security Paradigms Workshop (NSPW 00)*, pp.51-65, 2000.

[3] Apache Lucene. `http://lucene.apache.org/java/docs/`

[4] AOL Keyword Searches. `http://dontdelete.com/default.asp`

[5] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. *The New York Times*, August 9, 2006.

[6] `http://www.bongonews.com/layout1.php?event=2315`

[7] W. Broad. U. S. Web Archive is Said to Reveal a Nuclear Primer. *The New York Times*, November 3, 2006.

[8] `http://www.judicialwatch.org/archive/2005/osama.pdf`

[9] Executive Order 12958, Classified National Security Information. `http://www.dss.mil/seclib/eo12958.htm`

[10] B. Davison, D. Deschenes and D. Lewanda. Finding relevant website queries. *Twelfth International World Wide Web Conference*, 2003.

[11] O. de Vel, A. Anderson, M. Corney and G. Mohay. Mining email content for author identification forensics. *SIGMOD Record*, Vol. 30, No. 4, December 2001.

[12] Mike Dowman, Valentin Tablan, Hamish Cunningham and Borislav Popov. Web-Assisted Annotation, Semantic Indexing and Search of Television and Radio News. *WWW*, 2005.

[13] Factiva Insight: Reputation Intelligence. `http://www.factiva.com`

[14] Fetch Technologies. `http://www.fetch.com`

[15] GATE: General Architecture for Text Engineering. http://gate.ac.uk/projects.html

[16] N. Glance. Community Search Assistant. *IUI*, 2001.

[17] P. Golle. Revisiting the Uniqueness of Simple Demographics in the US Population. *Workshop on Privacy in the Electronic Society*, 2006.

[18] Google SOAP search API. `http://code.google.com/apis/soapsearch/`

[19] J. Hale and S. Shenoi. Catalytic inference analysis: detecting inference threats due to knowledge discovery. *IEEE Symposium on Security and Privacy*, 1997.

[20] S. Hill and F. Provost. The myth of the double-blind review? Author identification using only citations. *SIGKDD Explorations*, 2003.

[21] T. Hinke. Database inference engine design approach. *Database Security II: Status and Prospects*, 1990.

[22] D. Jones. Google's PowerPoint blunder was preventable. IR Web Report. `http://www.irwebreport.com/perspectives/2006/mar/google_blunder.htm`

[23] E. Kin, Y. Matsuo, M. Ishizuka. Extracting a social network among entities by web mining. *ISWC '06 Workshop on Web Content Mining with Human Language Technologies*, 2006.

[24] M. Koppel and J. Schler. Authorship verification as a one-class classification problem. *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[25] M. Koppel, J. Schler, S. Argamon and E. Messeri. Authorship attribution with thousands of candidate authors. *SIGIR '06*.

[26] M. Lapata and F. Keller. The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks, *HLT-NAACL*, 2004.

[27] G. Leech, P. Rayson and A. Wilson. *Word frequencies in written and spoken english: based on the British National Corpus.*, Longman, London, 2001.

[28] C. Manning and H. Schutze. Foundations of statistical natural language processing. MIT Press, 1999.

[29] MedicineNet.com. `http://www.medterms.com/script/main/hp.asp`

[30] P. Nakov and M. Hearst. Using the Web as an Implicit Training Set: Application to Structural Ambiguity Resolution. *HLT-NAACL*, 2005.

[31] Nstein Technologies. `http://www.nstein.com/pim.asp`

[32] G. Pant, S. Bradshaw and F. Menczer. Search engine-crawler symbiosis: adapting to community interests.*7th European Conference on Digital Libraries*, 2003.

[33] X. Qian, M. Stickel, P. Karp, T. Lunt and T. Garvey. Detection and elimination of inference channels in multilevel relational database systems. *IEEE Symposium on Security and Privacy*, 1993.

[34] M. Steyvers, P. Smyth, M. Rosen-Zvi and T. Griffiths. Probabilistic author-topic models for information discovery. *KDD '04*.

[35] L. Sweeney. AI Technologies to Defeat Identity Theft Vulnerabilities. *AAAI Spring Symposium on AI TEchnologies for Homeland Security, 2005*.

[36] L. Sweeney. Uniqueness of Simple Demographics in the U.S. Population. *LIDAP-WP4. Carnegie Mellon University, Laboratory for International Data Privacy, Pittsburgh, PA*, 2000.

[37] P. Turney. Coherent Keyphrase Extraction via Web Mining. *IJCAI*, 2002.

[38] Unified Medical Language System. `http://www.umm.edu/glossary/a/index.html`

[39] Personal communication.

[40] Wikipedia. Alcoholism. `http://en.wikipedia.org/wiki/Alcoholism`

[41] Wikipedia. Sexually transmitted disease. `http://en.wikipedia.org/wiki/Sexually_transmitted_disease`

[42] `http://wordweb.info/free/`

[43] R. Yip and K. Levitt. Data level inference detection in database systems. *IEEE Eleventh Computer Security Foundations Workshop*, 1998.

[44] D. Zhao and T. Sapp. AOL Search Database. `http://www.aolsearchdatabase.com/`

[45] `http://www.zoominfo.com/`

Notes

[1] http://www.popandpolitics.com/2005/09/06/and-lite-jazz-singers-shall-lead-the-way/, www.popandpolitics.com/2006/10/06/our-paris/

[2] http://en.wikipedia.org/wiki/Madonna_and_the_gay_community, http://gaybookreviews.info/review/2807/615, http://www.youtube.com/results?search_type=related&search_query=madonna%20oh%20father

[3] Example results from our experiments appear in section 5. Because of the dynamic nature of the Web, issuing the same queries today may yield somewhat different results.

[4] The AOL data can potentially be used to demonstrate the Web's ability to de-anonymize ([5] may be one such example), which is one of the goals of our algorithms, however because our target application is the protection of English language content, we opted not to vet our algorithms with that data.

[5] The vast majority of the biographies we used identified their subject by both a first and last name with no middle name or initial. Also, name suffixes (e.g. Jr. or annotations made by Wikipedia authors regarding profession), were ignored.

[6] This was done to avoid difficulties parsing non-ascii pages.

[7] These are the first three links that appear on the results page, whether or not one URL is a substring of another.

[8] Here "known site" means any site with "medterm" or "medword" in the URL. As this certainly not sufficient to remove all medical terms sites, we manually reviewed the results before generating the example keyword pairs in Figure 3.

[9] Note this extracted non-word indicates a flaw in our text-from-html extraction algorithm.

[10] In a manual review of the word pairs from W'_B yielding a top hit containing word(s) in K^*_{STD}, we did not find any hits using the word pair in a meaningful way in relation to a sensitive word. Rather, the hits generally turned out to be medical term lists.

[11] Since all of our sensitive words pertain to the same topic, alcoholism, we did not record which particular sensitive word was contained in the top hit (if any).

[12] Note this is the 4^{th} returned hit, indicating a change in our search strategy would improve recall.

[13] The biography only mentions "Boston" in a citation, so this is a conservative redaction choice.

[14] Alternative metrics for validity are of course possible. For example, a more thorough algorithms might look for shared topic (e.g. the events of September 11, 2001) amongst links, and retain any links pertaining to the most popular topic as valid.

[15] We tended to experience problems communicating with Google when when executing algorithm runs that exceeded 1500 queries, hence we chose values of $\{n_i\}_i$ that yielded query counts in the range of $1000 - 1500$.