Detecting Privacy Leaks Using Corpus-based Association Rules

Richard Chow
Palo Alto Research Center
Palo Alto, CA 94304, USA
rchow@parc.com

Philippe Golle
Palo Alto Research Center
Palo Alto, CA 94304, USA
pgolle@parc.com

Jessica Staddon
Palo Alto Research Center
Palo Alto, CA 94304, USA
staddon@parc.com

ABSTRACT

Detecting inferences in documents is critical for ensuring privacy when sharing information. In this paper, we propose a refined and practical model of inference detection using a reference corpus. Our model is inspired by association rule mining: inferences are based on word co-occurrences. Using the model and taking the Web as the reference corpus, we can find inferences and measure their strength through web-mining algorithms that leverage search engines such as Google or Yahoo!.

Our model also includes the important case of private corpora, to model inference detection in enterprise settings in which there is a large private document repository. We find inferences in private corpora by using analogues of our Web-mining algorithms, relying on an index for the corpus rather than a Web search engine.

We present results from two experiments. The first experiment demonstrates the performance of our techniques in identifying all the keywords that allow for inference of a particular topic (e.g. “HIV") with confidence above a certain threshold. The second experiment uses the public Enron e-mail dataset. We postulate a sensitive topic and use the Enron corpus and the Web together to find inferences for the topic.

These experiments demonstrate that our techniques are practical, and that our model of inference based on word co-occurrence is well-suited to efficient inference detection.

Categories and Subject Descriptors

H.2.7 [**Database Management**]: Database Administration—*security, integrity, and protection*; I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition*; I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*inference engines*; K.4.1 [**Computers and Society**]: Public Policy Issues—*privacy*

General Terms

Algorithms, Security, Performance, Experimentation

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.

Keywords

Inference detection, inference control, association rule mining, web mining, search engine

1. INTRODUCTION

Imagine a team of government employees tasked with preparing military documents for Web site publishing. The corpus of documents is too large for anything more than cursory review by a human, and the topics covered by the documents include a broad array of sensitive topics (e.g. weapons development) as well as nonsensitive topics (e.g. purchase orders for office supplies). The team understands that it is not enough to look for known sensitive terms like “missile” and so they begin to look for other, seemingly innocuous, terms that might allow the government's missile development activities to be inferred. That is, the team works on the *inference detection* problem in this semi-structured data set.

To detect sensitive inferences, they turn to association rule mining technology (e.g. [2, 1]). Given a reference corpus, association rule mining analyzes the contents of the corpus to identify words closely associated with “missile”, thus potentially providing the team with an efficient way to identify the documents that might be sensitive and need human review. Since the team has no repository tailored to missiles, they decide to use the Web as their reference corpus.

The team quickly runs into problems. The massive size of the Web makes existing association rule mining algorithms impractical. In addition, attempts to work on smaller portions of the Web are still problematic because the well-known algorithms only detect associations with high support (e.g. [2]) or they depend on the user to lower-bound the support requirements (e.g. [24]) and the team simply doesn't have that information. In the end, the association mining algorithms fail to detect the association between the set of terms “infrared, gyroscope, radar” and “missile” due to the relatively small number of Web documents that contain any of the first three terms and “missile”. Consequently, a sensitive document concerning missile development is released.

Though this story is fictitious, such data leaks are commonplace. The Iraqi Freedom Document Portal and the Nuclear Regulatory Commission's site, are two recent examples of government Web sites that were shut down (in the case of the Portal, [6]) or significantly overhauled (the NRC's site, [3]) when they were found to contain sensitive documents.

We present a model and algorithms for mining such sensitive associations using large corpora such as the Web. This new model captures the essence of inference detection in a simple formalism inspired by association rule mining: inferences are based on word co-occurrences. Our model, however, captures a far larger variety of inferences than basic association rule mining. While association rule mining is restricted to “conjunctive" inferences (co-occurrences of items), our model also supports “disjunctive” inferences that establish a relationship between a (conjunctive) set of precedents (e.g., attributes of a person), $A_1, \ldots, A_n$ and a disjunctive set of consequents (e.g. suspected entities):

$$A_1 \wedge \ldots \wedge A_n \Rightarrow B_1 \vee \ldots \vee B_j.$$

A key challenge in inference detection that is not present in the traditional association rule mining setting is modelling the knowledge of an “adversary” (i.e. any individual from which sensitive data should be withheld) in order to predict what the adversary can infer. Since there is no database or corpus of the adversary’s knowledge, we approximate such a corpus with the most appropriate data set available (e.g. the Web). Hence, our reference corpus is an approximation, and consequently there is the possibility of false positives and negatives amongst the associations. A significant contribution of this work is adapting the information retrieval notions of precision and recall to this setting, thus providing metrics for assessing the accuracy of the discovered associations.

In [20] the notion of using the Web to represent the adversary’s knowledge is introduced. However, the model in [20] is incompletely specified and the algorithms proposed are computationally expensive and no mechanisms are provided for evaluating their success or failure. We build on this work with a fully-functional model that connects association rule mining and inference detection and supports far more efficient algorithms. For example, evaluating a single inference using the algorithm of [20] takes 150 seconds, while the fastest algorithm in this paper takes a mere 1.5 seconds.

We test our algorithms in two settings. First, we explore their use in healthcare privacy legislation compliance. Most U.S. states place restrictions on sharing information about the following sensitive topics: HIV/AIDS, genetic information, mental health, and communicable diseases [12]. Abiding by the intention of this legislation is very challenging as it requires protecting any information in medical records that can allow these sensitive topics to be inferred. It is not enough to protect obviously sensitive terms such as “HIV”; any medications or symptoms from which an HIV diagnosis can be inferred should be protected as well [22]. We provide experimental data demonstrating the performance of our algorithm in identifying all the keywords that allow for inference of a particular topic (“HIV" in our experiment) with confidence above a certain threshold.

The second experiment explores the protection of a corporation’s sensitive information (e.g. intellectual property, client data, etc.) using an internal data set as the reference corpus. In particular, we use the public Enron e-mail dataset to demonstrate how sensitive topics can be protected with our algorithms. We divide the Enron corpus into test and training sets and postulate a sensitive topic (“Wharton" in our experiment). We then use the training part of the Enron corpus and the Web together to find inferences for the topic. We evaluate the inferences found using the test part of the Enron corpus.

To estimate the precision and recall of our experiments we employ human review, lower bound calculations and a stability analysis over different training sets. The sum total of this analysis is substantial evidence of the good precision and recall achieved by our algorithms.

Together with our model, our experiments demonstrate that it is possible to mine associations efficiently even with a reference corpus as large as the Web, and to use these associations to protect privacy.

Overview. The rest of this paper is organized as follows. We review related work in the rest of this section. In Section 2, we give a precise definition of the problems of inference detection and inference control. In Section 3, we present our model of knowledge and inferences. We illustrate this model with two simple examples in Section 4. We discuss our algorithms for detecting inferences in Section 5, and present our experimental results in Sections 6 and 7. Finally, we conclude with a discussion of future work in Section 8.

1.1 Related Work

Our inference detection algorithms can also be viewed as algorithms for finding *association rules* that are of a sensitive nature. In our setting, an association rule is an implication of the form $A \Rightarrow B$, where A and B are disjoint sets of words, and B is of a sensitive nature, e.g., a medical condition like HIV or a person’s identity. Recall that an association rule is said to have high confidence if $\Pr(B|A)$ is large, and large support if $\Pr(A \wedge B)$ is large. Much of the association rule mining literature focuses on finding rules that have both high confidence and high support (see, for example, [2, 1]).

There are 3 important differences between our work on inference detection and conventional association rule mining.

The first difference is that in privacy applications, unlike data-mining applications, inferences need not have high support to be considered important or sensitive. Our goal is to find *all* associations of a sensitive nature, and thus we cannot limit ourselves to rules with large support. Indeed, a feature that makes the privacy problem so challenging (and any automated algorithmic solution so valuable) is that sensitive inferences are almost certain to have low support when viewed in the context of a large corpus such as the Enron data set or the Web. Hence, algorithms such as Apriori and AprioriTid [2] which prune low-support item sets (sets of words, in our case) are not directly applicable to our setting.

Recent research explores the discovery of rules meeting certain support constraints, thus potentially allowing lower support rules to be discovered (e.g. [24]). However, this approach assumes an understanding of minimum support constraints that is very difficult to achieve in the "needle in a hay stack" problem of finding sensitive inferences.

Second, conventional association rule mining assumes full access to a structured database (e.g. a supermarket database of transactions) or semi-structured corpus which, by definition of the task at hand, contains all the association rules of interest. Our goal is to detect the associations that may lead an adversary to make undesired inferences and unfortunately, there is no database of adversary knowledge from which to extract these associations. We approximate such a database using the best publicly available corpus for the context. For example, we experiment with using the Web to detect sensitive medical inferences, and a corporate email repository (the Enron corpus, [9]) to demonstrate the detection of inferences pertaining to a sensitive topic (e.g. new product plans). To mine these corpora we use Web search engines and Lucene [4], respectively.

Innovative algorithms have been developed for mining semi-structured data for association rules (e.g. [19, 5]) but again this work assumes the corpus at hand contains all the wanted association rules, so any that are discovered (and meet the required confidence and support goals) are necessarily valid. Our necessary reliance on corpora that approximate the nonexistent "adversary knowledge corpus", may lead to the discovery of erroneous rules and the failure to discover accurate ones. To deal with this issue, we discuss ways to measure the precision and recall of our results (see Sections 6 and 7).

A third difference stems from the fact that association rules are typically mined in the “market basket” setting with the goal of discovering when the purchase of a collection of goods is likely to be accompanied by the purchase of another good (e.g. milk is almost always purchased along with butter and bread). Implications stemming from a conjunction of items ($butter \wedge bread \rightarrow milk$) are typically enough to support the marketing recommendations (e.g. grocery store layout) that are the most common goal of association rule mining. Since our concern is implications that impact privacy, we are interested in more complex implications, beyond association rules, as well. For example, if $A \wedge B \rightarrow C \vee D$ and $A' \wedge B' \rightarrow C \vee E$ then in our setting it may be critical to notice that $A \wedge B \wedge A' \wedge B' \rightarrow C$, if, for example, C is the identity of an individual. Our model supports the detection of more complex inferences such as these.

Our approach to inference detection is in the same spirit as the use of the Web by Nakov and Hearst [18] to resolve language ambiguities. Their idea is to use co-occurrence on the Web to disambiguate phrases. In this paper we also make use of the Web to side-step the issue of training data, but with a different application, the detection of potential privacy violations. We use co-occurrence on the Web to model adversary knowledge and consequently, to detect undesired inferences that may be drawn from text.

We also note that a huge body of work in information retrieval, natural language processing (NLP) and data mining is based on the powerful word co-occurrence feature. Indeed, co-occurrence is at the root of many techniques for detecting synonyms (e.g. [23]), interpreting search engine queries (e.g. [11]), automatic indexing and annotation [7, 8] and problems in structural linguistics like discovering conventional expressions (e.g. [17]). Here, we exploit co-occurrence for a new application, the identification of sensitive inferences to support privacy.

The notion of using the Web to detect inferences was introduced in [20]. Our approach is significantly different from the algorithms in [20]. Every algorithm in [20] relies on analysis of Web pages to identify inferences. For example, to determine if conditions A and B imply a diagnosis of HIV, [20] would issue a search engine query "$A\,B$" and examine the resulting hits for the appearance of the term "HIV". In large part due to this analysis, the algorithms of [20] are slow and consequently, the algorithms contain shortcuts that lessen the depth of inference detection that is possible in order to keep running time at a reasonable duration. For example, in [20] a sensitive term such as "HIV" is only found if it occurs in the first 5000 lines of the html of a Web site. For sites with significant graphics this restriction is likely to enable the analysis of only a small fraction of the actual text.

In contrast, we leverage the indexing power of the search engine to avoid this costly step while doing more thorough inference detection. In short, we use the fact that the search engine can measure co-occurrence of terms *for* us to avoid doing any more content analysis than retrieving the hit counts. In addition, we develop the first rigorous model for corpora-based inference detection (with the Web and the Enron emails being our example corpora) and present approaches to measuring the precision and recall of our algorithms.

Finally we note that inference detection is a well-studied problem in the database community (see, for example [10]), where the problem is to find ways in which classified or otherwise sensitive information can be learned through a sequence of database queries for unclassified or non-sensitive information. In addition, Sweeney has looked at the problem of using the Web to identify inferences based on regular expressions such as social security numbers and account numbers [21]. Inference detection in structured databases, while certainly difficult, is nevertheless a simpler problem than the problem we consider of detecting inferences in free-form text documents.

2. PROBLEM DEFINITION

Let D denote a document, or a collection of documents. Informally, let $K(D)$ denote the "knowledge" (or facts, or axioms) that can be extracted from D. We assume the existence of knowledge composition rules, which specify how to derive new knowledge from the combination of existing pieces of knowledge. We write $\overline{K}(D)$ for the closure of $K(D)$ under the knowledge composition rules, i.e. the closed set of all knowledge obtained from $K(D)$ by repeated application of the composition rules.

Inference detection. Now 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. Informally stated, the problem of inference control comes from the fact that the "knowledge" that can be computed from the union of the private and reference collections $\overline{K}(\mathcal{C} \cup \mathcal{R})$ is typically greater than the union $\overline{K}(\mathcal{C}) \cup \overline{K}(\mathcal{R})$ of what can be extracted separately from $\mathcal{C}$ and $\mathcal{R}$. In its most general formulation, the inference detection problem is to understand the difference

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

Inference control. In almost all applications, we have a set S of sensitive or secret knowledge that the publication of $\mathcal{C}$ should not expose. In that case, the problem of inference control can be stated more precisely as the problem of ensuring that the intersection $S \cap \delta(\mathcal{C}, \mathcal{R})$ is empty. Inference control is closely tied with *redaction*, that is, the sanitization of a document by removal of some of the document's content. When the intersection $S \cap \delta(\mathcal{C}, \mathcal{R})$ is non-empty, an additional goal of inference control is to identify a subset $\mathcal{C}^{sub} \subset \mathcal{C}$ such that $S \cap \delta(\mathcal{C}^{sub}, \mathcal{R}) = \emptyset$. While $\mathcal{C}^{sub} = \emptyset$ trivially satisfies this condition, the goal might be to identify a subset $\mathcal{C}^{sub}$ of maximum size or to preserve as much information on a certain topic while protecting S.

Controlling privacy leaks by identifying the set $\mathcal{C}^{sub}$ that is appropriate to release is a highly challenging problem (particularly in light of attacks such as [16]) that is *not* the subject of this work. That said, since it is a crucial part of the content security problem, we outline some of the research issues around it when discussing open problems in Section 8.

Adversarial model. This formulation of the problem implies assumptions about our adversarial model, which we now detail explicitly. We assume that the adversary does not have any additional private knowledge beyond the reference collection $\mathcal{R}$. We also assume that changes to the collection $\mathcal{C}$ prior to its publication (e.g. redaction, obfuscation) do not themselves allow the adversary to infer information about $\mathcal{C}$. We also note that inference control may be accomplished in other ways than redaction; for example, words may be replaced or words may be added to provide inference control. In this paper our focus is on identifying, as opposed to controlling, inferences.

3. INFERENCE MODEL

In this section, we define the "knowledge extraction" function K, and the knowledge composition rules that allow us to compute $\overline{K}$ from K. Sophisticated formal languages have been developed in the NLP community to represent human knowledge (see [15] for a survey), but these heavy-weight languages would not allow for efficient computation of inferences. In this work, we adopt a simple formal representation of knowledge inspired by association rule mining: knowledge extraction is based on word co-occurrences and knowledge composition rules are the rules of Boolean logic. This model is well suited to efficient inference detection.

Association rule mining. The goal of association rule mining is to discover elements that frequently co-occur in a given data set. The best-known use of association rule mining is in market-basket analysis, where the data set consists of customers' purchases and the goal is to discover products that are often purchased together (e.g. bread and milk). We use an approach similar to association rule mining to model the knowledge present in a collection, $\mathcal{C}$, of documents. We search for sets of keywords that frequently co-occur in the collection $\mathcal{C}$, and model the knowledge in $\mathcal{C}$ with these sets. For example, if $\mathcal{C}$ is a collection of medical documents about HIV, we may learn the following sets of frequently co-occurring keywords: $\{$HIV,AIDS$\}$ or $\{$HIV,gp120$\}$. There is no question that this simple model captures only a fraction of the human knowledge embedded in $\mathcal{C}$. Nevertheless, we will demonstrate the power of this model to efficiently detect inferences.

Inference model. Adapting the framework of association rule mining to the detection of inferences in free form text documents raises both theoretical and algorithmic challenges. We address first the main theoretical differences between our model and association rule mining (algorithmic challenges are discussed in Section 5):

- Given a collection $\mathcal{C}$ of documents, the first challenge is to define the items associated with $\mathcal{C}$ and to represent $\mathcal{C}$ as a collection of item sets. One possible approach is to make every word in $\mathcal{C}$ an item, and to group words that appear in the same sentence into an item set. Other possibilities are discussed in Section 3.1.
- The second challenge is to define item sets of interest. In traditional association rule mining, interesting item sets are frequent item sets. In addition to frequent item sets, we propose definitions of interesting item sets that are more closely tailored to our privacy application (Section 3.2).

3.1 Items and Item Sets

Let $\mathcal{V}$ (for vocabulary) denote the set of all the words that appear in the collection $\mathcal{C}$ of documents. We assume that $\mathcal{V}$ contains, without restriction, all the words, numbers, dates, symbols, entity names, etc, which appear in the collection $\mathcal{C}$. We could define the set of items to be $\mathcal{V}$ itself, but this ignores the lexical structure of the language and the semantic content of $\mathcal{C}$. To take these into account, we let $\mathcal{T}$ denote the set of items and define a function $f : \mathcal{V} \Rightarrow \mathcal{T}$ which maps any word $v \in \mathcal{V}$ to an item $t \in \mathcal{T}$. The function f may take into account any (or all) of the following:

- **Lexical structure:** mapping a word to its lexical stem (e.g. “walks" ⇒ “walk").
- **Syntactic structure:** mapping a word to its grammatical function (e.g. “child" ⇒ “Subject:child"). Our experiments do not implement syntactic parsing.
- **Semantic structure:** filtering words to keep only those related to a particular application area. For example, f may define items for words in $\mathcal{V}$ related to medicine, or finance, or patent law, etc, and discard other words.

Boolean Formulas. Given the set of items $\mathcal{T}$, we can define boolean formulas of items. Let $\wedge$ denote the boolean operator AND, and $\vee$ denote OR. If $A \in \mathcal{T}$, we let $\overline{A}$ denote the negation of A. The notation $A \Rightarrow B$ is equivalent to $\overline{A} \vee B$.

Item Sets. Given a definition of items, we represent the collection, $\mathcal{C}$, of documents as a collection of item sets. The simplest representation is to create one item set for each document in the collection. The item set associated with a document consists of all the items contained in that document. More generally, we allow for finer grained definitions of item sets. We decompose every document in $\mathcal{C}$ into a collection of textual units, where a textual unit can be a sentence, a paragraph, a page, a section, etc. Let $\mathcal{C}'$ denote the resulting collection of item sets. We then associate one item set with every textual unit in $\mathcal{C}'$, consisting of all the items in that textual unit.

Our algorithms search for inference precedents in each textual unit, so for efficiency reasons it is desirable that a textual unit be associated with a single inference, that is, the topic of the textual unit. Our experimental results choose textual units with this in mind.

Support of a formula. Let $S \in \mathcal{C}'$ be an item set, and let F be a Boolean formula of terms. Viewing the set S as a conjunction of items, we say that S satisfies F if $S \Rightarrow F$. We define the support $\text{Supp}(F)$ as the probability that S satisfies F for an item set $S \in \mathcal{C}'$. In other words:

$$\text{Supp}(F) = \Pr_{S \in \mathcal{C}'}[S \Rightarrow F]$$

3.2 Inference Rules

The knowledge extracted from $\mathcal{C}$ is represented as a list of inferences of the form $A \Rightarrow B$, where A and B are Boolean formulas of items. We adopt the terminology of association rule mining and call A the *antecedent* and B the *consequent* of the inference. We review briefly definitions used in association rule mining to determine the importance of an inference, and discuss the relevance of these definitions to inference detection.

DEFINITION 1. *The support of an inference $A \Rightarrow B$ is the support of $A \wedge B$.*

DEFINITION 2. *The confidence of an inference $A \Rightarrow B$ is the ratio Supp$(A \wedge B)$/Supp(A).*

Association rule mining typically searches for inferences with high support and high confidence. High support is a much weaker indicator of the importance of an inference in our privacy application, since even an inference with low support may allow an adversary to draw damaging conclusions.

Logical closure. Our algorithms will search for inferences with confidence above certain thresholds. These inferences will constitute the knowledge $K(\mathcal{C})$ extracted from the collection $\mathcal{C}$ of documents. Given this seed set of inferences, the closure $\overline{K}(\mathcal{C})$ is computed from the inferences by application of standard Boolean rules.

4. EXAMPLES

Before describing our inference detection algorithms, we illustrate the model of Section 3 and the challenges of inference detection using large corpora with two examples.

Simple Inference. Let's assume that the private collection $\mathcal{C}$ consists of the medical record of a single patient. We assume that the reference collection $\mathcal{R}$ consists of all Web pages indexed by a search engine, and we consider each Web page as a distinct textual unit.

In this example, items are medical keywords (we ignore other words). Assume that we have extracted the keyword gp120 from the collection $\mathcal{C}$, and we want to measure the confidence of the inferences (gp120 ⇒ HIV) and (gp120 ⇒ Flu) using the knowledge extracted from $\mathcal{R}$.

Since we have defined the textual units of $\mathcal{R}$ to be Web pages, the support $\text{Supp}(W)$ of an item W is simply the fraction of Web pages which contain the keyword W. This fraction can be obtained from the search engine with a single query. Using Google for example, we learn that Supp(gp120) $= 991,000$ and Supp(gp120 $\wedge$ HIV) $= 919,000$. It follows that

$$\text{Confidence (gp120} \Rightarrow \text{HIV)} \approx 0.93$$

Similarly, Supp(gp120 $\wedge$ Flu) $= 27,500$, and thus

$$\text{Confidence (gp120} \Rightarrow \text{Flu)} \approx 0.03$$

With a confidence threshold set at 0.2, the inference (gp120 $\Rightarrow$ Flu) would (correctly) not be considered significant, but the inference (gp120 $\Rightarrow$ HIV) would (correctly) be considered very significant (gp120 is a glycoprotein that attaches to the HIV retrovirus).

Complex Inferences. In the simple example above, the precedent (gp120) and consequent (HIV) of the inference consist of a single keyword. Our model, however, allows the precedent and consequent to be arbitrarily complex Boolean formulas of keywords. For example, we can define the confidence of the inference $A \vee B \Rightarrow (C \vee (D \wedge E))$, where A, B, C, D and E represent keywords. We can not only define these complex inferences, but also measure their confidence, since most search engines support both disjunctive and conjunctive queries. In practice, our ability to measure the confidence of complex inferences is limited only by the sparseness of the Web, which we discuss next.

Let F and G denote Boolean formulas of terms. The confidence of the inference $F \Rightarrow G$ is the ratio of the support of F and the support of $F \wedge G$. For complex formulas F and G, the support of $F \wedge G$ may be zero: there are no documents on the Web that satisfy both the formulas F and G. This makes it impossible to directly compute the confidence of $F \Rightarrow G$. Consider the following example:

$$F = (\text{Lowes Lake} \wedge \text{Las Vegas} \wedge \text{C5.0} \wedge \text{SVM})$$

$$G = \text{KDD-08}$$

The support of $F \wedge G$, as measured by Google, is zero. This is not to say that the inference $F \Rightarrow G$ is invalid. In fact, mention of “C5.0" and “SVM" in Las Vegas at the Lowes Lake Hotel, most likely implies the KDD-08 conference. But the confidence of this inference cannot be directly measured due to the sparseness of the Web.

The knowledge composition rules defined in our model can in theory help mitigate the problem of the sparseness of the Web. We offer the following made-up example as illustration. Suppose that we have found the following inferences to have high confidence:

$$(\text{Las Vegas} \wedge \text{Lowes Lake}) \Rightarrow (\text{KDD} \vee \ \vee \dots)$$

$$(\text{C5.0} \wedge \text{SVM}) \Rightarrow (\text{KDD} \vee \text{SIGMOD} \vee \dots)$$

If “KDD" is the only term in the intersection of the consequents of these two inferences, Boolean composition rules allow us to infer $F \Rightarrow G$ with high confidence.

In the rest of this paper, we focus on algorithms for measuring the support and confidence of fairly simple inferences. Our algorithms and experiments do not illustrate the logical combination of simple inferences to compute more complex inferences. Logical combination is nevertheless a powerful feature of our model, that we plan to explore in future work. Note that one way to improve the results of Section 6 would be to leverage logical combinations.

5. ALGORITHMS FOR MEASURING INFERENCE CONFIDENCE

In Section 3.1, we define the support of a formula as

$$\text{Supp}(A) = \Pr_{S \in \mathcal{C}'}(S \Rightarrow A).$$

In this section, we discuss how to compute Supp(A) in practice. Recall that the definition of support depends on a collection of documents $\mathcal{C}$ and on the textual sub-units used to define $\mathcal{C}'$.

Given the collection of documents $\mathcal{C}$ and sufficient computational resources, it is easy to compute Supp(A) for any formula A. Unfortunately, unrestricted access to $\mathcal{C}$ is not always possible. In what follows, we focus on the difficult but common case in which the collection $\mathcal{C}$ consists of all documents publicly available on the Web. Crawling the whole Web is an expensive operation, so we cannot assume direct knowledge of $\mathcal{C}$. Instead, we rely on search engines to mediate access to $\mathcal{C}$ and discuss various techniques for estimating Supp(A) via queries to a search engine.

Measuring support. For simplicity, we start with the assumption that the formula A is a conjunction of terms: $A = V_1 \wedge \dots \wedge V_k$. (Some search engines also provide limited support for disjunctive queries). We issue a query to a Web search engine for the keywords $V_1, \dots, V_k$ (note that search engines interpret such queries conjunctively by default), and let n_A denote the number of documents found by the search engine to match this query. Let N denote the total number of documents indexed by the search engine. We estimate the support of A as Supp(A) $\approx n_A/N$. Note that the normalizing factor N serves to ensure Supp(A) $\in [0, 1]$ and need not be known precisely.

Estimate of confidence. The technique above for estimating the support of a formula allows us to estimate the confidence of $A \Rightarrow B$ as

$$\text{Confidence}(A \Rightarrow B) \approx n_{A \wedge B}/n_A.$$

We call this estimate the PMI-IR estimate of confidence, after Turney [23], who used the same technique on single terms to rate their similarity. Note that the PMI-IR estimate of confidence can be computed very efficiently: it requires only two search engine queries.

6. ENUMERATION EXPERIMENT

The inference enumeration problem is the problem of determining all the precedents A which imply a given consequent B with confidence above a certain threshold. In our experiment, we chose the consequent “HIV" and searched for precedents that allow for inference of HIV. This experiment is motivated by the practical problem of redacting from a medical record all information that might allow for inference of HIV infection. We limited our search for precedents to single keywords, but the same technique would allow us to test pairs, triplets, and more generally tuples of keywords.

6.1 Generation of candidate inferences

We took a simple approach to generating candidate precedents that generalizes easily. We issued a query for “HIV" to a search engine (via Yahoo!'s web search API [25]) and retrieved the top 10 hits. Results included the Wikipedia article on HIV and “HIV In-Site", a site with information on HIV maintained by the University of California San Francisco. We discuss the choice of the number of documents to retrieve in Section 6.2.

We processed these 10 documents with the Apache Lucene indexer [4] and extracted 2349 distinct keywords from the documents. For all these keywords, we measured the confidence of the

1.00	hiv-1	0.84	emtricitabine	0.74	didanosine	0.68	emedicinehealth	0.60	invirase
1.00	hiv-2	0.83	coinfected	0.74	indinavir	0.68	coreceptor	0.60	croi
0.99	nnrti	0.83	aidsinfo	0.74	abacavir	0.68	gp160	0.58	cxcr4
0.96	aidsmap	0.82	lopinavir	0.74	fosamprenavir	0.68	retroviruses	0.57	fortovase
0.94	etravirine	0.82	lipodystrophy	0.74	mycobutin	0.67	viread	0.57	medterms
0.91	antiretroviral	0.81	nelfinavir	0.73	atazanavir	0.65	lexiva	0.57	pneumocystis
0.91	thebody.com	0.80	microbicides	0.72	epzicom	0.65	immunodeficiency	0.56	carinii
0.91	retrovirals	0.78	stavudine	0.72	zalcitabine	0.64	rifabutin	0.55	crixivan
0.89	gp41	0.77	truvada	0.71	ziagen	0.64	seroconversion	0.55	kaletra
0.89	enfuvirtide	0.77	saquinavir	0.70	trizivir	0.64	viracept	0.54	mtct
0.86	rescriptor	0.76	delavirdine	0.70	ritonavir	0.63	retrovir	0.53	lentivirus
0.86	gp120	0.76	amprenavir	0.70	lamivudine	0.62	lymphotropic	0.53	hemophiliacs
0.85	tenofovir	0.76	zidovudine	0.70	combivir	0.62	htlv	0.50	foscarnet
0.84	unaids	0.76	ccr5	0.69	integrase	0.61	norvir		
0.84	nevirapine	0.75	efavirenz	0.68	progressors	0.61	vaginosis		

Figure 1: Top precedents which imply HIV, with support greater than 100,000, ranked in decreasing order of confidence. Confidences were computed using the PMI-IR algorithm described in section 5.

inference Keyword ⇒ HIV using the PMI-IR algorithm defined in Section 5.

6.2 Results

The PMI-IR algorithm requires two queries to a search engine to estimate the confidence of an inference. Our implementation of the PMI-IR algorithm tested all 2349 keywords in approximately 70 seconds, whereas the algorithm of [20] reportedly took more than 6 hours to test 435 inferences.

The output of our algorithm was a list of all 2349 precedents, ranked in decreasing order of the confidence with which they imply HIV. The table of Figure 1 shows the top precedents which imply HIV with confidence greater than 0.50, among precedents with support greater or equal to 100,000. There were 70 of these precedents in all. The lower-bound of 100,000 on the support of the inference is arbitrary, and was chosen only to present the reader with a list of precedents that is not too obscure. In a specialized medical application, precedents with lower support would be equally important.

Precision. We estimate the precision of our inference enumeration algorithm as the fraction of "correct" inferences among these 70 inferences (i.e. the inferences of Figure 1). A medical expert (a licensed physician in internal medicine with HIV-infected patients) evaluated the 70 inferences of Figure 1 manually. He classified 53 of the 70 inferences as correct, i.e. the precedent of these inferences is clearly related to HIV. Of the remainder, some precedents were not necessarily connected to HIV but did trigger the thought of HIV (for example, "hemophiliacs").

Interestingly, some of the precedents deemed by our medical expert to not imply HIV were "ccr5" (an HIV protein), "UNAIDS" (the United Nations AIDS effort), and various AIDS web-sites such as "aidsinfo" and "thebody.com". This suggests that a more complete review of our inferences would require a panel of HIV experts, including perhaps a molecular biologist or an HIV sociologist. The difficulty in obtaining a complete review of our results underscores our point that detecting inferences is a difficult, time-consuming and costly task. It also highlights the potential of our automated approach for enumerating inferences, which uses the Web as a proxy for all human knowledge and thus draws from all disciplines.

Recall. We define the recall of our algorithm as the fraction of precedents that allow for definite inference of HIV found by our algorithm. The recall of the algorithm is hard to estimate, since there exists no comprehensive list of precedents that allow for inference of HIV. To estimate recall, we resort to indirect evidence that our algorithm found "most" precedents that definitely imply HIV.

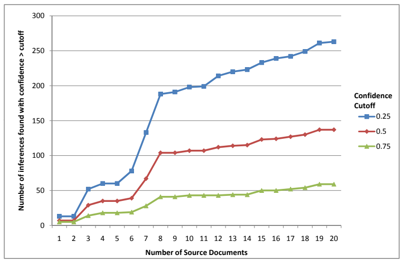

Figure 2: Number of inferences found for HIV as a function of the number of source documents, for different values of the confidence cutoff.

Figure 2 shows the number of inferences found as a function of the number of source documents used to generate candidate precedents, for different values of the cutoff below which inferences are deemed insignificant. Not surprisingly, the graph shows that generating more candidate precedents from more source documents yields more inferences. However, the graph suggests a sublinear relationship between the number of source documents and the number of inferences found. The addition of new source documents yields few additional inferences after around 10 documents, implying that at least for the "HIV" precedent, reasonable recall would be achieved when the algorithm uses 10 source documents.

In many applications the inference detection occurs in the context of a private corpus. In Section 7, we show that the use of a private corpus helps focus the inferences found and improves recall.

7. PRIVATE CORPUS

For our second experiment, we describe the use of our inference detection algorithms in the presence of a private corpus. Potential application scenarios include:

E-discovery. A corporation is sub-poenaed for all documents related to a given topic. We describe inference detection algorithms that help find all these documents, using the private corpus of internal corporate documents.

Data Leak Prevention. A corporation wants to ensure details on highly secretive projects are not leaked. For instance, before the release of the iPhone, terms such as SIM card or GSM contained in outbound Apple e-mails might merit closer scrutiny.

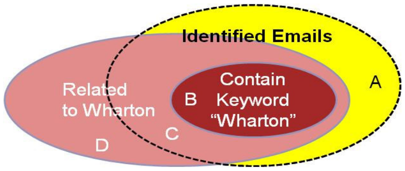

Figure 3: Precision and Recall Schematic

We describe inference detection algorithms that will help find these terms using the internal Apple corporate documents.

In our experiment, we held part of the private corpus in reserve as a test set and used the remainder as input to our algorithms to generate inferences for a given sensitive topic. We then used these inferences on the test set to obtain a set of flagged documents. We analyzed the set of flagged documents in terms of precision and recall.

Of course, for these applications one can proceed as in Section 6 and simply use the Web to find inferences for a given topic. We argue, however, that this approach does not leverage the knowledge contained in the private corpus and better results will be obtained using the techniques that follow.

Description of Enron e-mail corpus. The Enron e-mail corpus consists of e-mail from about 150 senior managers of Enron and contains over half a million messages. The corpus was made public as part of the Federal Energy Regulatory Commission Enron investigation. We use a cleaned version of the dataset available at [9]. The data leak prevention company InBoxer [13] uses the same Enron corpus to demonstrate their outbound e-mail scanning technology [14].

Generation of candidate inferences. In our experiment, we chose the topic of “Wharton", the business school of the University of Pennsylvania. This is a well-known public entity, so that it is easy to evaluate the quality of the inferences detected. Almost 800 messages in the Enron corpus contain the term “Wharton". Note that there are a handful of e-mails that contain the term Wharton in reference to Wharton, the county of Texas. We divided the Enron corpus into test and training sets by date, so that roughly each set contained half the e-mail with the term Wharton.

We generated candidate inferences for Wharton by taking all terms from all e-mails in the training set containing the term Wharton. Each candidate inference has two measures of confidence, the usual Web confidence, as in Section 5, and an analogous confidence computed using the private corpus:

$$\text{Confidence}(A \Rightarrow B) \approx \frac{\text{\# docs containing A and B}}{\text{\# docs containing A}}$$

Note that we also generated candidate inferences by the method of Section 6, but we found that this additional step rarely generated anything of value. Indeed, the value of the private corpus is that it helps provide an efficient way to come up with relevant candidate inferences. The private corpus compensates for the sparseness of the Web and also acts as a counter balance to the all-encompassing nature of the Web.

Estimation of Precision. Precision is the percentage of identified e-mails that are indeed about Wharton. To calculate precision, we approximate by assuming the e-mails about Wharton contain the Wharton keyword. Note that this is an underestimate of precision, neglecting the e-mails about Wharton, the county in Texas.

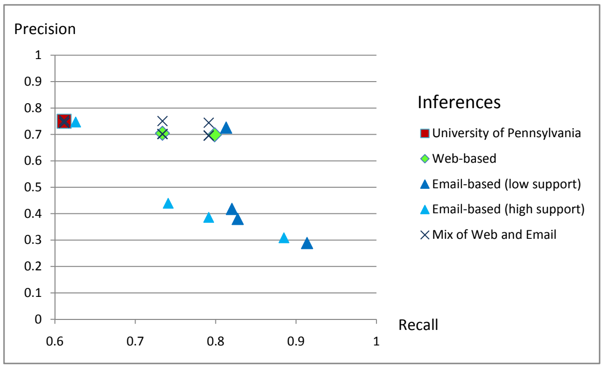

Figure 4: Precision and Recall curve of “Wharton" inferences

In Figure 3, precision is $(B+C)/(A+B+C)$. We approximate precision with $B/(A+B+C)$.

Estimation of Recall. Recall is the percentage of e-mails that are identified out of all e-mails about Wharton. Recall is difficult to practically evaluate, because it requires looking at several hundred thousand e-mails to see if they are about Wharton.

Hence, to calculate recall, we manually reviewed the e-mails containing the Wharton keyword to see if Wharton could be inferred even without the "Wharton" keyword. In this way, we obtained a set of e-mails that could be inferred to be about Wharton even without the Wharton term. Then the estimate for recall is the percentage of these e-mails identified *where we do not use the trivial inference (wharton ⇒ wharton)*.

In Figure 3, recall is $(B+C)/(B+C+D)$. We approximate recall by restricting to subset B.

7.1 Results

Our results are shown in Figure 4. The figure shows the trade-off between precision and recall (as defined in the previous section) for emails identified as sensitive in the Enron corpus via different sets of inferences.

We use as a baseline a simple set of two inferences that was manually generated: (wharton ⇒ wharton), and (university of pennsylvania ⇒ wharton). These two inferences are a good proxy for inferences that a non-expert human would find manually in a short period of time. Identification of sensitive emails using this basic set of two inferences achieves high precision (75%) but mediocre recall (61%). This baseline data-point is shown as a square in Figure 4.

Next, we used our inference detection technology to generate additional inferences, using the training portion of the Enron dataset. For each inference, we computed two (confidence, support)-pairs, corresponding to the two corpuses of the Web and Enron training set. For various cutoff values of confidence and support in the two corpuses, we obtained points in a precision-recall graph, as seen in Figure 4.

- The triangles show the precision and recall obtained using purely the Enron training set confidence and support. The light triangles correspond to a high support cutoff and the dark triangles correspond to a low support cutoff.
- The diamonds show the precision and recall obtained using purely Web confidence and support.
- The crosses show the precision and recall obtained using both Web and Enron training set confidences and supports.

Wharton professors	Kleindorfer, Kunreuther, Kunruether, Reibstein, Harker, Farber
Wharton students	Degiacinto
University of Pennsylvania	upenn, www.upenn.edu, dh/6371, Sansom
Hotel on Wharton campus	www.innatpenn.com, 1-800-809-7001
Wharton ZIP codes	19104, 19104-6371, 19104-6366
Wharton phone numbers	573-5727, 573-8394, 573-2130, 215-222-4600, 215-573-7722, 215-573-2129, 215.222.0200, 215-222-0200, 215-573-2130, 215-898-4589
Other business schools	Draganska, Tadelis, Phanish, INSEAD
Unknown / errors	ERASMUS, bringstogether, energyexpress, ipnetwork.com, barker.online, communicade, Halich, Mattesich, 853-6848, 482-8411

Figure 5: Top 40 precedents which imply Wharton.

To summarize our results, our tools for generating additional inferences allow us to increase recall by 20% (from 61% to 81%) with only a barely noticeable degradation in precision (from 75% to 73%). We achieved recalls as high as 91%, but at the cost of a significant drop in precision (to 29%).

To illustrate the types of inferences that we were able to detect, we give in Figure 5 the list of the top 40 precedents that we found for the consequent “Wharton", using both the Web and the private Enron corpus. This list of 40 inferences was generated with the following parameters. For the Email corpus, we requested a support ≥ 2 and confidence > 0.6. For the Web corpus, we requested support > 5 and confidence > 0.01. The list was manually classified into categories for the sake of clarity. The number and diversity of inferences found automatically by our inference detection tool suggest that these tools would be very valuable in helping humans review and detect sensitive inferences.

8. CONCLUSION AND FUTURE WORK

We have given a general theoretical framework for describing inferences and using Web-based probabilities to rate their strength. We evaluated the strength of the identified inferences using known web-mining algorithms. We presented a case study of detecting HIV inferences: we generated precedents that implied HIV, rated their strength, and submitted the results for human review. We presented another case study of detecting inferences for “Wharton" using the Enron e-mail corpus.

Our techniques provide an efficient mechanism for detecting sensitive content. In practice, they might be used to identify documents that need human review before their release.

As mentioned in Section 2, the focus of this paper is detecting inferences rather than the equally challenging task of protecting against inferences. Protection options include selective removal of text (i.e. redaction), encryption of text, sanitization of text (including introducing noise) and quarantining of content. Each approach comes with security and usability concerns that are difficult to evaluate. We highlight the need for a system that supports the user in making protection decisions as an open research problem.

An additional avenue for future work is leveraging the analysis of Web structure and document content to improve our inference detection algorithms. For instance, an analysis of the effect of subdocument item sets and the weighting of item sets according to some measure of authoritativeness may help reduce false positives and false negatives.

9. ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support for this project by Fujitsu Limited, Japan. We would also like to thank the anonymous reviewers of previous versions of this paper and Tomoyoshi Takebayashi and Toshihiro Sonoda of Fujitsu Laboratories Ltd. for many helpful comments.

10. REFERENCES

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. AAAI/MIT Press, 1996.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[3] M. Ahlers. Blueprints for terrorists? On the Web at `http://www.cnn.com/2004/US/10/19/terror.nrc/index.html`.

[4] Apache Lucene project. On the Web at `http://lucene.apache.org/`.

[5] M. Berardi, M. Lapi, P. Leo, and C. Loglisci. Mining generalized association rules on biomedical literature. In *IEA/AIE'2005: Proceedings of the 18th international conference on Innovations in Applied Artificial Intelligence*, pages 500–509, London, UK, 2005. Springer-Verlag.

[6] W. Broad. U. S. web archive is said to reveal a nuclear primer. On the Web at `http://www.nytimes.com/2006/11/03/world/middleeast/03documents.html`.

[7] P. Cimiano and S. Staab. Learning by googling. *SIGKDD Explor. Newsl.*, 6(2):24–33, 2004.

[8] M. Dowman, V. Tablan, H. Cunningham, and B. Popov. Web-assisted annotation, semantic indexing and search of television and radio news. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 225–234, New York, NY, USA, 2005. ACM.

[9] Enron corpus. On the Web at `http://www.cs.cmu.edu/~enron/`.

[10] C. Farkas and S. Jajodia. The inference problem: a survey. *SIGKDD Explor. Newsl.*, 4(2):6–11, 2002.

[11] N. S. Glance. Community search assistant. In *Intelligent User Interfaces*, pages 91–96, 2001.

[12] Health Privacy Project. On the Web at `http://www.healthprivacy.org/`.

[13] Inboxer. On the Web at `http://www.inboxer.com/`.

[14] Inboxer's Enron demonstration site. On the Web at `http://www.enronemail.com/`.

[15] L. M. Iwanska and S. C. Shapiro. *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. AAAI Press, 2000.

[16] D. P. Lopresti and A. L. Spitz. Information leakage through document redaction: attacks and countermeasures. In *DRR*, pages 183–190, 2005.

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

[18] P. Nakov and M. Hearst. Using the web as an implicit training set: application to structural ambiguity resolution. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural*

Language Processing, pages 835–842, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[19] L. Singh, P. Scheuermann, and B. Chen. Generating association rules from semi-structured documents using an extended concept hierarchy. In F. Golshani and K. Makki, editors, *Proceedings of the Sixth International Conference on Information and Knowledge Management (CIKM'97), Las Vegas, Nevada, November 10-14, 1997*, pages 193–200. ACM, 1997.

[20] J. Staddon, P. Golle, and B. Zimny. Web-based inference detection. In *Proceedings of 16th USENIX Security Symposium*, pages 71–86, Boston, MA, 2007. USENIX Association.

[21] L. Sweeney. AI technologies to defeat identity theft vulnerabilities. In *AAAI Spring Symposium on AI TEchnologies for Homeland Security*, 2005.

[22] N. Terry and L. Francis. Ensuring the privacy and confidentiality of electronic health records. *Illinois Law Review*, 2007(2).

[23] P. D. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, pages 491–502, London, UK, 2001. Springer-Verlag.

[24] K. Wang, Y. He, and J. Han. Pushing support constraints into association rules mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):642–658, 2003.

[25] Yahoo! Web Search API. On the Web at `http://developer.yahoo.com/search/web/`.